sciendo

JAISCR, 2024, Vol. 14, No. 3, pp. 207 – 235
10.2478/jaiscr-2024-0012

A HYBRID EQUILIBRIUM OPTIMIZER BASED ON MOTH FLAME OPTIMIZATION ALGORITHM TO SOLVE GLOBAL OPTIMIZATION PROBLEMS

Zongshan Wang[1], Ali Ala[2], Zekui Liu[3,*], Wei Cui[4], Hongwei Ding[1], Gushen Jin[5], Xu Lu[1]

[1]*School of Information Science and Engineering, Yunnan University, Kunming, China*

[2]*Department of Industrial Engineering and Management, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China*

[3]*Electronic Information School, Chongqing Institute of Engineering, Chongqing, China*

[4]*College of Computer Science and Technology Chongqing University of Posts and Telecommunications, Chongqing, China*

[5]*Glasgow College, University of Electronic Science and Technology of China, Chengdu, China*

[*]*E-mail: zswang.ynu@gmail.com*

Submitted: 5th September 2023; Accepted: 3td March 2024

Abstract

Equilibrium optimizer (EO) is a novel metaheuristic algorithm that exhibits superior performance in solving global optimization problems, but it may encounter drawbacks such as imbalance between exploration and exploitation capabilities, and tendency to fall into local optimization in tricky multimodal problems. In order to address these problems, this study proposes a novel ensemble algorithm called hybrid moth equilibrium optimizer (HMEO), leveraging both the moth flame optimization (MFO) and EO. The proposed approach first integrates the exploitation potential of EO and then introduces the exploration capability of MFO to help enhance global search, local fine-tuning, and an appropriate balance during the search process. To verify the performance of the proposed hybrid algorithm, the suggested HMEO is applied on 29 test functions of the CEC 2017 benchmark test suite. The test results of the developed method are compared with several well-known metaheuristics, including the basic EO, the basic MFO, and some popular EO and MFO variants. Friedman rank test is employed to measure the performance of the newly proposed algorithm statistically. Moreover, the introduced method has been applied to address the mobile robot path planning (MRPP) problem to investigate its problem-solving ability of real-world problems. The experimental results show that the reported HMEO algorithm is superior to the comparative approaches.

Keywords: hybrid algorithm, equilibrium optimizer, moth flame optimization algorithm, metaheuristics, benchmark functions, mobile robot path planning

1 Introduction

The optimization process is central to addressing numerous complex problems in applied mathematics and engineering sciences. Optimization involves identifying the best feasible solution from a given set of alternatives while considering the problem's constraints and objectives [1, 88]. It encompasses various objectives such as maximizing profits or quality and minimizing time, cost, or risk. Many engineering problems inherently require optimization techniques. However, finding precise solutions within a reasonable time frame can be challenging [2, 89].

Conventional optimization methods often face limitations such as sensitivity to initialization estimates and a tendency to converge on local optima. Furthermore, as the dimensions of the optimization problem increase, classical methods struggle to find the global optimum solution. This limitation arises due to the exponential growth of the search space [3, 4]. To overcome these challenges, metaheuristic algorithms have emerged as effective approaches for solving such problems. Metaheuristics offer robust and versatile optimization frameworks that can tackle complex real-world optimization problems by leveraging heuristics inspired by natural phenomena or problem characteristics. Therefore, metaheuristic algorithms have gained widespread popularity as they are able to find acceptable solutions with relatively fewer computational costs [5].

Based on their source of inspiration, metaheuristic algorithms can be categorized into four main groups: evolution-based, swarm intelligence-based, human-based, and physics-based. Evolutionary algorithms mimic the principles of biological evolution [6, 7], and these algorithms usually use operators inspired by natural evolutionary processes. One notable algorithm in this category is the genetic algorithm (GA) [8], which draws inspiration from Darwin's theory of evolution. GA utilizes operators such as crossover and mutation, which are based on biological behaviors, to modify the initial population. Building upon the success of GA, several other evolution-based algorithms have been introduced, including evolution strategy (EA) [9], differential evolution (DE) [10], and evolutionary programming (EP) [11].

Swarm intelligence represents another class of metaheuristic algorithms that simulate the collective behavior of animals in groups, such as hunting or migrating, to solve optimization problems. The key characteristic of these algorithms is the interaction of information among individuals during the optimization process, enabling the search for the optimal solution [12]. One prominent example in this group is the particle swarm optimization (PSO) [13], which mimics the behavior of birds flying in flocks searching for food. The core mechanism of PSO involves particles updating their flight speeds based on their historical best positions and the group's best position, allowing them to explore promising areas within the search space. Other notable algorithms in this category include the firefly algorithm (FA) [14, 15], artificial bee colony (ABC) [16], ant colony optimization (ACO) [17], grey wolf optimizer (GWO) [18], salp swarm algorithm (SSA) [19], spotted hyena optimizer (SHO) [20], seagull optimization algorithm (SOA) [21], harris hawks optimization (HHO) [22], moth flame optimization (MFO) [23], marine predators algorithm (MPA) [24], naked mole-rat algorithm (NMRA) [25], slime mould algorithm (SMA) [26], colony predation algorithm (CPA) [27], and monarch butterfly optimization (MBO) [28], among others.

The family of human-based metaheuristic algorithms draws inspiration from human collaborative behavior and group interaction. These algorithms aim to implement optimization processes by emulating the dynamics of human interaction [29]. One example is the teaching-learning-based optimization (TLBO) [30], which is inspired by the pedagogical relationship between teachers and students. The TLBO algorithm consists of two phases: the teacher phase, where the teacher imparts knowledge to the learners, and the student phase, where learners exchange information with each other to make progress. Other instances of human-based algorithms include the imperialist competitive algorithm (ICA) [31], socio evolution learning optimization algorithm (SELOA) [32], hunger games search (HGS) [33], and poor and rich optimization (RPO) [34].

The final category of metaheuristic techniques is physics-based algorithms, which draw inspiration from the principles of physics. These algorithms often model physical processes and derive corre-

sponding mathematical models to describe the interactions among search agents [35]. One widely known algorithm in this group is the simulated annealing (SA) [36], which emulates the annealing process in solids and is a probability-based stochastic optimization technique. Another example is the big bang-big crunch (BBBC) [37], which simulates the theories of the evolution of the universe. The gravity search algorithm (GSA) [38] is another popular algorithm inspired by the interplay between gravity and mass. It employs the laws of gravity to update the search agents' positions in order to discover the optimal solution. Additionally, there are several other well-regarded physics-based algorithms, including but not limited to the multi-verse optimizer (MVO) [39], galaxy-based search algorithm (GbSA) [40], special relativity search (SRS) [41], artificial electric field algorithm (AEFA) [42], rime optimization algorithm (RIME) [43], and equilibrium optimizer (EO) [44].

Moreover, a segment of algorithms finds its inspiration in the realm of art. These algorithms draw inspiration from the artistic creativity exhibited by professionals such as architects, painters, and musicians, utilizing the unique behaviors manifested in the creative process. Despite the limited scope of application for art-inspired algorithms, they have demonstrated significant success in the field of global optimization. Notably, the harmony search (HS) algorithm [45] serves as a typical example, reflecting the spontaneous and improvisational nature of musicians striving for perfect harmonic states in musical note composition. Since its inception, HS has attracted researchers from various disciplines, giving rise to several other art-inspired algorithms. Examples include the stochastic paint optimizer (SPO) [46], color harmony algorithm (CHA) [47], and interior search algorithm (ISA) [48]. SPO derives its inspiration from the creative strategies employed by artists in painting, incorporating principles of color theory, color wheels, and techniques for combining colors. CHA outlines its search dynamics by blending harmonious colors, grounded in their relative positions along the hue circle within the Munsell color system and conforming to consonant templates. Influenced by interior design and decoration, ISA has designed a distinctive search framework with the ability to identify global optima.

All the aforementioned metaheuristic algorithms aim to find the global optimum from all possible inputs [49]. Collaboration between search strategies plays a crucial role in these algorithms. Specifically, exploitation and exploration are two key characteristics of metaheuristic techniques. Exploration involves the ability to discover unexplored regions, while development is the capability to fine-tune the search space with potential [50]. However, these two aspects conflict with each other, enhancing one aspect's capability exacerbates the weakness of the other. This implies that maintaining an optimal balance between development and exploration to achieve the best algorithmic performance is challenging. Intensive exploration and limited exploitation often result in the algorithm's inability to reach the target region. On the other hand, strong exploitation and weak exploration lead to getting stuck in local optima. Researchers often combine one algorithm with other optimization algorithms or those with complementary characteristics to enhance the search capabilities of the original algorithm.

The collaboration pattern and level between individual search agents helps to improve the algorithm performance. EO, proposed by A. Faramarzi et al., is a physics-based metaheuristic algorithm. In comparison to algorithms such as PSO, GWO, and SSA, EO exhibits superior search performance. Additionally, EO has become one of the most popular metaheuristic techniques due to its simplicity and user-friendly structure [51]. It has been successfully applied to various engineering optimization problems, including solar photovoltaic parameter extraction and distributed power allocation. Despite its significant success, EO has drawbacks such as local optimum convergence and poor exploration capability. To address these issues, researchers have proposed different versions of EO variants. In [52], a tangent search mechanism and symbiotic phase were introduced to EO to eliminate stagnation in local optimal solutions on the basic EO search space. Test results on 34 multidimensional optimization functions and 14 real-world constrained engineering problems confirmed the effectiveness of the EO variants. In [53], chaos theory was employed to enhance algorithm performance, and the developed algorithm was tested on benchmark suites and feature selection problems. Experimental results indicated that the Chaos EO al-

gorithm outperformed its competitors. In [54], a quasi-oppositional learning mechanism and chaotic mapping were injected into basic EO to improve algorithm scalability while reducing local stagnation. The proposed EO variant was tested on 96 benchmark functions and 6 engineering optimization problems, comparing favorably against various advanced metaheuristic techniques. Finally, in [55], adaptive mechanisms, Cauchy perturbation, and cosine search strategies were introduced into EO to enhance the overall search capabilities of the algorithm. Testing on 15 classic test functions and the CEC 2017 benchmark test set demonstrated that the algorithm outperformed the comparison algorithms.

On the other hand, MFO is a widely embraced swarm-based global optimization algorithm that exhibits substantial exploration capability. However, due to the nature of its operators, the algorithm's exploitation ability is insufficient, thereby impacting solution accuracy. To address this limitation, researchers have proposed numerous novel variants of MFO. In [56], a generalized opposition-based MFO with a crossover operation was introduced to overcome the deficiency in the basic MFO's development capability. The generalized opposition operation is responsible for enhancing population diversity and expanding the search space, while the crossover mechanism is employed to improve the algorithm's ability to fine-tune established regions. The proposed algorithm was tested using the CEC 2017 benchmark test suite and a multi-threshold image segmentation problem. Experimental results demonstrated that the method outperformed the basic MFO algorithm in terms of convergence speed and solution accuracy. In [57], an enhanced MFO algorithm, termed MFO-SFR, was developed. MFO-SFR introduces a stagnation search and replace strategy to enrich population diversity and consequently enhance the algorithm's search capability. Performance testing of the MFO-SFR algorithm was conducted using the CEC 2018 and CEC 2020 benchmark test suites, with results indicating its superiority over advanced MFO variants in solving global optimization problems. In [58], a novel MFO variant named WF-MFO was introduced by incorporating Fibonacci search process-based techniques and Levy flight mechanism into the basic MFO to overcome the weak development capability of the basic MFO. The algorithm's performance was tested using 29 benchmark test functions and 2 engineering design problems. Test results revealed that WF-MFO's performance significantly surpassed many popular algorithms and MFO variants.

Many scholars believe that no single metaheuristic technique can be universally suitable for a wide range of optimization problems. Therefore, combining one algorithm with others possessing complementary characteristics can effectively enhance operability while improving search capabilities. The primary objective of hybrid metaheuristic algorithms is to capitalize on the strengths of different algorithms to compensate for individual algorithm weaknesses, ultimately forming a high-performance algorithm. Some of these studies will be discussed in the next section.

Based on the above discussion, the EO algorithm and MFO algorithm are two popular metaheuristic algorithms, both achieving significant success. However, both algorithms have certain drawbacks. Upon closer examination, it can be observed that EO and MFO are two algorithmic approaches with complementary strengths; specifically, the EO algorithm exhibits strong exploration capabilities and weak exploitation capabilities, while MFO exhibits the opposite characteristics. This implies that combining these two algorithms can eliminate the shortcomings of individual algorithms, resulting in a potentially powerful hybrid algorithm. Inspired by this idea, this work proposes an ensemble algorithm called HMEO. The proposed HMEO is based on the exploitation characteristics of EO and the exploration model of MFO to form a new hybrid concept. The suggested algorithm preserves the overall framework and search pattern of EO while incorporating the search equations of MFO into the concentration updating process. By blending these two algorithms, leveraging the strengths of EO in exploration and MFO in exploitation, it is expected to achieve a better algorithm with appropriate exploration and exploitation operations. Herein, the CEC 2017 benchmark test suite is chosen to assess the performance of HMEO, considering various advanced metaheuristics, including basic EO, basic MFO, and advanced variants of EO and MFO. The effectiveness of the proposed HMEO is observed through convergence analysis and statistical tests. Additionally, the projected HMEO evaluates the MRPP problem and is compared with classical

metaheuristic-based MRPP methods to validate the effectiveness of the new approach.

This paper's contribution can be summarized by the following points:

1. A hybrid algorithm that combines EO and MFO, called HMEO, was proposed for global optimization.
2. The HMEO was proposed as an alternative MRPP methodology.
3. The performance of the introduced HMEO was validated by comparing it with several well-performing optimizers.
4. The experimental and simulation outcomes validate the efficiency of the HMEO.

The rest of the paper is structured as follows: Section 2 introduces some of the work conducted by various researchers in the field of hybrid metaheuristic algorithms. Background on EO and MFO are discussed in Section 3. The hybrid EO and MFO (HMEO) is discussed in Section 4. Experimental and result are reported in Section 5. Section 6 describes the proposed HMEO-based MRPP method. The summary of the proposed HMEO along with the drawbacks and profound impacts is summarized in detail in Section 7. Conclusion of the research is provided in Section 8.

2 Literature review

The efficiency of the metaheuristic algorithms relies on achieving a balance between their local exploitation capabilities and global exploration capabilities during the search process. Exploitation involves extracting new solutions from the neighborhood of existing solutions within the search space, while exploration refers to the process of discovering new regions in the search space [59]. Hybrid metaheuristic algorithms have been proposed as a solution to achieve a balance between exploitation and exploration. These algorithms integrate two or more metaheuristics to solve optimization problems, aiming to leverage the strengths of each algorithm while mitigating their respective weaknesses. As mentioned earlier, certain metaheuristics excel in exploitation, while others are more proficient in exploration. Hybrid metaheuristic algorithms are designed to strike a balance between these two phases. Additionally, in metaheuristic algorithms, there is an important trade-off between convergence and accuracy [60]. A hybrid metaheuristic algorithm should aim to maintain a good balance between achieving accurate solutions and preserving diversity within the population, which helps prevent premature convergence. This trade-off between convergence and accuracy is crucial in the design and implementation of hybrid metaheuristic algorithms.

