Arabian Journal for Science and Engineering (2023) 48:10493–10516
https://doi.org/10.1007/s13369-023-07683-2

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

A Multi-strategy Improved Outpost and Differential Evolution Mutation Marine Predators Algorithm for Global Optimization

Shuhan Zhang[1,2] · Shengsheng Wang[1,2] · Ruyi Dong[3] · Kai Zhang[1] · Xiaohui Zhang[4]

Received: 4 August 2022 / Accepted: 29 January 2023 / Published online: 17 February 2023

Abstract

Marine Predators Algorithm (MPA) is a recent efficient metaheuristic algorithm that is enlightened by the biological behavior of ocean predators and prey. This algorithm simulates the Levy and Brownian movements of prevalent foraging strategy and has been applied to many complex optimization problems. However, the algorithm has defects such as a low diversity of the solutions, ease into the local optimal solutions, and decreasing convergence speed in dealing with complex problems. A modified version of this algorithm called ODMPA is proposed based on the tent map, the outpost mechanism, and the differential evolution mutation with simulated annealing (DE-SA) mechanism. The tent map and DE-SA mechanism are added to enhance the exploration capability of MPA by increasing the diversity of the search agents, and the outpost mechanism is mainly used to improve the convergence speed of MPA. To validate the outstanding performance of the ODMPA, a series of global optimization problems are selected as the test sets, including the standard IEEE CEC2014 benchmark functions, which are the authoritative test set, three well-known engineering problems, and photovoltaic model parameters tasks. Compared with some famous algorithms, the results reveal that ODMPA has achieved better performance than its counterparts in CEC2014 benchmark functions. And in solving real-world optimization problems, ODMPA could get higher accuracy than other metaheuristic algorithms. These practical results demonstrate that the mechanisms introduced positively affect the original MPA, and the proposed ODMPA can be a widely effective tool in tackling many optimization problems.

Keywords Marine Predators Algorithm · Outpost