In [61], a hybrid version of MPA and NMRA is proposed to overcome the poor exploitation of MPA and the limited exploration of NMRA. The performance of the proposed algorithm was tested on CEC 2005, CEC 2014 and CEC 2019 benchmark function problems. The experimental results show that the proposed hybrid algorithm outperforms MPA, NMRA and other comparative algorithms. In [62], a hybrid algorithm based on PSO and WOA was developed to improve the shortcomings of the basic algorithm. The results on 23 benchmark test functions and engineering design cases show that the proposed hybrid version outperforms both WOA and PSO. In [63], a novel hybrid algorithm is proposed by combining Biogeography-Based Optimization (BBO) with SSA. In the proposed algorithm, BBO exhibits a strong exploration capability, while SSA is employed for fine-tuning the search space. The hybrid algorithm is tested on the CEC 2014 benchmark suite and 10 real-world problems and compared with the basic BBO, SSA and various state-of-the-art algorithms. The comparison results highlight the effectiveness of the hybrid algorithm. In [64], SMA and teaching-learning based optimization (TLBO) were mixed to overcome the low solution accuracy and premature convergence drawbacks exhibited by the SMA algorithm in complex problems. The introduced algorithm is validated on 24 benchmark test functions and five reliability-based design optimization problems and compared with several state-of-the-art metaheuristics. The test results show that the proposed algorithm provides competitive performance on almost all problems. In [65], a hybrid approach called GBOSMA is proposed by combining GBO and SMA, which utilizes the advantageous factors of Gradient-Based Optimizer (GBO) and SMA. In the hybrid algorithm, SMA is used for local exploitation of GBO

to discover more unknown regions in the search space. The GBOSMA algorithm is used to solve CEC 2017 benchmark functions and feature selection problems and is compared with the most recent optimization algorithms. The results show that GBOSMA outperforms its competitors for both types of problems. In [66], reformed Backtracking Search Algorithm (BSA) was used to address the imbalance between exploration and exploitation capabilities of the PSO algorithm. The reported technique integrates the operational potential of PSO and the global search advantages of BSA with a view to providing an optimizer with a good balance between exploitation and exploration. CEC 2014 and CEC 2017 benchmark test sets were considered to evaluate the performance of the algorithms. In addition, 11 engineering design cases were used to investigate the applicability of the algorithms. The obtained results were compared with a variety of popular PSO and BSA variants. The analysis of the results shows that the proposed algorithm outperforms the comparison algorithm on almost all cases. In [67], a hybrid (arithmetic optimization algorithm) AOA algorithm, called AOA-NM, was proposed to facilitate the exploration-exploitation operation of the AOA algorithm. In AOA-NM, the local search behavior of Nelder-Mead is embedded in AOA to overcome the deficiency of local optimal stagnation of the algorithm. CEC 2020 test set and 10 engineering design cases are used to investigate the effectiveness of AOA-NM, and the test results show that it is robust in handling challenging optimization problems. In [68], WOA and Sine Cosine Algorithm (SCA) are combined to form a new well-performing algorithm called WOASCALF. The main idea of WOASCALF is that WOA has superior performance in the exploration phase, while SCA is seasoned in the development phase. Combining the advantages of both algorithms to compensate for their respective shortcomings can enhance the algorithm's capabilities. In addition, an additional levy flight mechanism is introduced to further improve the local search capability of the algorithm. the performance of WOASCALF is evaluated by 23 widely used benchmark test functions and some engineering design tasks. The results analysis shows that WOASCALF has a strong search capability. In [69], the operators of the SCA algorithm were used to enhance the development-exploration balance of the AOA algorithm. To investigate the efficiency of the developed method, 10 benchmark test functions and five engineering optimization cases were used. The experimental results show that the developed enhanced AOA algorithm is more efficient than the comparison algorithms. In [70], a novel hybrid algorithm was introduced that integrates MFO and Butterfly Optimization Algorithm (BOA), overcoming the shortcomings of poor convergence of MFO and inadequate exploitation capability of BOA and inheriting their advantages. To evaluate the algorithm performance, 23 benchmark test functions with different characteristics and two real-life problems are used. Experimental results show that the hybrid algorithm provides competitive performance compared to the comparison algorithm. In [71], a hybrid optimization algorithm based on PSO and BSA is proposed, which takes advantage of the high convergence rate of PSO and the strong global search capability of BSA, while overcoming the drawbacks of the loss of population diversity of PSO and the limitations of the poor local exploration capability of BSA. Results on several benchmark problems show that the algorithm outperforms the compared PSO variant and BSA variant. In [72], a new hybrid algorithm, called HSSOGSA, was proposed. The basic concept behind HSSOGSA is to leverage the strong exploitation capability of SSO and the strong exploration capability of GSA to achieve an excellent optimization algorithm. To verify the performance of the HSSOGSA algorithm, the CEC 2017 test suite was used. The experimental results show that the proposed hybrid algorithm has better convergence performance compared to the basic SSO and GSA. In [73], an EO-based hybrid WOA variant was developed. Upon mathematical modeling, the search models of WOA and EO are combined to form an effective hybrid algorithm. The performance of the algorithm is evaluated on several classical function optimization problems. Experimental results show that the algorithm has good convergence performance.

Based on the above studies, combining two algorithms with complementary features can effectively enhance algorithm performance. To the best of our knowledge, there is currently no research utilizing a hybrid approach to integrate EO with other population-based algorithms to enhance its search process. Therefore, this paper introduces, for the first time, a hybrid MFO and EO algorithm to com-

pensate for the deficiencies of both, forming an efficient algorithm.

3 Background

The mathematical notation of the EO and MFO are discussed in this section.

3.1 Equilibrium optimizer

EO is an innovative metaheuristic approach developed for addressing global optimization problems. The algorithm draws inspiration from the dynamic mass balance law in physics [44]. In EO, search agents are represented by particles with varying concentrations, where the concentration of each particle represents a solution to the optimization problem. The algorithm employs the mass balance equation to analyze the concentration of non-reactive components within the control volume, reflecting the physical phenomena of mass entry, departure, and generation within the volume. By striving to stabilize the system's concentration, the optimization algorithm mimics the process through which the system aims to attain equilibrium. Consequently, this iterative process represents the algorithm's endeavor to converge towards the global optimum.

Similar to its counterparts in metaheuristic algorithms, EO initiates the search procedure by generating a set of initial particles in a random manner. The concentrations of these particles are formulated based on the following equation.

$$C_i^{initial} = C_{min} + rand_i(C_{max} - C_{min}) i = 1, 2, ..., N \tag{1}$$

where $C_i^{initial}$ indicate the concentration of the ith particle, $rand_i$ is a random vector within 0 and 1, C_{min} and C_{max} denote the upper and lower bound of the search space, respectively, N represents the population size.

Following the initialization of particles using Eq. (1), their concentrations are subsequently updated using Eq. (2).

$$C = C_{eq} + (C - C_{eq}) \cdot F + \frac{G}{\lambda V}(1 - F) \tag{2}$$

where C_{eq} is the concentration of particles in the equilibrium pool. The equilibrium pool comprises four equilibrium candidates, namely $C_{eq(1)}$, $C_{eq(2)}$, $C_{eq(3)}$, $C_{eq(4)}$, which have the best concentrations observed so far. The average concentration of these candidates, denoted as $C_{eq(ave)}$, is also determined. To address the lack of initial information about the equilibrium state during the optimization process, the particles within the equilibrium pool play a crucial role in bridging this knowledge gap. They offer a concentration update pattern for the particles involved. The equilibrium pool is constructed as follows:

$$C_{eq,pool} = \left\{ C_{eq(1)}, C_{eq(2)}, C_{eq(3)}, C_{eq(4)}, C_{eq(ave)} \right\} \tag{3}$$

In Eq. (2), an adjustment factor is introduced, denoted as F, which is an exponential term responsible for influencing the exploitation and exploration behavior of the algorithm. The mathematical representation of F is given by the following expression:

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

In Eq. (2), the function denoted as t is dependent on the number of iterations and plays a crucial role in the algorithm. It is defined as follows:

$$t = \left(1 - \frac{Iter}{Max_iter}\right)^{\left(a_2 \frac{Iter}{Max_iter}\right)} \tag{5}$$

where $Iter$ and Max_iter denote the present generation and the maximum generation, respectively. Additionally, the algorithm incorporates a constant, a_2, which governs the local search capability. To achieve a balanced trade-off between convergence speed and solution accuracy, the algorithm takes into account the following factors:

$$t_0 = \frac{1}{\lambda} \ln\left(-a_1 sign\,(r - 0.5)\left(1 - e^{\lambda t}\right)\right) + t \tag{6}$$

where a_1 is the parameter used to adjust the global search behavior. A higher value of a_1 emphasizes the algorithm's global exploration capability, while a lower value of a_1 results in weaker exploration. Similarly, a_2 governs the exploitation operations. A larger a_2 enhances the algorithm's ability to explore local regions, but weakens its exploration ability. In

this context, a_1 is set to 0 and a_2 is set to 1. Additionally, the variable r represents a random vector within 0 and 1, and sign(r-0.5) determines the searching direction. By substituting Eq. (6) into Eq. (4), we obtain a revised version of the expression for F as follows:

$$F = a_1 sign\,(r-0.5)\left[e^{-\lambda t}-1\right] \tag{7}$$

The development phase of the algorithm places great importance on the generation rate, denoted as G, as it plays a key role in enhancing the solution accuracy by fine-tuning the promising search area. The value of this factor is computed using the following calculation:

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

where

$$G_0 = GCP\,(C_{eq} - \lambda C) \tag{9}$$

$$GCP = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \tag{10}$$

where r_1 and r_2 are random numbers within 0 and 1, The generation rate control factor, known as GCP, determines the probability of the generation rate's involvement in the concentration update process. The generation rate control factor, known as GCP, determines the probability of the generation rate's involvement in the concentration update process. On the other hand, the generation probability, referred to as GP, determines which particle is eligible to update the concentration using G.

According to the concentration update equation Eq. (2), where the first term corresponds to the equilibrium candidate that identifies the desired equilibrium state. The second and third terms denote the concentration variables. The second term directly captures the disparity between the current particle and the equilibrium particle, leading to the generation of significant concentration variables during the optimization process. Consequently, this term plays a crucial role in facilitating exploration. On the other hand, the third term, as inferred from its expression, is indirectly associated with the distinction between the sample particle and the equilibrium particle. This term contributes to the development operation. The pseudo-code of the EO algorithm is described in Algorithm 1, and the flowchart is shown in Figure 1.

Algorithm 1 Pseudo-code of EO

Input: Maximum generation Max_iter, upper bound C_{max} and lower bound C_{min}, randomly generated initial population $C_i^{initial}$, i=1,2,...,N and the dimension D

Output: The concentration of equilibrium candidate solution $C_{eq(1)}$

```
Initialize the population C_i^initial (i = 1,2,...,N) by considering C_max and C_min
Evaluate the fitness value of each particle
while Iter < Max_iter do
    Establish the equilibrium pool
    Implement memory saving (if Iter >1)
    for i = 1 to N do
        Randomly select a particle from the equilibrium pool
        Calculate the exponential term F according to Eq. (7)
        Calculate the generation rate G according to Eq. (8)
        Update the concentration of the ith particle according to Eq. (2)
    end for
    Iter=Iter+1
end while
Return the equilibrium candidate C_eq(1)
```

3.2 Moth flame optimization algorithm

Nocturnal moths possess the ability to navigate accurately over wide distances by using celestial light sources. They maintain a straight flight path with a consistent angle to the light source, and this maneuver is known as lateral orientation. However, when a moth encounters an artificial light source during its flight, it will be misdirected. As a result, the moth needs to persistently adjust its movement angle to maintain a direct course toward the light source, causing a spiral motion around the artificial light source. This intriguing behavior was mathematically modeled in 2015 by Mirjalili, who developed the MFO algorithm for solving global optimization problems [23]. The details of this algorithm are described in the following sections. In MFO, the moth's position undergoes constant updates throughout the optimization process. In the

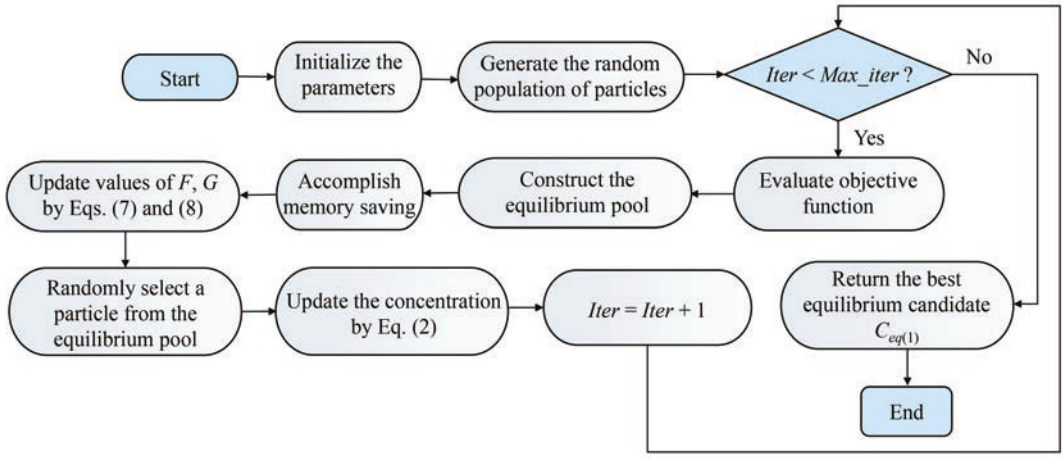

Figure 1. The flowchart of EO

initial stage, the particles are randomly distributed in the search space, that is, the search procedure of MFO commences by randomly generating a set of initial particles. The mathematical expression for this phase is shown below.

$$C_i^{initial} = C_{min} + rand_i(C_{max} - C_{min}) i = 1, 2, ..., N \tag{11}$$

For subsequent iterations, the moth updates its position according to the position of the flame. To calculate the number of flames, the following equation is employed.

$$R = round\left(N - Iter \cdot \frac{N - Iter}{Max_iter}\right) \tag{12}$$

where N denote the population size, $Iter$ and Max_iter demonstrate the present generation and the maximum generation, respectively.

After calculating the number of flames R, the moth updates the position according to Eq. (13). According to Eq. (13), there are two different trials of the position update model considering the number of flames, and the mathematical expression is illustrated as follows.

$$C_i = \begin{cases} D_i \cdot e^{bt} \cdot cos(2\pi t) + F_i & i \leq R \\ D_i \cdot e^{bt} \cdot cos(2\pi t) + F_R & i > R \end{cases} \tag{13}$$

where D_i represents the distance between the ith seed and its target flame (F_j), parameter b is used to control the search shape of the spiral shape flight path, t represents a random number ranging between -1 and 1, signifying the proximity of the moth to its respective flame. To calculate the distance D_i, the following equation is employed.

$$D_i = |F_i - C_i| \tag{14}$$

Regarding the value of t, a tunable scheme is used to dynamically decay its value during the iterative process, thus switching from exploration to exploitation in the early to late search period. To calculate the value of t, the following equation is applied.

$$t = (r - 1) \cdot rand + 1 \tag{15}$$

$$r = -1 + Iter \cdot \left(\frac{-1}{Max_iter}\right) \tag{16}$$

In the second case (when $i > R$), the distance D_i is calculated using Eq. (17), and F_R is the position of the Rth flame.

$$D_i = |F_R - C_i| \tag{17}$$

The pseudo-code of MFO is described in Algorithm 2, and the flowchart is shown in Figure 2.

4 The Proposed HMEO

The main motto of any nature-inspired algorithm is to maintain a balance between the devel-

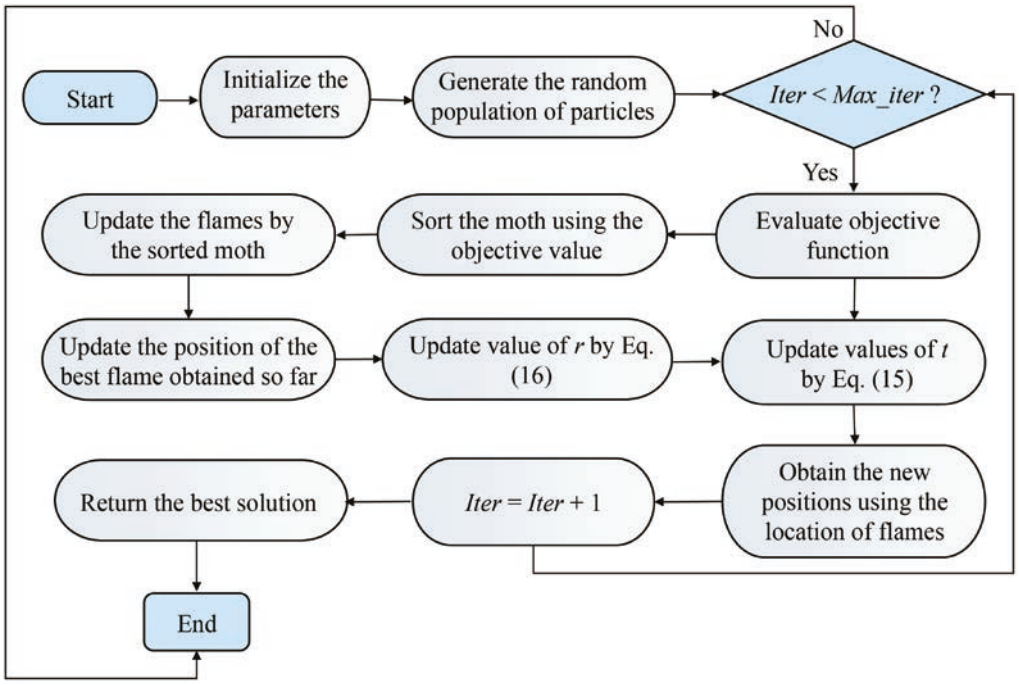

Figure 2. The flowchart of MFO

opment and exploration phases. Obviously, excessive exploration means that the algorithm spends too much effort on unpromising regions, which is the main reason why the algorithm misses the desired solution. On the other hand, too much exploitation means that the algorithm devotes considerable attention to monotonous regions, which leads to loss of population diversity and premature convergence of the algorithm. Therefore, the preferred performance of any algorithm is obtained when it captures a proper balance between diversification and intensification.

The EO algorithm possesses strong exploitation capability and weak exploration capability. The main reason is that the items governing the exploration operation are a_1 and *GP*. The parameter a_1 magnifies the concentration difference between the current seeds and equilibrium candidates, and it should be sufficiently large to facilitate significant concentration updates and improve exploration capability.

However, excessively large values can lead to ineffective boundary searching by particles, indicating that modifying a_1 to overcome the lack of exploration capability is unwise. On the other hand, the value of *GP* controls the probability of the third term in the concentration update rule participating in the concentration update process. A higher *GP* value provides strong exploitation capability and weak exploration capability, while a lower *GP* value signifies weaker exploitation and stronger exploration. It is evident that determining the optimal *GP* value relies on empirical testing, and attempting to enhance exploration capability by adjusting the *GP* value is not advisable. Therefore, it is cumbersome to enhance the exploration capability of the EO algorithm by adjusting parameters or modifying the concentration update equation. This implies that reinforcing the global search capability of EO is a challenging task.

In contrast, MFO exhibits strong global search capability. MFO is a bio-inspired optimization algorithm that introduces the crucial mechanism of moths spiraling around flames, offering innovative search patterns. The exploration and exploitation operations in MFO are derived from this spiral motion. In MFO, moths follow a spiral trajectory around their flames and update the flames based on the survival-of-the-fittest mechanism, retaining high-quality flames. This mechanism empowers the MFO algorithm with strong global search capability but can also result in moths getting trapped in local optima.

Algorithm 2 Pseudo-code of MFO

Input: Maximum generation Max_iter, upper bound C_{max} and lower bound C_{min}, randomly generated initial population $C_i^{initial}$, i=1,2,...,N, flame number, and the dimension D

Output: The location of the optimal flame.

```
Initialize the population C_i^initial (i = 1,2,...,N) by considering C_max and C_min
while Iter < Max_iter do
    Calculate the number of flames according to Eq. (12)
    Evaluate the fitness value of each moth
    if Iter==1 then
        Sort the first population of moths according to the fitness value
        Update the flames
    else
        Sort the moths according to the fitness value
        Update the flames
    end if
    Update the position of the best flame obtained so far
    Calculate the parameter r according to Eq. (16)
    for i = 1 to N do
        for j = 1 to D do
            Calculate the parameter t accorting to Eq. (15)
            Calculate the distance D_i according to Eq.(14) or Eq. (17)
            Update moths position as to their particular flame using Eq. (13)
        end for
    end for
    Iter=Iter+1
end while
Return the best solution with the minimum fitness function value in the ecosystem
```

Therefore, to avoid the conflict between these two algorithms and be attracted by the compatibility of EO and MFO, we try to combine these two algorithms by embedding the spiral search mechanism of MFO into the EO algorithm to build a new well-performing hybrid algorithm, called HMEO. The goal of this algorithm is to maximize the strong exploitation capability of the EO algorithm and the rich exploration capability of the MFO algorithm, while minimizing the shortcomings of the weak global search operation of EO and the poor local search ability of MFO through the hybrid. Overall, the proposed HMEO algorithm has both swarm intelligence and physics-based properties, where the swarm intelligence mechanism provides diversity for superior global search and the physics-based operation provides intensification for better local development.

The introduced HMEO algorithm is mainly formulated based on the search framework of EO and the spiral motion of MFO. In general, the search framework of EO is the foundation, while the behavioral mechanism of MFO is the merged material. In the HMEO algorithm, we also start the algorithm like EO and then apply the search mechanism of the EO algorithm for position updating. Subsequently, the particles search in the solution space using the spiral search mechanism of MFO. The flowchart of the proposed HMEO algorithm is shown in Figure 3. The pseudo-code of the HMEO algorithm is described in Algorithm 3, and it can be described through a series of steps as outlined below:

Step 1: Initialize all parameters such as the maximum number of iterations, the number of populations, the dimensionality of the problem, and the search boundaries.

Step 2: Initialize the population randomly by considering the search boundaries.

Step 3: Construct the equilibrium pool and save memory.

Step 4: Calculate the exponential term F and the generation rate G according to Eqs. (7) and (8), respectively.

Step 5: Use Eq. (2) to update the concentration of the ith particle, and the updated concentration is noted as $F1$.

Step 6: Calculate the parameters r and t according to Eqs. (15) and (16). Then, use Eq. (14) to calculate the distance between the ith particle and the equilibrium candidate individual $C_{eq(1)}$.

Step 7: Use Eq. (13) to update the concentration of the ith particle, and the updated concentration is noted as $F2$.

Step 8: Compare $F1$ and $F2$ and select the best.

Step 9: Proceed to Step 2 and repeat until the stopping criteria are met.

The computational complexity of HMEO primarily relies on key parameters, including the maximum iteration count (T), the population size (N), the problem's input dimensions (D), and the function evaluation cost (f). The foundational aspects governing the computational complexity in HMEO encompass:

$$\begin{aligned} O(\text{HMEO}) &= O(Initialization) \\ &+ O(T(Functionevaluations)) \\ &+ O(T(Memorysaving)) \\ &+ O(T(Concentrationupdate)) \end{aligned} \tag{18}$$

As a result, the overall computational complexity of HMEO can be expressed in the following manner:

$$\begin{aligned} O(\text{HMEO}) &= O(ND + TfN + TN + 2TND) \\ &\cong O(TND + TfN) \end{aligned} \tag{19}$$

5 Simulation Results and Discussion

In this section, extensive experiments are conducted to thoroughly evaluate the performance of the newly proposed HMEO algorithm, alongside a comparison with several state-of-the-art evolutionary algorithms. Further details regarding the experiments will be provided in the subsequent subsections.

5.1 Benchmark functions

To thoroughly evaluate the performance of the proposed HMEO algorithm, an extensive examination was conducted using the CEC 2017 benchmark test suite. This test suite consists of 29 functions with diverse characteristics, widely employed for assessing the effectiveness of optimization algorithms. The benchmark functions are divided into two categories: continuous single-objective optimization problems and expensive optimization problems. The continuous single-objective optimization problems encompass classical benchmark, combinatorial, and hybrid functions. On the other hand, the expensive optimization problems simulate real-world black-box optimization problems commonly encountered. The 29 benchmark functions are further categorized into unimodal, multimodal, hybrid, and combinatorial functions. The details of the 29 functions from CEC 2017 benchmark collection utilized in this effort are presented in Table 1. By analyzing the algorithm's performance on these functions, a comprehensive evaluation of the algorithm's efficacy can be achieved.

Algorithm 3 Pseudo-code of HMEO

Input: Maximum generation Max_iter, upper bound C_{max} and lower bound C_{min}, randomly generated initial population $C_i^{initial}$, i=1,2,...,N and the dimension D

Output: The concentration of equilibrium candidate solution $C_{eq(1)}$

Initialize the population $C_i^{initial}$ $(i = 1,2,\ldots,N)$ by considering C_{max} and C_{min}
Evaluate the fitness value of each particle
while $Iter < Max_iter$ **do**
  Establish the equilibrium pool
  Implement memory saving (if $Iter > 1$)
  for $i = 1$ to N **do**
    Randomly select a particle from the equilibrium pool
    Calculate the exponential term F according to Eq. (7)
    Calculate the generation rate G according to Eq. (8)
    Update the concentration of the ith particle according to Eq. (2), and the updated concentration is noted as $F1$
    Calculate the parameter r according to Eq. (16)
    Calculate the parameter t accorting to Eq. (15)
    Calculate the distance between the ith particle and the equilibrium candidate solution $C_{eq(1)}$
    Update the concentration of the ith particle according to Eq. (13), and the updated concentration is noted as $F2$
    Compare $F1$ and $F2$ and select the best.
  end for
  $Iter$=$Iter$+1
end while
Return the equilibrium candidate $C_{eq(1)}$

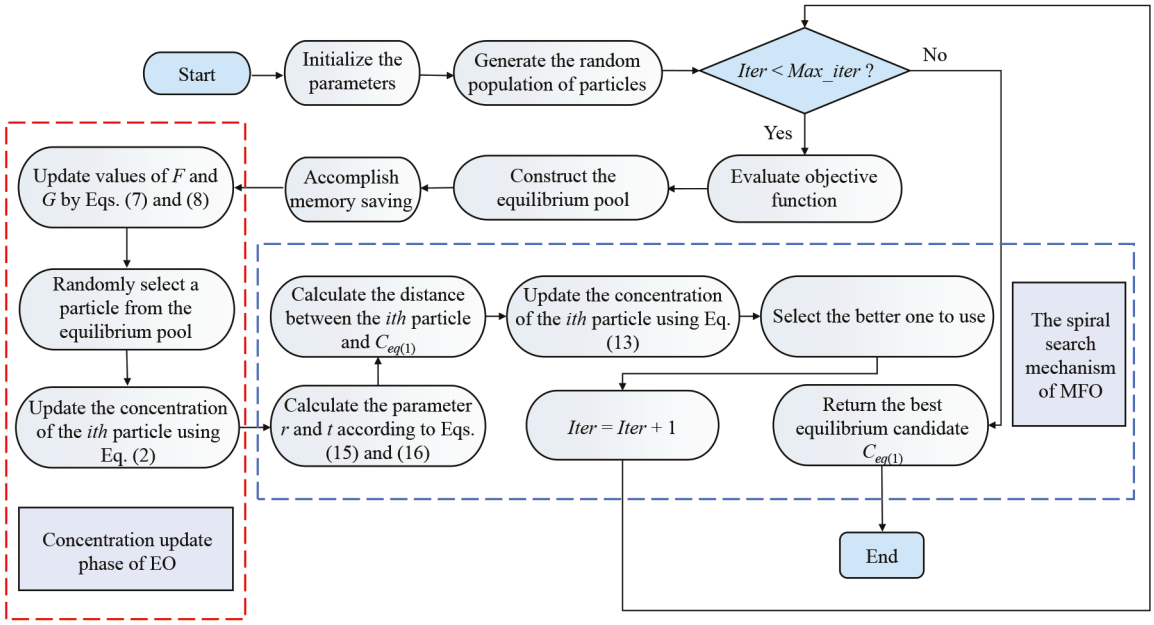

Figure 3. The flowchart of HMEO

5.2 Experimental setup

To assess the performance of the proposed HMEO, it is compared with the traditional EO, MFO, as well as some advanced EO and MFO variants. The proposed hybrid version based on EO and MFO does not introduce any additional parameters, and the parameters of the individual algorithms remain unchanged from their basic versions. The specific parameters for the compared algorithms are obtained from the respective literature sources. For result analysis, the maximum number of iterations is set to 500, the population size is set to 30, and each algorithm is independently run 30 times on each test function. Evaluation metrics such as mean and standard deviation are recorded for the results. The source code for all algorithms is implemented on the MATLAB R2016b software running on the Windows 10 operating system. The computations are carried out on a computer equipped with an Intel Core i7-7700 CPU @ 3.60 GHz and 8.00 GB of RAM.

5.3 Comparison of HMEO with other well-known metaheuristic algorithms

In this section, we evaluate the performance of the introduced HMEO algorithm using the 29 benchmark functions from CEC 2017, as outlined in Table 1. It is important to note that F2 is excluded because it exhibits unstable behavior. The results of HMEO are compared with the MFO [23], EO [44], mEO [74], LWMEO [75], ISEO [76], IEO [77], WEMFO [78], and DMMFO [79] algorithms. The comparison results, focusing on a dimensionality of 30, are presented in Table 2. Additionally, when the dimensionality is increased to 100, the corresponding comparison results are shown in Table 3.

Based on the reported experimental results in Table 2, it can be observed that the HMEO algorithm achieves the best efficiency in 16 test functions and the second-best efficiency in 10 test functions. This indicates that HMEO obtains a top-two ranking in nearly all functions. Based on pairwise comparisons, HMEO outperforms ISEO, MFO, WEMFO, and DMMFO across all functions. Particularly, HMEO performs better than EO in 27 out of 29 functions, except for F17 and F29 where it shows inferior performance. In terms of comparison with IEO, HMEO surpasses it in 19 cases but falls behind in 10 cases. When comparing with LWMEO, HMEO outperforms it in 25 functions, except for F1, F3, F4, and F13 where LWMEO shows better results. Additionally, HMEO outperforms mEO in all functions except for F17. These findings highlight the effectiveness of the HMEO algorithm in solving the optimization problems. It exhibits superior performance in terms of conver-

gence and solution quality compared to EO, MFO, and the compared EO and MFO variants. However, there are specific cases, such as F1 and F13, where alternative algorithms like LWMEO and IEO may offer better results. Furthermore, Figure 4 displays the Friedman mean ranks attained by the utilized approaches across these functions. According to Figure 4, HMEO obtains the highest rank, followed by IEO, EO, mEO, LWMEO, DMMFO, WEMFO, ISEO, and MFO. Thus, HMEO emerges as the algorithm with the best performance among all the implemented algorithms based on the achieved Friedman mean ranks.

From Table 3, The HMEO algorithm achieves the first rank in 16 tested functions, the second rank in 9 functions, and the third rank in four functions. This indicates that HMEO obtains a top-three performance across all functions. Based on the extensive evaluation presented in Table 3, the HMEO algorithm consistently demonstrates remarkable performance across a wide range of tested functions. It significantly outperforms EO in the majority of cases, reaffirming its superior optimization capabilities. With the exception of F19, HMEO surpasses mEO, MFO, and DMMFO on all evaluated functions, highlighting its effectiveness on 100-dimensional optimization problems. Furthermore, the comparative analysis reveals that HMEO achieves superiority over IEO in a significant number of cases while slightly lagging behind in some instances. Notably, HMEO outperforms WEMFO on almost all functions, except for F7, suggesting its robustness and competitiveness in solving complex optimization problems. Moreover, HMEO's performance shines in comparison to ISEO, as it outperforms ISEO on most of test functions, demonstrating its capability to provide more favorable optimization results. Despite one function where it falls behind, HMEO consistently exhibits strong performance in terms of overall rankings. In comparison to LWMEO, HMEO exhibits better performance on most functions. However, it shows comparatively weaker performance on some functions, such as F11, F13, F15, and F18, where LWMEO and IEO prove to be more competitive. Figure 5 displays the Friedman mean ranks attained by the utilized approaches across these functions. The average rankings further reinforce HMEO's superiority, as it secures the top position among all evaluated algorithms. This indicates that HMEO consistently outperforms other algorithms in terms of overall optimization performance, providing a highly effective and reliable solution to a wide range of optimization problems.

In summary, the comprehensive experimental analysis demonstrates that HMEO is a robust and efficient algorithm, surpassing various state-of-the-art optimization methods in terms of performance. Its remarkable performance across different dimensions confirms its versatility and reliability, making HMEO a highly promising algorithm for solving complex optimization problems.

6 Architecture of MRPP Using HMEO

To further challenge the HMEO algorithm, which is employed for solving the MRPP problem, it is essential to recognize the significance of studying MRPP techniques, as they constitute a core technology within robotics and hold crucial practical and theoretical implications [80]. Firstly, researching the MRPP problem aids in designing optimal paths for robots, enabling them to navigate around obstacles and adopt safe and efficient navigation strategies. This enhances the flexibility and reliability of robots in executing their assigned tasks. Secondly, effective path planning allows mobile robots to select the shortest or optimal routes, minimizing unnecessary movement and energy consumption. Consequently, this optimizes resource utilization and energy efficiency, thereby extending operational time, improving efficiency, and reducing maintenance costs. Additionally, path planning technology contributes to obstacle avoidance, collision prevention, and overall operational safety for mobile robots. Precise path planning reduces the likelihood of accidents, ensuring the protection of the robot itself and its surrounding environment. Importantly, path planning technology for mobile robots is a key aspect of robotics, and its exploration plays a pivotal role in promoting robotics development and innovation [81]. Continuous improvement and optimization of path planning algorithms and methods enhance the navigation capabilities and intelligence of robots, further facilitating the widespread application and adoption of robotics. Therefore, the study of path planning technology for mobile robots encompasses signif-

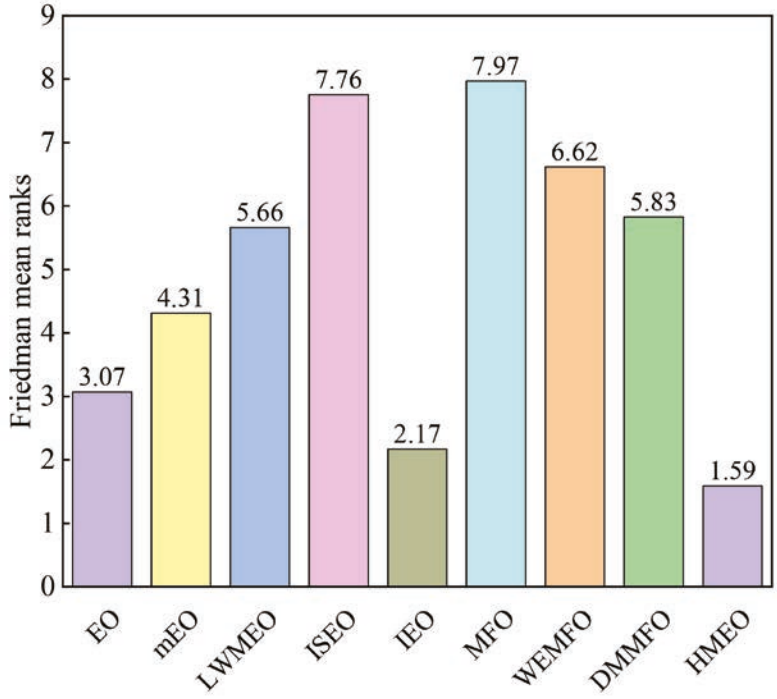

Figure 4. Friedman mean ranks obtained by the employed algorithms on CEC 2017 benchmark functions with 30 dimensions

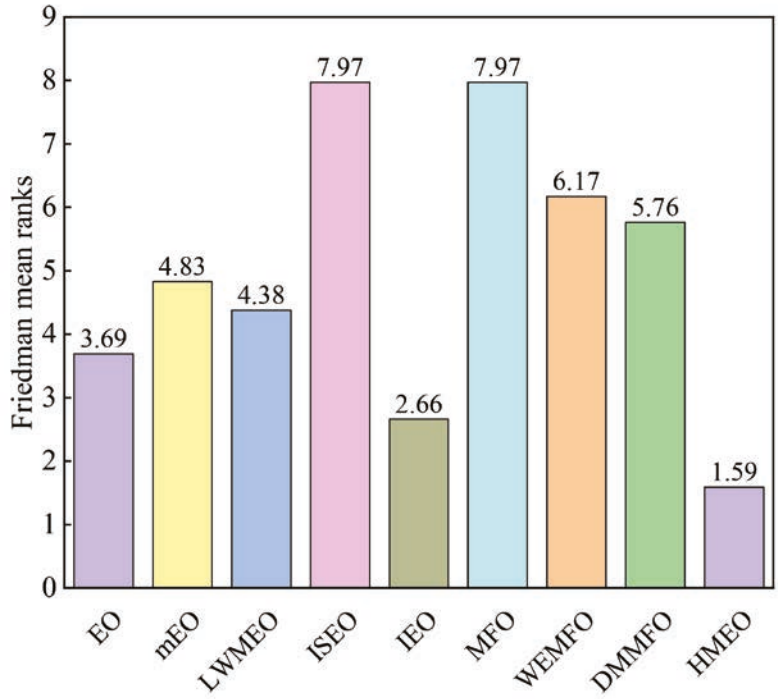

Figure 5. Friedman mean ranks obtained by the employed algorithms on CEC 2017 benchmark functions with 100 dimensions

Table 1. Summary of the 30 CEC 2017 benchmark problems

Class	No.	Description	Search Range	Optimal
Unimodal	F1	Shifted and Rotated Bent Cigar Function	[-100, 100]	100
	F2	Shifted and Rotated Sum of Different Power Function	[-100, 100]	200
	F3	Shifted and Rotated Zakharov Function	[-100, 100]	300
Multimodal	F4	Shifted and Rotated Rosenbrock's Function	[-100, 100]	400
	F5	Shifted and Rotated Rastrigin's Function	[-100, 100]	500
	F6	Shifted and Rotated Expanded Scaffer's Function	[-100, 100]	600
	F7	Shifted and Rotated Lunacek Bi-Rastrigin Function	[-100, 100]	700
	F8	Shifted and Rotated Non-Continuous Rastrigin's Function	[-100, 100]	800
	F9	Shifted and Rotated Levy Function	[-100, 100]	900
	F10	Shifted and Rotated Schwefel's Function	[-100, 100]	1000
Hybrid	F11	Hybrid Function 1 (N = 3)	[-100, 100]	1100
	F12	Hybrid Function 2 (N = 3)	[-100, 100]	1200
	F13	Hybrid Function 3 (N = 3)	[-100, 100]	1300
	F14	Hybrid Function 4 (N = 4)	[-100, 100]	1400
	F15	Hybrid Function 5 (N = 4)	[-100, 100]	1500
	F16	Hybrid Function 6 (N = 4)	[-100, 100]	1600
	F17	Hybrid Function 6 (N = 5)	[-100, 100]	1700
	F18	Hybrid Function 6 (N = 5)	[-100, 100]	1800
	F19	Hybrid Function 6 (N = 5)	[-100, 100]	1900
	F20	Hybrid Function 6 (N = 6)	[-100, 100]	2000
Composition	F21	Composition Function 1 (N=3)	[-100, 100]	2100
	F22	Composition Function 2 (N=3)	[-100, 100]	2200
	F23	Composition Function 3 (N=4)	[-100, 100]	2300
	F24	Composition Function 4 (N=4)	[-100, 100]	2400
	F25	Composition Function 5 (N=5)	[-100, 100]	2500
	F26	Composition Function 6 (N=5)	[-100, 100]	2600
	F27	Composition Function 7 (N=6)	[-100, 100]	2700
	F28	Composition Function 8 (N=6)	[-100, 100]	2800
	F29	Composition Function 9 (N=3)	[-100, 100]	2900
	F30	Composition Function 10 (N=3)	[-100, 100]	3000

Table 2. Comparisons of nine algorithms on CEC 2017 benchmark functions with 30 dimensions

Function	Results	EO	mEO	LWMEO	ISEO	IEO	MFO	WEMFO	DMMFO	HMEO
F1	Mean	5.82E+03	4.97E+06	5.40E+03	5.81E+09	5.31E+03	6.20E+09	2.83E+07	1.04E+08	5.75E+03
	Std	5.29E+03	5.96E+06	6.24E+03	1.42E+09	6.09E+03	6.68E+09	2.71E+07	2.63E+08	6.07E+03
	f-rank	4	5	2	8	1	9	6	7	3
F3	Mean	1.45E+04	1.38E+04	2.29E+03	8.31E+04	8.64E+03	1.09E+05	5.18E+04	1.31E+05	3.95E+03
	Std	5.13E+03	4.61E+03	1.78E+03	1.76E+04	3.29E+03	3.23E+04	8.85E+03	3.27E+04	2.13E+03
	f-rank	5	4	1	7	3	8	6	9	2
F4	Mean	5.03E+02	5.06E+02	4.82E+02	8.58E+02	4.93E+02	8.09E+02	5.36E+02	5.24E+02	4.92E+02
	Std	1.83E+01	1.88E+01	2.89E+01	1.55E+02	1.69E+01	3.52E+02	3.54E+01	2.51E+01	3.09E+01
	f-rank	4	5	1	9	3	8	7	6	2
F5	Mean	5.61E+02	5.87E+02	6.68E+02	7.13E+02	5.46E+02	6.73E+02	6.50E+02	6.28E+02	5.57E+02
	Std	1.61E+01	2.59E+01	3.88E+01	1.25E+01	1.17E+01	3.82E+01	4.46E+01	2.27E+01	1.87E+01
	f-rank	3	4	7	9	1	8	6	5	2
F6	Mean	6.00E+02	6.03E+02	6.34E+02	6.11E+02	6.00E+02	6.25E+02	6.16E+02	6.10E+02	6.00E+02
	Std	5.11E-02	1.22E+00	8.33E+00	1.93E+00	1.36E-02	8.47E+00	9.37E+00	6.32E+00	1.50E-02
	f-rank	3	4	9	6	1	8	7	5	2
F7	Mean	7.98E+02	8.25E+02	1.04E+03	1.12E+03	7.77E+02	1.05E+03	8.76E+02	8.61E+02	7.89E+02
	Std	2.43E+01	2.28E+01	8.00E+01	5.46E+01	1.50E+01	1.74E+02	4.48E+01	3.93E+01	1.65E+01
	f-rank	3	4	7	9	1	8	6	5	2
F8	Mean	8.59E+02	8.83E+02	9.38E+02	1.01E+03	8.47E+02	9.80E+02	9.54E+02	9.38E+02	8.59E+02
	Std	1.70E+01	2.10E+01	3.70E+01	1.57E+01	1.42E+01	4.43E+01	3.81E+01	3.51E+01	1.56E+01
	f-rank	3	4	6	9	1	8	7	5	2
F9	Mean	9.22E+02	1.07E+03	4.72E+03	1.50E+03	9.02E+02	5.83E+03	5.08E+03	2.83E+03	9.06E+02
	Std	4.57E+01	1.08E+02	1.12E+03	1.70E+02	1.63E+00	2.02E+03	2.52E+03	9.68E+02	5.02E+00
	f-rank	3	4	7	5	1	9	8	6	2
F10	Mean	4.60E+03	5.07E+03	5.37E+03	8.47E+03	4.49E+03	5.42E+03	5.13E+03	4.81E+03	4.08E+03
	Std	6.46E+02	8.57E+03	6.75E+02	3.05E+02	7.43E+02	7.47E+02	7.81E+02	6.62E+02	4.83E+02
	f-rank	3	5	7	9	2	8	6	4	1
F11	Mean	1.17E+03	1.26E+03	1.28E+03	1.62E+03	1.17E+03	2.09E+03	1.42E+03	1.81E+03	1.14E+03
	Std	3.63E+01	4.50E+01	6.46E+01	9.30E+01	4.28E+01	1.54E+03	1.53E+02	1.25E+03	3.38E+01
	f-rank	2	4	5	7	3	9	6	8	1
F12	Mean	7.25E+05	3.73E+06	1.99E+05	7.90E+07	1.89E+05	1.23E+08	1.89E+07	5.68E+06	1.48E+05
	Std	6.17E+05	2.90E+06	1.41E+05	3.05E+07	2.07E+05	3.49E+08	2.91E+07	6.03E+06	8.87E+04
	f-rank	4	5	3	8	2	9	7	6	1
F13	Mean	2.85E+04	1.14E+05	1.79E+04	4.36E+06	2.25E+04	3.23E+06	7.81E+05	5.01E+04	2.39E+04
	Std	2.00E+04	9.48E+04	1.50E+04	3.17E+06	1.85E+04	1.30E+07	1.84E+06	1.64E+05	2.07E+04
	f-rank	4	6	1	9	2	8	7	5	3
F14	Mean	3.76E+04	5.13E+04	1.62E+04	7.87E+04	1.84E+04	1.33E+05	1.60E+05	1.78E+05	1.45E+04
	Std	2.97E+04	4.30E+04	1.20E+04	4.19E+04	1.58E+04	1.08E+05	2.33E+05	2.78E+05	1.07E+04
	f-rank	4	5	2	6	3	7	8	9	1
F15	Mean	4.41E+03	1.21E+04	1.00E+04	8.18E+04	5.99E+03	4.28E+04	3.06E+04	1.68E+04	4.00E+03
	Std	2.83E+03	9.71E+03	1.27E+04	2.00E+05	5.73E+03	4.58E+04	2.47E+04	1.32E+04	4.20E+03
	f-rank	2	5	4	9	3	8	7	6	1
F16	Mean	2.35E+03	2.47E+03	2.96E+03	3.15E+03	2.20E+03	3.14E+03	2.77E+03	2.67E+03	2.35E+03
	Std	3.30E+02	3.23E+02	3.39E+02	2.04E+02	2.38E+02	4.02E+02	3.44E+02	3.42E+02	2.93E+02
	f-rank	3	4	7	9	1	8	6	5	2
F17	Mean	1.97E+03	1.98E+03	2.43E+03	2.22E+03	1.90E+03	2.39E+03	2.31E+03	2.18E+03	2.04E+03
	Std	1.60E+02	1.68E+02	3.03E+02	2.05E+02	1.07E+02	2.47E+02	2.14E+02	1.91E+02	1.85E+02
	f-rank	2	3	9	6	1	8	7	5	4
F18	Mean	4.96E+05	6.34E+05	1.64E+05	4.33E+06	2.68E+05	3.66E+06	2.21E+06	3.75E+06	1.47E+05
	Std	5.61E+05	5.15E+05	9.79E+04	2.49E+06	1.77E+05	5.51E+06	1.85E+06	4.42E+06	9.38E+04
	f-rank	4	5	2	9	3	7	6	8	1
F19	Mean	8.98E+03	7.35E+03	9.83E+03	1.71E+05	9.31E+03	8.85E+06	1.42E+05	2.59E+04	6.87E+03
	Std	1.10E+04	6.99E+03	1.12E+04	1.49E+05	1.32E+04	3.52E+07	3.82E+05	1.91E+04	6.96E+03
	f-rank	3	2	5	8	4	9	7	6	1
F20	Mean	2.27E+03	2.28E+03	2.64E+03	2.52E+03	2.19E+03	2.58E+03	2.56E+03	2.45E+03	2.15E+03
	Std	1.43E+02	1.33E+02	1.97E+02	2.51E+02	1.36E+02	2.11E+02	2.48E+02	1.74E+02	1.27E+02
	f-rank	3	4	9	6	2	8	7	5	1
F21	Mean	2.35E+03	2.38E+03	2.47E+03	2.50E+03	2.34E+03	2.48E+03	2.45E+03	2.45E+03	2.34E+03
	Std	1.56E+01	2.06E+01	5.17E+01	1.29E+01	1.08E+01	3.39E+01	4.31E+01	4.31E+01	1.86E+01
	f-rank	3	4	7	9	1	8	5	5	2
F22	Mean	3.13E+03	2.31E+03	5.47E+03	6.33E+03	2.55E+03	6.30E+03	5.70E+03	3.65E+03	2.30E+03
	Std	1.59E+03	3.48E+00	2.04E+03	3.38E+03	9.57E+02	1.65E+03	2.17E+03	1.96E+03	6.23E-01
	f-rank	4	2	6	9	3	8	7	5	1
F23	Mean	2.70E+03	2.73E+03	2.91E+03	2.85E+03	2.69E+03	2.80E+03	2.78E+03	2.77E+03	2.69E+03
	Std	1.62E+01	2.01E+01	6.10E+01	1.08E+01	1.56E+01	2.88E+01	3.30E+01	3.10E+01	1.44E+01
	f-rank	3	4	9	8	2	7	6	5	1
F24	Mean	2.88E+03	2.89E+03	3.08E+03	3.01E+03	2.86E+03	2.96E+03	2.95E+03	2.93E+03	2.86E+03
	Std	2.06E+01	1.86E+01	7.97E+01	1.26E+01	1.40E+01	2.68E+01	3.21E+01	2.39E+01	1.36E+01
	f-rank	3	4	9	8	2	7	6	5	1
F25	Mean	2.89E+03	2.91E+03	2.90E+03	3.10E+03	2.89E+03	3.13E+03	2.92E+03	2.94E+03	2.88E+03
	Std	2.76E+00	1.78E+02	1.81E+01	6.83E+01	1.04E+01	2.06E+02	3.16E+01	4.29E+01	7.17E+00
	f-rank	2	5	4	8	3	9	6	7	1
F26	Mean	3.95E+03	4.33E+03	6.65E+03	5.65E+03	4.06E+03	5.53E+03	5.16E+03	4.94E+03	3.95E+03
	Std	4.86E+02	3.17E+02	1.04E+03	1.23E+02	1.80E+02	4.46E+02	5.19E+02	3.46E+02	4.47E+02
	f-rank	2	4	9	8	3	7	6	5	1
F27	Mean	3.21E+03	3.22E+03	3.26E+03	3.21E+03	3.21E+03	3.23E+03	3.23E+03	3.22E+03	3.21E+03
	Std	9.31E+00	1.15E+01	2.71E+01	5.19E+00	8.78E+00	1.65E+01	2.08E+01	1.36E+01	1.31E+00
	f-rank	4	5	9	2	3	7	8	6	1
F28	Mean	3.21E+03	3.26E+03	3.21E+03	3.40E+03	3.21E+03	3.68E+03	3.35E+03	3.34E+03	3.18E+03
	Std	1.52E+01	3.84E+01	2.70E+01	3.80E+01	1.80E+01	4.82E+02	5.95E+01	6.30E+01	4.71E+01
	f-rank	2	5	4	8	3	9	7	6	1
F29	Mean	3.53E+03	3.73E+03	4.10E+03	4.09E+03	3.56E+03	4.02E+03	3.99E+03	3.80E+03	3.55E+03
	Std	1.42E+02	2.09E+02	2.67E+02	2.36E+02	1.31E+02	1.93E+02	2.13E+02	1.99E+02	1.69E+02
	f-rank	1	4	9	8	3	7	6	5	2
F30	Mean	9.33E+03	8.91E+04	9.44E+03	7.83E+05	8.96E+03	3.00E+05	5.99E+05	3.31E+04	8.29E+03
	Std	3.55E+03	1.07E+05	3.45E+03	3.80E+05	3.10E+03	4.82E+05	2.11E+06	2.88E+04	3.33E+03
	f-rank	3	6	4	9	2	7	8	5	1
	Average f-rank	3.07	4.31	5.66	7.76	2.17	7.97	6.62	5.83	1.59
	Overall f-rank	3	4	5	8	2	9	7	6	1

Table 3. Comparisons of nine algorithms on CEC 2017 benchmark functions with 100 dimensions

Function	Results	EO	mEO	LWMEO	ISEO	IEO	MFO	WEMFO	DMMFO	HMEO
F1	Mean	1.84E+08	9.07E+09	7.47E+05	1.74E+11	2.15E+07	1.31E+11	3.51E+10	2.99E+10	2.88E+04
	Std	2.39E+08	2.35E+09	8.45E+05	1.82E+10	1.86E+07	3.66E+10	3.65E+09	7.47E+09	1.33E+04
	f-rank	4	5	2	9	3	8	7	6	1
F3	Mean	4.06E+05	3.22E+05	4.95E+05	8.91E+05	3.81E+05	9.23E+05	3.45E+05	8.68E+05	2.05E+05
	Std	5.91E+04	3.59E+04	1.81E+05	1.06E+06	4.44E+04	1.73E+05	2.14E+04	1.09E+05	1.79E+04
	f-rank	5	2	6	8	4	9	3	7	1
F4	Mean	9.54E+02	1.87E+03	8.48E+02	2.77E+04	8.96E+02	1.61E+04	4.24E+03	3.41E+03	7.37E+02
	Std	6.96E+01	3.05E+02	8.69E+01	6.69E+03	4.77E+01	6.22E+03	1.11E+03	6.99E+02	4.21E+01
	f-rank	4	5	2	9	3	8	7	6	1
F5	Mean	1.06E+03	1.25E+03	1.34E+03	1.56E+03	9.30E+02	1.77E+03	1.37E+03	1.43E+03	9.45E+02
	Std	7.53E+01	8.99E+01	9.95E+01	3.46E+01	6.28E+01	1.52E+02	6.82E+01	1.04E+02	5.25E+01
	f-rank	3	4	5	8	1	9	6	7	2
F6	Mean	6.16E+02	6.34E+02	6.54E+02	6.41E+02	6.07E+02	6.77E+02	6.72E+02	6.66E+02	6.06E+02
	Std	7.46E+00	8.01E+00	5.83E+00	4.41E+00	1.84E+02	6.32E+00	1.43E+01	1.01E+01	2.98E+00
	f-rank	3	4	6	5	2	9	8	7	1
F7	Mean	1.59E+03	2.07E+03	3.23E+03	3.56E+03	1.37E+03	4.61E+03	2.61E+02	3.32E+03	1.37E+03
	Std	1.24E+02	1.59E+02	2.74E+02	6.83E+02	9.38E+01	5.75E+02	1.38E+02	5.02E+02	1.18E+01
	f-rank	4	5	6	8	3	9	1	7	2
F8	Mean	1.34E+03	1.55E+03	1.66E+03	1.87E+03	1.21E+03	2.04E+03	1.71E+03	1.72E+03	1.23E+03
	Std	9.19E+01	7.49E+01	8.69E+01	4.77E+01	7.51E+01	1.47E+02	1.19E+02	1.14E+02	5.55E+01
	f-rank	3	4	5	8	1	9	6	7	2
F9	Mean	1.96E+04	2.55E+04	2.55E+04	1.48E+04	9.81E+03	5.48E+04	4.67E+04	3.67E+04	1.56E+04
	Std	5.63E+03	4.79E+03	2.84E+03	2.61E+03	3.11E+03	6.39E+03	1.54E+04	9.95E+03	2.58E+03
	f-rank	4	6	5	2	1	9	8	7	3
F10	Mean	1.99E+04	2.12E+04	1.63E+04	3.24E+04	2.03E+04	1.83E+04	1.96E+04	1.81E+04	1.41E+04
	Std	2.21E+03	1.79E+03	1.45E+03	7.99E+02	2.32E+03	1.49E+03	1.79E+03	1.27E+03	1.86E+03
	f-rank	6	8	2	9	7	4	5	3	1
F11	Mean	1.96E+04	1.98E+04	4.03E+03	1.41E+05	1.41E+04	1.82E+05	7.85E+04	1.41E+05	4.29E+03
	Std	4.56E+03	4.29E+03	1.77E+03	2.64E+04	3.94E+03	5.34E+04	1.41E+04	2.84E+04	9.09E+02
	f-rank	4	5	1	7	3	9	6	8	2
F12	Mean	6.71E+07	7.79E+08	2.67E+07	2.79E+10	2.84E+07	2.78E+10	3.23E+09	2.62E+09	1.63E+07
	Std	2.62E+07	1.87E+08	2.06E+07	7.79E+09	1.28E+07	1.22E+10	1.09E+09	1.28E+09	7.86E+06
	f-rank	4	5	2	9	3	8	7	6	1
F13	Mean	2.19E+04	1.96E+06	1.39E+04	5.19E+09	1.37E+04	3.46E+09	2.84E+07	1.45E+07	1.52E+04
	Std	9.08E+03	2.25E+06	5.79E+03	1.15E+09	6.71E+03	2.29E+09	3.97E+07	3.91E+07	8.42E+03
	f-rank	4	5	2	9	1	8	7	6	3
F14	Mean	1.86E+06	3.42E+06	9.65E+05	4.38E+07	1.71E+06	1.39E+07	1.02E+07	1.23E+07	6.54E+05
	Std	8.17E+05	2.09E+06	4.95E+05	1.81E+07	9.42E+05	1.57E+07	5.10E+06	5.44E+06	3.29E+05
	f-rank	4	5	2	9	3	8	6	7	1
F15	Mean	6.45E+03	2.06E+05	5.09E+03	7.36E+08	5.28E+03	8.37E+08	4.85E+06	3.90E+05	6.01E+03
	Std	2.85E+03	1.84E+05	1.48E+03	2.02E+08	2.79E+03	1.20E+09	1.59E+07	1.38E+06	3.76E+03
	f-rank	4	5	1	8	2	9	7	6	3
F16	Mean	5.99E+03	6.72E+03	6.46E+03	1.04E+04	5.53E+03	8.16E+03	7.43E+03	6.66E+03	4.95E+03
	Std	9.54E+02	9.18E+02	6.80E+02	3.76E+02	9.06E+02	6.93E+02	7.42E+02	7.32E+02	7.71E+02
	f-rank	3	6	4	9	2	8	7	5	1
F17	Mean	4.98E+03	5.51E+03	6.06E+03	8.46E+03	4.59E+03	7.77E+03	6.28E+03	6.07E+03	4.77E+03
	Std	6.17E+02	6.19E+02	5.88E+02	4.23E+02	6.22E+02	1.41E+03	6.88E+02	5.77E+02	5.69E+02
	f-rank	3	4	5	9	1	8	7	6	2
F18	Mean	2.90E+06	4.65E+06	1.16E+06	9.32E+07	2.43E+06	1.66E+07	1.45E+07	1.60E+07	1.34E+06
	Std	1.45E+06	1.86E+06	5.39E+05	3.35E+07	1.13E+06	1.13E+07	6.15E+06	8.41E+06	3.98E+05
	f-rank	4	5	1	9	3	8	6	7	2
F19	Mean	6.33E+03	1.54E+06	7.49E+03	8.40E+08	5.70E+03	3.33E+08	9.15E+06	1.77E+06	6.50E+03
	Std	5.59E+03	9.78E+05	5.14E+03	2.00E+08	4.88E+03	4.62E+08	1.56E+07	4.27E+06	5.34E+03
	f-rank	2	5	4	9	1	8	7	6	3
F20	Mean	5.31E+03	5.26E+03	5.65E+03	7.57E+03	4.78E+03	5.76E+03	5.78E+03	5.22E+03	4.58E+03
	Std	5.18E+02	5.66E+02	4.52E+02	2.71E+02	4.99E+02	5.77E+02	4.96E+02	5.20E+02	5.90E+02
	f-rank	5	4	6	9	2	7	8	3	1
F21	Mean	2.77E+03	2.94E+03	3.58E+03	3.34E+03	2.65E+03	3.57E+03	3.21E+03	3.26E+03	2.65E+03
	Std	7.10E+01	7.78E+01	2.63E+02	3.38E+01	3.74E+01	1.19E+02	1.05E+02	1.34E+02	5.78E+01
	f-rank	3	4	9	7	1	8	5	6	2
F22	Mean	2.29E+04	2.35E+04	1.93E+04	3.44E+04	2.26E+04	2.09E+04	2.20E+04	2.08E+04	1.58E+04
	Std	2.02E+03	2.01E+03	2.18E+03	7.21E+02	2.04E+03	1.72E+03	1.15E+03	1.57E+03	5.61E+03
	f-rank	7	8	2	9	6	4	5	3	1
F23	Mean	3.22E+03	3.38E+03	4.14E+03	3.74E+03	3.11E+03	3.75E+03	3.72E+03	3.58E+03	3.15E+03
	Std	5.80E+01	1.06E+02	2.43E+02	3.82E+01	5.46E+01	9.86E+01	1.23E+02	9.39E+01	5.08E+01
	f-rank	3	4	9	7	1	8	6	5	2
F24	Mean	3.66E+03	3.85E+03	5.06E+03	4.24E+03	3.57E+03	4.28E+03	4.23E+03	4.08E+03	3.62E+03
	Std	7.38E+01	9.70E+01	2.81E+02	2.83E+01	6.35E+01	1.48E+02	1.16E+02	1.03E+02	7.63E+01
	f-rank	3	4	9	7	1	8	6	5	2
F25	Mean	3.63E+03	4.49E+03	3.53E+03	2.03E+04	3.54E+03	1.50E+04	6.41E+03	7.86E+03	3.38E+03
	Std	6.23E+01	2.11E+02	9.84E+01	3.33E+03	5.61E+01	3.94E+03	6.90E+02	1.33E+03	5.63E+01
	f-rank	4	5	2	9	3	8	6	7	1
F26	Mean	1.03E+04	1.27E+04	2.29E+04	1.63E+04	9.03E+04	1.76E+04	1.62E+04	1.51E+04	1.00E+04
	Std	8.58E+02	1.34E+03	2.54E+03	5.21E+02	8.36E+02	1.12E+03	1.05E+03	1.14E+03	1.59E+03
	f-rank	2	3	8	6	9	7	5	4	1
F27	Mean	3.50E+03	3.62E+03	3.95E+03	3.85E+03	3.51E+03	3.85E+03	3.80E+03	3.70E+03	3.46E+03
	Std	5.35E+01	6.03E+01	1.81E+02	1.33E+02	4.84E+01	2.00E+02	1.14E+02	1.13E+02	4.43E+01
	f-rank	2	4	9	7	3	8	6	5	1
F28	Mean	3.79E+03	4.92E+03	3.67E+03	1.64E+04	3.68E+03	1.78E+04	1.51E+04	1.51E+04	3.51E+03
	Std	7.14E+01	4.93E+02	9.31E+01	1.59E+03	5.41E+01	1.55E+03	3.07E+03	1.83E+03	4.94E+01
	f-rank	4	5	2	8	3	9	7	6	1
F29	Mean	6.59E+03	7.57E+03	7.82E+03	1.11E+04	6.27E+03	9.49E+03	8.65E+03	7.54E+03	6.08E+03
	Std	7.51E+02	7.43E+02	6.73E+02	5.41E+02	6.81E+02	1.30E+03	7.59E+02	6.38E+02	5.93E+02
	f-rank	3	5	6	9	2	8	7	4	1
F30	Mean	1.96E+05	1.60E+07	2.18E+05	1.46E+09	5.34E+04	1.43E+09	2.90E+07	8.20E+06	4.50E+04
	Std	1.66E+05	9.80E+06	2.19E+05	4.45E+08	3.60E+04	1.22E+09	2.14E+07	8.85E+06	2.55E+04
	f-rank	3	6	4	9	2	8	7	5	1
	Average f-rank	3.69	4.83	4.38	7.97	2.66	7.97	6.17	5.76	1.59
	Overall f-rank	3	5	4	8	2	8	7	6	1

icant practical and theoretical implications, contributing to advancements in autonomous navigation capabilities, resource optimization, operational efficiency, safety, and the overall progress of the robotics field.

The purpose of studying path planning techniques for mobile robots is to determine an obstacle-free path between the initial and target positions in an environment containing various obstacles [82]. Methodologically, the solutions to this problem can be divided into traditional and intelligent methods. The traditional methods used in MRPP include grid-based methods, cell decomposition methods, and visual mapping methods. However, these traditional strategies suffer from computational complexity and inefficiency in the face of complex environments. Therefore, researchers strive to adopt intelligent methods to address these limitations and achieve more efficient solutions to MRPP problems. An influential class of intelligent techniques in this area revolves around metaheuristic-based MRPP methods that have demonstrated their competitive advantage and effectiveness through extensive research efforts. These methods make use of optimization algorithms inspired by natural phenomena or social behavior, such as PSO, FA, ABC, and SSA. Through the exploration of intelligent methods, researchers have worked to improve the efficiency, adaptability, and robustness of MRPP algorithms, enabling mobile robots to navigate complex environments more effectively.

In [83], the search capability of the SSA algorithm is enhanced by employing reverse learning based on the principle of optical lens imaging and a rank-driven dynamic learning mechanism. The improved SSA algorithm is referred to as OOSSA, and it is utilized for designing obstacle-free paths for autonomous mobile robots (AMR) in different environments. Experimental results demonstrate that OOSSA provides optimal paths compared to classical metaheuristic algorithms. In [84], an enhanced SSA algorithm based on orthogonal quasi-opposite learning, known as OBDSSA, is proposed. In OBDSSA, an enhanced reverse learning mechanism is employed to help the population escape local optima, and dynamic learning techniques are used to facilitate individual exploration of unexplored regions. The performance of the MRPP method based on OBDSSA is investigated in five different environments and compared with advanced metaheuristic algorithms. The comparative results indicate that OBDSSA is an effective path planning algorithm. In [85], an improved genetic algorithm-based path planning method is proposed. The enhanced genetic algorithm incorporates dynamic mutation rates and switchable global-local search operations to improve convergence speed and reduce computational costs. The improved genetic algorithm is directly applied to provide smooth paths for mobile robots. The effectiveness of the algorithm is validated through a series of experiments. In [86], a quantum-behaved particle swarm optimization (PSO) algorithm for the MRPP problem, named EDPSO, is introduced. In EDPSO, a diversity peak strategy is implemented to enhance the diversity of the population, thereby reducing the probability of the algorithm getting trapped in local optima. The proposed algorithm is simulated in four environments with different complexities. Through simulations, the EDPSO algorithm is proven capable of generating collision-free and planned trajectories. Overall, these studies contribute to the advancement of path planning techniques for mobile robots by introducing novel algorithmic enhancements and demonstrating their effectiveness through experimental evaluations. The impact of these advances extends to a wide range of real-world applications, including autonomous robotics, industrial automation, transportation systems, and search and rescue missions. In addition, continued research and development of MRPP technology contributes to advances in robotics, paving the way for increasingly intelligent and autonomous robotic systems of the future. The objective of this subsection is to apply the HMEO algorithm to solve the MRPP problem in AMRs. To validate the effectiveness of the proposed method, a series of systematic experiments were conducted to evaluate its performance.

6.1 Robot path planning problem description

From a topological perspective, the Mobile Robot Path Planning (MRPP) problem involves determining a path between two points in a plane that is both short in distance and free from obstacles. This problem can be transformed into an op-

timization problem, which can be effectively addressed using metaheuristic algorithms. The process of solving such problems with metaheuristic algorithms generally involves three steps: firstly, formulating the objective function that represents the problem's requirements; secondly, selecting an optimization tool in the metaheuristic family; and finally, iteratively optimizing the objective function using the chosen algorithm to obtain the optimal solution. Undoubtedly, the design of the objective function is of utmost importance as it directly influences the quality of the path generated by the algorithm. In this study, the objective function is crafted by considering two essential factors: the first objective aims to identify the shortest path, while the second objective focuses on circumventing all obstacles. Drawing upon the aforementioned considerations, the following objective function is devised:

$$\text{objectiveValue}(\text{OV}) = L(1+\sigma\cdot\eta) \tag{20}$$

where OV indicates the objective value, L means the path length, σ denotes the penalty factor, and η represents the sign variable. The metaheuristic technique aims to minimize the aforementioned objective function through iterative computations. The first constraint is fulfilled by reducing the path length, while the second constraint is met by minimizing the sign variable. The flag variable is utilized to indicate whether the path encounters any obstacles or not. A flag variable equal to 0 indicates that the current path is obstacle-free, whereas a higher flag variable discourages paths that traverse through obstacles.

6.2 Experiment and results

In this subsection, the MRPP problem is emulated in five distinct environmental maps of varying complexity utilizing a range of nature-inspired intelligent algorithms. The effectiveness of the HMEO algorithm in addressing the MRPP problem is examined by comparing its results with those obtained by ABC, FA, PSO, and GWO. The five topographies employed for simulating the MRPP problem were selected from [87], and their specific details are outlined in Table 4. To ensure a fair comparison, the specific parameters of the comparison algorithms utilized were extracted from the original literature sources for each respective algorithm. The path lengths generated by each algorithm under different environmental conditions are presented in Table 5, accompanied by the corresponding trajectories depicted in Figures 6-10.

Table 5 presents a comparative analysis of the path lengths obtained by the five involved algorithms in five different environmental settings. Based on the comparison results, it is evident that the HMEO algorithm consistently yields the shortest paths across all terrains. This compelling outcome underscores the efficacy and practical applicability of the HMEO-based MRPP method. These findings highlight the superior performance and efficiency of the HMEO algorithm in addressing the MRPP problem, further demonstrating its potential for real-world implementation. These results affirm the value of adopting nature-inspired metaheuristics, especially the HMEO algorithm, for effective path planning in different environmental scenarios. The advantages of HMEO in achieving shorter path lengths in all environmental setups emphasize its significance as a promising tool for MRPP challenges in AMRs.

According to Figure 6, the paths generated by GWO and SSA exhibit similar trajectories, while ABC, FA, PSO, and HMEO produce another set of similar trajectories. It is evident that the paths generated by ABC, FA, and PSO exhibit redundancy, indicating that these methods may have become trapped in local optima. In contrast, HMEO yields the shortest path.

From Figure 7, it can be observed that the paths generated by GWO and SSA are the least favorable, showing significant redundancy. On the other hand, ABC, FA, PSO, and HMEO produce more direct paths. In comparison to the other three algorithms, the path generated by HMEO is smoother and has a shorter length.

Considering Figure 8, the paths generated by ABC, FA, GWO, and SSA exhibit more convoluted trajectories, resulting in longer path lengths. In contrast, PSO and HMEO generate more direct obstacle-free paths, effectively reducing the path length. Compared to the PSO algorithm, the path planned by HMEO is more sensible, resulting in a shorter path length.

From Figure 9, it can be observed that all six algorithms generate similar obstacle-free trajectories.

Table 4. Type of environment

Terrain	No. obstacle	Initial coordinates	Final coordinates	X axis	Y axis	Obstacle radius
Map 1	3	0, 0	4, 6	[1 1.8 4.5]	[1 5.0 0.9]	[0.8 1.5 1]
Map 2	6	0, 0	10, 10	[1.5 8.5 3.2 6.0 1.2 7.0]	[4.5 6.5 2.5 3.5 1.5 8.0]	[1.5 0.9 0.4 0.6 0.8 0.6]
Map 3	13	3, 3	14, 14	[1.5 4.0 1.2 5.2 9.5 6.5 10.8 5.9 3.4 8.6 11.6 3.3 11.8]	[4.5 3.0 1.5 3.7 10.3 7.3 6.3 9.9 5.6 8.2 8.6 11.5 11.5]	[0.5 0.4 0.4 0.8 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7 0.7]
Map 4	30	3, 3	14, 14	[10.1 10.6 11.1 11.6 12.1 11.2 11.7 12.2 12.7 13.2 11.4 11.9 12.4 12.9 13.4 8 8.5 9 9.5 10 9.3 9.8 10.3 10.8 11.3 5.9 6.4 6.9 7.4 7.9]	[8.8 8.8 8.8 8.8 8.8 11.7 11.7 11.7 11.7 11.7 9.3 9.3 9.3 9.3 9.3 5.3 5.3 5.3 5.3 5.3 6.7 6.7 6.7 6.7 6.7 8.4 8.4 8.4 8.4 8.4]	[0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4]
Map 5	45	0, 0	15, 15	[2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4 6 6 6 8 8 8 8 8 8 8 8 8 8 10 10 10 10 10 10 10 10 10 12 12 12 12 12 14 14 14 14]	[8 8.5 9 9.5 10 10.5 3 3.5 4 4.5 5 5.5 6 6.5 7 11 11.5 12 1 1.5 2 2.5 3 3.4 4 4.5 5 5 6 6.5 7 7.5 8 8.5 9 9.5 10 10 10.5 11 11.5 12 10 10.5 11 11.5]	[0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4]

Table 5. The minimum route length comparison of HMEO-based MRPP method and comparison approaches under five environmental setups

Terrain	PSO	FA	ABC	GWO	SSA	HMEO
	Path length	Path length	Path length	Path length	Path length	Path length
Map 1	7.4272	7.8184	7.4914	7.7769	8.0519	**7.4146**
Map 2	14.3192	14.4166	14.4468	16.6788	16.8134	**14.3081**
Map 3	16.8099	16.6557	18.6264	18.9443	16.6091	**15.8608**
Map 4	16.2252	16.4926	16.3881	17.3659	17.4373	**16.1969**
Map 5	24.7897	22.2073	22.7309	22.3784	22.8333	**21.5777**

Clearly, the paths generated by ABC, FA, GWO, and SSA contain redundant segments, indicating that these algorithms may have become trapped in local optima during the iterative optimization process. The paths planned by PSO and HMEO are more reasonable, leading to shorter path lengths. In comparison to PSO, the path generated by HMEO is smoother, closely following the obstacles while avoiding collisions, making it a more reasonable choice.

Figure 10 displays the paths planned by the six algorithms in the challenging environment setting 5. It is evident that ABC, FA, GWO, PSO, and SSA all become trapped in local optima, resulting in unnecessary turns. In contrast, HMEO plans a smooth trajectory that navigates smoothly between obstacles without passing through any of them. This demonstrates the powerful search capabilities of the HMEO algorithm, making it an excellent path planner.

Based on the above description, the experimental analysis reveals that the HMEO algorithm outperforms the compared algorithms in terms of generating shorter, more direct, and smoother paths in different environmental settings.

7 Discussions

According to the comparative experimental results on the CEC 2017 test suite, when comparing EO and HMEO on the same set of functions, HMEO not only demonstrates superiority in terms of convergence speed but also exhibits significantly better convergence accuracy than the basic EO. Furthermore, the performance of the developed HMEO in escaping local optima and stability surpasses that of EO. Comparative experiments with various advanced metaheuristic algorithms also validate the superior convergence performance of the proposed algorithm. To assess the impact of the proposed hybrid algorithm on the optimization capability of individual algorithms, this paper compares the HMEO algorithm with the basic EO and basic MFO algorithms. The experimental results indicate that the performance of the hybrid algorithm outperforms both individual algorithms, confirming the effectiveness of the proposed hybrid approach. However, due to the integration of operators from two algorithms during the search process, this algorithm incurs higher computational costs compared to basic EO and MFO. In further experiments applying HMEO to the MRPP, performance

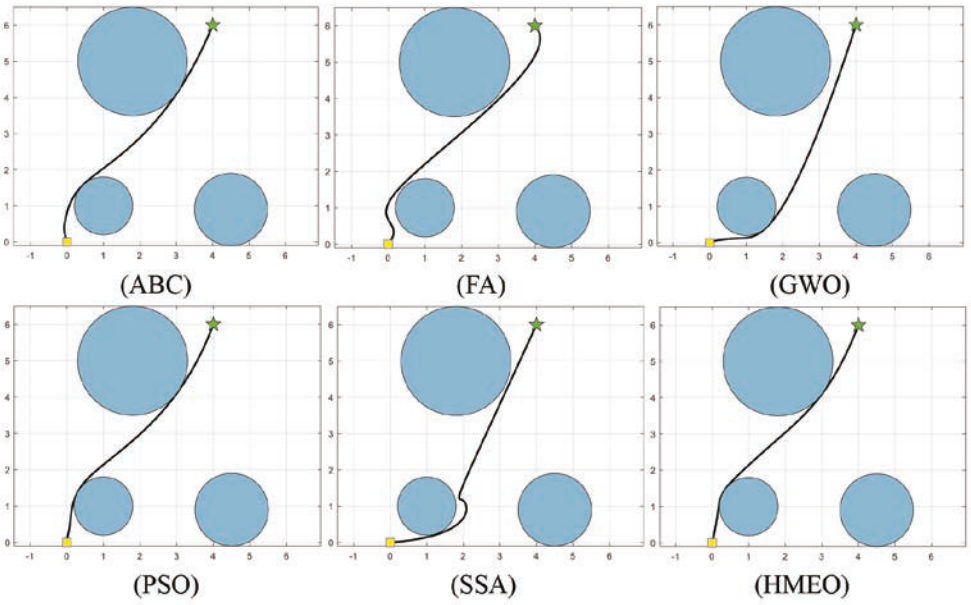

Figure 6. Map 1 (a) ABC, (b) FA, (c) GWO, (d) PSO, (e) SSA, and (f) HMEO

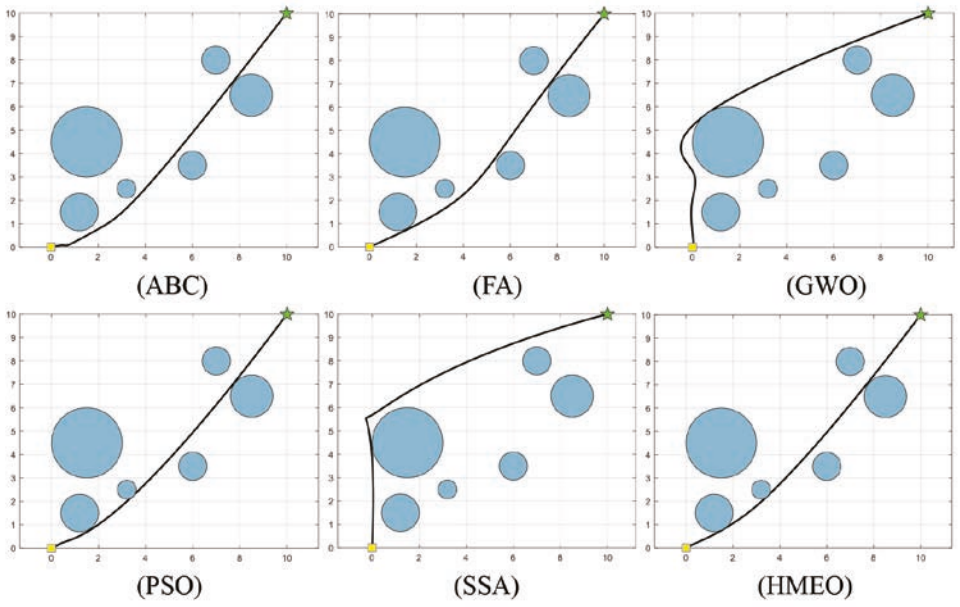

Figure 7. Map 2 (a) ABC, (b) FA, (c) GWO, (d) PSO, (e) SSA, and (f) HMEO

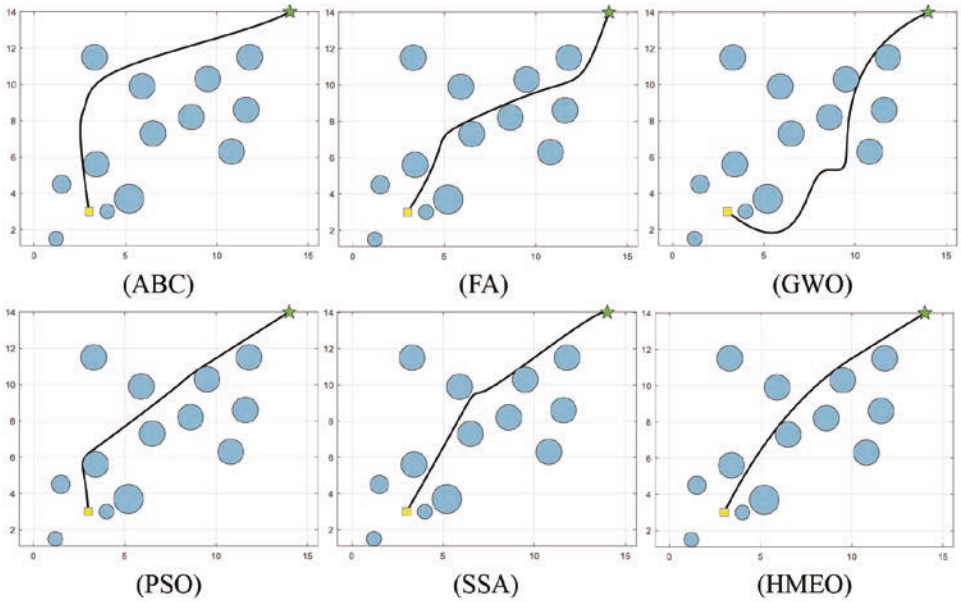

Figure 8. Map 3 (a) ABC, (b) FA, (c) GWO, (d) PSO, (e) SSA, and (f) HMEO

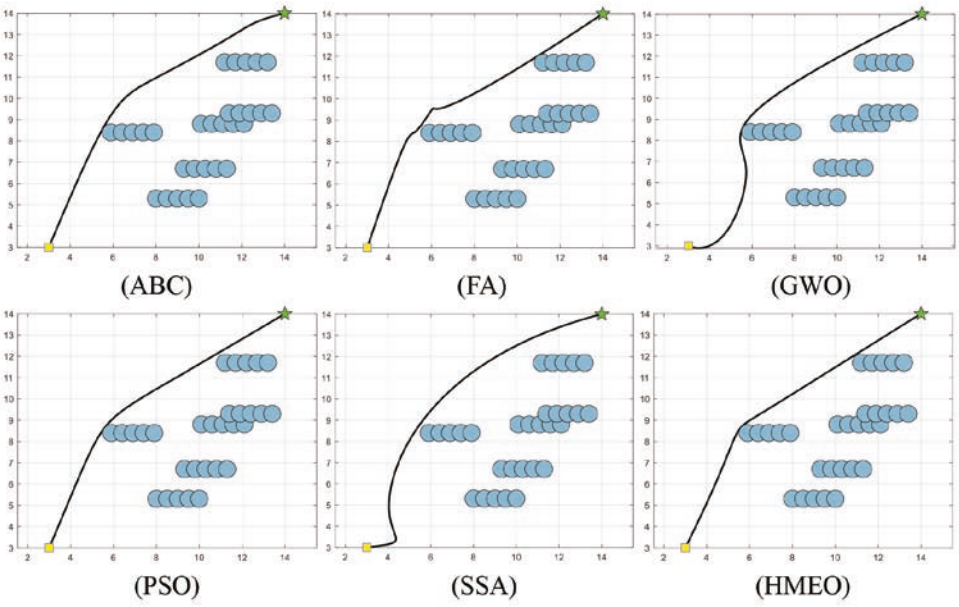

Figure 9. Map 4 (a) ABC, (b) FA, (c) GWO, (d) PSO, (e) SSA, and (f) HMEO

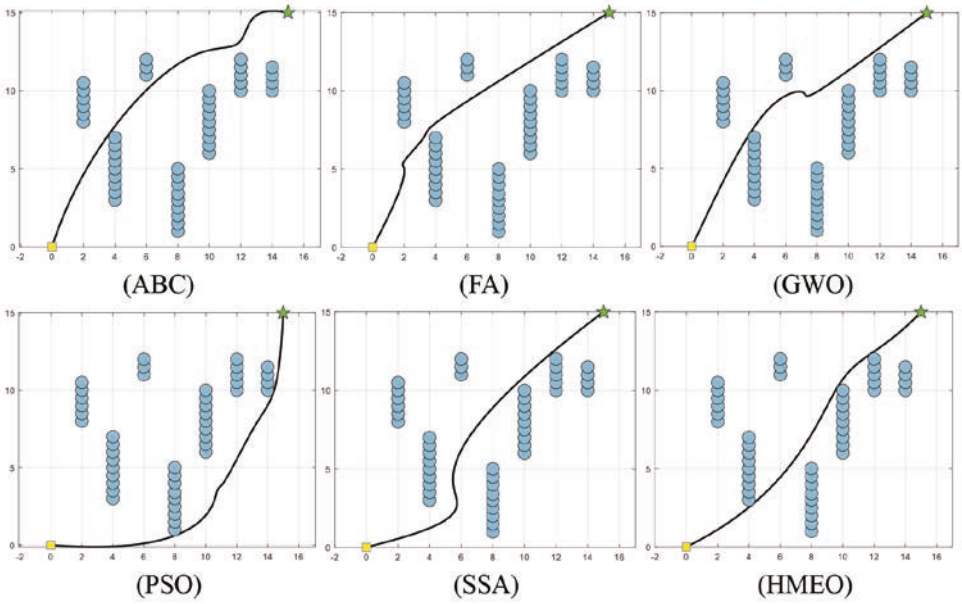

Figure 10. Map 5 (a) ABC, (b) FA, (c) GWO, (d) PSO, (e) SSA, and (f) HMEO

analysis based on shortest paths and scalability reveals that when HMEO is applied to the MRPP problem, its performance exceeds that of classical metaheuristic-based MRPP methods. Nevertheless, the CPU computation time required for path planning slightly exceeds that of comparative methods due to increased computational complexity when applying the HMEO algorithm to the MRPP problem. This drawback can be mitigated by employing parallel computing. Finally, the advocated HMEO's further application areas are the focus of our future research, such as image segmentation, image denoising, recommendation systems, and more.

Conclusion

This study proposes a novel hybrid algorithm based on EO and MFO, called HMEO. The HMEO algorithm utilizes the search framework of EO as its foundation, while incorporating the behavioral mechanisms of MFO for integration. By integrating the two algorithms, the HMEO algorithm aims to address the limitations of EO and MFO while leveraging their respective strengths. Extensive simulations using 29 benchmark functions from CEC 2017 test suite show that the proposed HMEO algorithm outperforms the original versions of EO and MFO, as well as other widely recognized variants of both algorithms. In addition, the study extends the application of HMEO to solve path planning problems for AMR and compares its performance with state-of-the-art metaheuristic algorithms. Experimental results conducted on five terrains of varying complexity confirm the competitive performance offered by the HMEO algorithm on the MRPP problem. Ultimately, this research aims to contribute to the optimization community by applying the novel HMEO algorithm to real-world problems and providing an alternative optimization tool for the engineering and scientific fields.

With the rapid development of modern society, the demands on algorithms for real-world engineering applications are escalating. Proposing an excellent algorithm to serve human life is our pursuit. Therefore, in the next steps, effective measures will be proposed to address the shortcomings of HMEO, aiming to significantly reduce the running time. In the future, HMEO will be applied to tackle even more complex real-world problems, such as image processing, recommendation systems, and beyond.

Acknowledgements This work was supported by the National Nature Science Foundation of China (grant no. 61461053, 61461054, and 61072079), Yunnan Provincial Education Department Scientific Research Fund Project (grant no. 2022Y008), the Yunnan University's Research Innovation Fund for Graduate Students, China (grant no. KC-22222706)

and Ding Hongwei expert grassroots research station of Youbei Technology Co, Yunnan Province.

Statements and Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Data Availability Statements

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests (financial and non-financial).

References

[1] P. Dziwiński, L. Bartczuk, J. Paszkowski. A new auto adaptive fuzzy hybrid particle swarm optimization and genetic algorithm. Journal of Artificial Intelligence and Soft Computing Research, 10(2): 95-111, 2020.

[2] F. Jia, S. Luo, G. Yin, Y. Ye. A novel variant of the salp swarm algorithm for engineering optimization. Journal of Artificial Intelligence and Soft Computing Research, 13, 2023.

[3] J. Xia, H. Zhang, R. Li, Z. Wang, Z. Cai, Z. Gu, H. Chen, Z. Pan. Adaptive barebones salp swarm algorithm with quasi-oppositional learning for medical diagnosis systems: A comprehensive analysis. Journal of Bionic Engineering, 2022.

[4] Z. Wang, H. Ding, B. Li, L. Bao, and Z. Yang. An energy efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks. IEEE Access, 8: 133577-133596, 2020.

[5] J. Xia, H. Zhang, R. Li, H. Chen, H. Turabieh, M. Mafarja, Z. Pan. Generalized oppositional moth flame optimization with crossover strategy: An approach for medical diagnosis. Journal of Bionic Engineering, 18: 991-1010, 2021.

[6] X. Wang, X. Dong, Y. Zhang, H. Chen. Crisscross harris hawks optimizer for global tasks and feature selection. Journal of Bionic Engineering, 2022.

[7] C. Lin, P. Wang, X. Zhao, and H Chen. Double mutational salp swarm algorithm: From optimal performance design to analysis. Journal of Bionic Engineering, 2022.

[8] J. H. Holland. Genetic algorithms. Scientific American, 267(1): 66-73, 1992.

[9] H. G. Beyer, H. P. Schwefel. Evolution strategies–a comprehensive introduction. Natural computing, 1: 3-52 2002.

[10] S. Das, P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation, 15(1): 4-31, 2010.

[11] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation, 3(2): 82-102, 1999.

[12] L. Hu, F. Lin, H. Li, C. Tong, Z. Pan, J. Liu, and H Chen. An intelligent prognostic system for analyzing patients with paraquat poisoning using arterial blood gas indexes. Journal of Pharmacological and Toxicological Methods, 84: 78-85, 2017.

[13] J. Kennedy, and R. Eberhart. Particle swarm optimization. in: Proceedings of ICNN'95-international conference on neural networks, 4: 1942–1948, IEEE, 1995.

[14] X.-S. Yang. Firefly algorithm: recent advances and applications. International Journal of Swarm Intelligence, 1(1): 36–50, 2013.

[15] Z. Wang, H. Ding, B. Li, L. Bao, Z. Yang, and Q. Liu. Energy efficient cluster based routing protocol for WSN using firefly algorithm and ant colony optimization. Wireless Personal Communications, 125(3): 2167-2200, 2022.

[16] D. Karaboga, and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Journal of Global Optimization, 39:459–471, 2007.

[17] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. IEEE Computational Intelligence Magazine, 1(4): 28–39, 2006.

[18] S. Mirjalili, S. M. Mirjalili, and A. Lewis. Grey wolf optimizer. Advances in Engineering Software, 69:(46–61), 2014.

[19] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, and H. Faris. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software, 114: 163–191, 2017.

[20] G. Dhiman, and V. Kumar. Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Advances in Engineering Software, 114: 48–70, 2017.

[21] G. Dhiman, and V. Kumar. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. Knowledge-Based Systems, 165: 169-196, 2019.

[22] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen. Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems, 97: 849-872, 2019.

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

[24] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. Gandomi. Marine predators algorithm: A nature-inspired metaheuristic. Expert Systems with Applications, 152: 113377, 2020.

[25] R. Salgotra, and U. Singh. The naked mole-rat algorithm. Neural Computing and Applications, 31: 8837-8857, 2019.

[26] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili. Slime mould algorithm: A new method for stochastic optimization. Future Generation Computer Systems, 111: 300-323, 2020.

[27] J. Tu, H. Chen, M. Wang, and A. H. Gandomi. The colony predation algorithm. Journal of Bionic Engineering, 18: 674-710, 2021.

[28] G. G. Wang, S. Deb, and Z. Cui. Monarch butterfly optimization. Neural Computing and Applications, 31: 1995-2014, 2019.

[29] H. Zhang, T. Liu, X. Ye, A. A. Heidari, G. Liang, H. Chen, and Z. Pan. Differential evolution-assisted salp swarm algorithm with chaotic structure for real-world problems. Engineering with Computers, 39(3): 1735-1769, 2023.

[30] R. Venkata Rao, V. J. Savsani, and D. P. Vakharia. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3): 303-315, 2011.

[31] E. Atashpaz-Gargari, and C. Lucas. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In 2007 IEEE congress on evolutionary computation, 4661-4667, IEEE, 2007.

[32] M. Kumar, A. J. Kulkarni, and S. C. Satapathy. Socio evolution and learning optimization algorithm: A socio-inspired optimization methodology. Future Generation Computer Systems, 81: 252-272, 2018.

[33] Y. Yang, H. 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, 177: 114864, 2021.

[34] S. H. S. Moosavi, and V. K. Bardsiri. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. Engineering Applications of Artificial Intelligence, 86: 165-181, 2019.

[35] L. Ren, A. A. Heidari, Z. Cai, Q. Shao, G. Liang, H. L. Chen, and Z. Pan. Gaussian kernel probability-driven slime mould algorithm with new movement mechanism for multi-level image segmentation. Measurement, 192: 110884, 2022.

[36] D. Bertsimas, and J. Tsitsiklis. Simulated annealing. Statistical Science, 8(1): 10-15, 1993.

[37] O. K. Erol, and I. Eksin. A new optimization method: big bang–big crunch. Advances in Engineering Software, 37(2): 106-111, 2006.

[38] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi. GSA: a gravitational search algorithm. Information Sciences, 179(13): 2232-2248, 2009.

[39] S. Mirjalili, and S. M. Mirjalili. Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications, 27: 495-513, 2016.

[40] H. Shah-Hosseini. Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimization. International Journal of Computational Science and Engineering, 6(1-2): 132-140, 2011.

[41] V. Goodarzimehr, S. Shojaee, S. Hamzehei-Javaran, and S. Talatahari. Special relativity search: A novel metaheuristic method based on special relativity physics. Knowledge-Based Systems, 257: 109484, 2022.

[42] A. Yadav. AEFA: Artificial electric field algorithm for global optimization. Swarm and Evolutionary Computation, 48: 93-108, 2019.

[43] H. Su, D. Zhao, A. A. Heidari, L. Liu, X. Zhang, M. Mafarja, and H. Chen. RIME: A physics-based optimization. Neurocomputing, 532: 183-214, 2023.

[44] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi. Equilibrium optimizer: A novel optimization algorithm. Knowledge-Based Systems, 191: 105190, 2020.

[45] Z. W. Geem, J. H. Kim, and G. V. Loganathan. A new heuristic optimization algorithm: harmony search. Simulation, 76(2): 60–68, 2001.

[46] 46. A. Kaveh, and S. Talatahari. Stochastic paint optimizer: theory and application in civil engineering. Engineering with Computers, 37: 1–32, 2020.

[47] M. Zaeimi, and A. Ghoddosian. Color harmony algorithm: an art-inspired metaheuristic for mathematical function optimization. Soft Computing, 24: 12027–12066, 2020.

[48] A. H. Gandomi. Interior search algorithm (isa): a novel approach for global optimization. ISA Transactions, 53(4): 1168–1183, 2014.

[49] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen. INFO: An efficient optimization algorithm based on weighted mean of vectors. Expert Systems with Applications, 195: 116516, 2022.

[50] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen. RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. Expert Systems with Applications, 181: 115079, 2021.

[51] Y. Liu, H. Ding, Z. Wang, G. Jin, B. Li, Z. Yang, and G. Dhiman. A chaos-based adaptive equilibrium optimizer algorithm for solving global optimization problems. Mathematical Biosciences and Engineering, 20(9): 17242-17271, 2023.

[52] O. E. Turgut, and M. S. Turgut. Diversity enhanced equilibrium optimization algorithm for solving unconstrained and constrained optimization problems. Evolutionary Intelligence, 1-52, 2023.

[53] G. I. Sayed, G. Khoriba, and M. H. Haggag. A novel chaotic equilibrium optimizer algorithm with S-shaped and V-shaped transfer functions for feature selection. Journal of Ambient Intelligence and Humanized Computing, 1-26, 2023.

[54] R. Atha, A. Rajan, and S. Mallick. An enhanced equilibrium optimizer for solving complex optimization problems. Information Sciences, 120077, 2024.

[55] Y. Liu, H. Ding, Z. Wang, G. Dhiman, Z. Yang, and P. Hu. An enhanced equilibrium optimizer for solving optimization tasks. Computers, Materials and Continua, 77(2), 2023.

[56] J. Xia, H. Zhang, R. Li, H. Chen, H. Turabieh, M. Mafarja, and Z. Pan. Generalized oppositional moth flame optimization with crossover strategy: an approach for medical diagnosis. Journal of Bionic Engineering, 18(4): 991-1010, 2021.

[57] M. H. Nadimi-Shahraki, H. Zamani, A. Fatahi, and S. Mirjalili. MFO-SFR: an enhanced moth-flame optimization algorithm using an effective stagnation finding and replacing strategy. Mathematics, 11(4): 862, 2023.

[58] S. K. Sahoo, S. Sharma, and A. K. Saha. A novel variant of moth flame optimizer for higher dimensional optimization problems. Journal of Bionic Engineering, 1-27, 2023.

[59] Z. Wang, H. Ding, J. Yang, P. Hou, G. Dhiman, J. Wang, Z. Yang, and A. Li. Orthogonal pinhole-imaging-based learning salp swarm algorithm with self-adaptive structure for global optimization. Frontiers in Bioengineering and Biotechnology, 10, 2022.

[60] K. Cpałka, A. Słowik, and K. Łapa. A population-based algorithm with the selection of evaluation precision and size of the population. Applied Soft Computing, 115, 108154, 2022.

[61] R. Salgotra, S. Singh, U. Singh, S. Mirjalili, and A. H. Gandomi. Marine predator-inspired naked mole-rat algorithm for global optimization. Expert Systems with Applications, 212, 118822, 2023.

[62] M. S. Uzer, and O. Inan. Application of improved hybrid whale optimization algorithm to optimization problems. Neural Computing and Applications, 35(17): 12433-12451, 2023.

[63] V. Garg, K. Deep, K. A. Alnowibet, H. M. Zawbaa, and A. W. Mohamed. Biogeography-based optimization with salp swarm optimizer inspired operator for solving non-linear continuous optimization problems. Alexandria Engineering Journal, 73: 321-341, 2023.

[64] C. Zhong, G. Li, and Z. Meng. A hybrid teaching–learning slime mould algorithm for global optimization and reliability-based design optimization problems. Neural Computing and Applications, 34, 19: 16617-16642, 2022.

[65] A. A. Ewees, H. I. Fatma, and T. S. Ahmed. Gradient-based optimizer improved by slime mould algorithm for global optimization and feature selection for diverse computation problems. Expert Systems with Applications, 213: 118872, 2023.

[66] S. Nama, A. K. Saha, S. Chakraborty, A. H. Gandomi, and L. Abualigah. Boosting particle swarm optimization by backtracking search algorithm for optimization problems. Swarm and Evolutionary Computation, 79: 101304, 2023.

[67] B. S. Yıldız, S. Kumar, N. Panagant, P. Mehta, S. M. Sait, A. R. Yildiz, N. Pholdee, S. Bureerat, and S. Mirjalili. A novel hybrid arithmetic optimization algorithm for solving constrained optimization problems. Knowledge-Based Systems, 271: 110554, 2023.

[68] A. Seyyedabbasi. WOASCALF: A new hybrid whale optimization algorithm based on sine cosine algorithm and levy flight to solve global optimization problems. Advances in Engineering Software, 173: 103272, 2022.

[69] L. Abualigah, A. A. Ewees, M. A. Al-Qaness, M. A. Elaziz, D. Yousri, R. A. Ibrahim, and M. Altalhi. Boosting arithmetic optimization algorithm by sine

cosine algorithm and levy flight distribution for solving engineering optimization problems. Neural Computing and Applications, 34(11): 8823-8852, 2022.

[70] S. K. Sahoo, and A. K. Saha. A hybrid moth flame optimization algorithm for global optimization. Journal of Bionic Engineering, 19(5): 1522-1543, 2022.

[71] 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, 38(4): 2797-2831, 2022.

[72] H. A. Shehadeh. A hybrid sperm swarm optimization and gravitational search algorithm (HSSOGSA) for global optimization. Neural Computing and Applications, 33(18): 11739-11752, 2021.

[73] W. H. Tan, and J. Mohamad-Saleh. A hybrid whale optimization algorithm based on equilibrium concept. Alexandria Engineering Journal 68: 763-786, 2023.

[74] S. Gupta, K. Deep. and S. Mirjalili. An efficient equilibrium optimizer with mutation strategy for numerical optimization, Applied Soft Computing 96, 106542, 2020.

[75] J. Liu, W. Li, Y. Li, Lwmeo: An efficient equilibrium optimizer for complex functions and engineering design problems, Expert Systems with Applications 198, 116828, 2022.

[76] X. Zhang, Q. Lin, Information-utilization strengthened equilibrium optimizer, Artificial Intelligence Review, 1–34, 2022.

[77] E. H. Houssein, B. E.-d. Helmy, D. Oliva, P. Jangir, M. Premkumar, A. A. Elngar, and H. Shaban. An efficient multi-thresholding based covid-19 CT images segmentation approach using an improved equilibrium optimizer, Biomedical Signal Processing and Control, 73, 103401, 2022.

[78] W. Shan, Z. Qiao, A. A. Heidari, H. Chen, H. Turabieh, and Y. Teng. Double adaptive weights for stabilization of moth flame optimizer: Balance analysis, engineering cases, and medical diagnosis. Knowledge-Based Systems, 214, 106728, 2021.

[79] L. Ma, C. Wang, N. G. Xie, M. Shi, Y. Ye, and L. Wang. Moth-flame optimization algorithm based on diversity and mutation strategy. Applied Intelligence, 51, 5836-5872, 2021.

[80] Z. Wang, and H. Ding. Opposition-based learning equilibrium ptimizer with application in mobile robot path planning. International Journal of Robotics and Automation Technology, 10, 64-74, 2023.

[81] H. Ding, X. Cao, Z. Wang, G. Dhiman, P. Hou, J. Wang, A. Li, and X. Hu. Velocity clamping-assisted adaptive salp swarm algorithm: balance analysis and case studies. Mathematical Biosciences and Engineering, 19(8): 7756-7804, 2022.

[82] Z. Wang, H. Ding, J. Wang, P. Hou, A. Li, Z. Yang, and X. Hu. Adaptive guided salp swarm algorithm with velocity clamping mechanism for solving optimization problems. Journal of Computational Design and Engineering, 9(6): 2022: 2196-2234.

[83] Z. Wang, H. Ding, Z. Yang, B. Li, Z. Guan, and L. Bao. Rank-driven salp swarm algorithm with orthogonal opposition-based learning for global optimization. Applied Intelligence, 5: 1-43, 2022.

[84] Z. Wang, H. Ding, J. Yang, J. Wang, B. Li, Z. Yang, and P. Hou. Advanced orthogonal opposition-based learning-driven dynamic salp swarm algorithm: framework and case studies. IET Control Theory and Applications, 16(10): 945-971, 2022.

[85] 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, 51(3): 1758-1774, 2023.

[86] P. B. Fernandes, R. C. L. Oliveira, and J. F. Neto. Trajectory planning of autonomous mobile robots applying a particle swarm optimization algorithm with peaks of diversity. Applied Soft Computing 116: 108108, 2022.

[87] 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, 107, 107372, 2021.

Zongshan Wang received his M.Sc. degree in Communication and Information System from Yunnan University, China, in 2020. He is currently pursuing a Ph.D. with the School of Information Science and Engineering, Yunnan University. His principle research interests include swarm intelligence optimization algorithms, energy-efficient routing protocol design for wireless sensor networks based on swarm intelligence algorithms, edge computing, robot path planning, and system modeling.
https://orcid.org/0000-0001-9945-3840

Ali Ala received his M.Sc. degree from Wuhan University of Technology in 2015 before embarking on his Ph.D. journey in healthcare optimization at Shanghai Jiao Tong University. He successfully completed his doctoral studies in 2022 under the supervision of the Department of Industrial Engineering and Management, School of Mechanical Engineering. Throughout his academic career, Ali has demonstrated a keen interest in optimization, healthcare operations, supply chain management, and decision making. His research has led to the publication of numerous articles in prestigious journals such as IEEE Transactions on Fuzzy Systems (TFS), Annals of Operations Research (ANOR), Expert Systems with Applications (ESWA), Applied Soft Computing (ASOC), Swarm and Evolutionary Computation (SCSI), and Computers in Biology and Medicine (CBM). With a strong foundation in both theoretical and applied research, Ali is dedicated to advancing the field of operations research and making significant contributions to real-world problem-solving.

Zekui Liu received the master's degree from Chongqing University of China in 2009. He is a lecturer in Chongqing Institute of Engineering. His research interests include mobile communications, optical network communications, deep learning algorithms for intelligent communication devices, robot control algorithms.

Wei Cui received the M.E. degree in control theory and control engineering in 2009. He is currently pursuing Ph.D. degree in Chongqing University of Posts and Telecommunications. His research interests include natural language processing, information mining and complex systems, multimodal content analysis, social network analysis, and evolutionary algorithms.

Hongwei Ding received his PhD. degree from the School of Information Science and Engineering, Yunnan University, Kunming, China, in 2002. He is currently a Professor and Ph.D. supervisor in the School of Information Science and Engineering, Yunnan University. He has published more than 100 research papers in International/National Conferences and Journals of good repute. His main research interests include wireless sensor networks, swarm intelligence optimization algorithms, communication and information systems, network and communication engineering, polling multiple access communication theory, and random multiple access communication systems.

Gushen Jin is an undergraduate student studies in Glasgow college in university of electronic science and technology of China. He is currently learning electronic information engineering. His main research interests include soft computing, swarm intelligence optimization algorithms, and robotics.

Xu Lu received his Bachelor's and Master's degrees in Communication and Information Systems from Yunnan University in 2014 and 2020, respectively. His main research interests include swarm intelligence optimization algorithms, energy-efficient routing protocol design for wireless sensor networks based on swarm intelligence algorithms, edge computing, robot path planning, and system modeling.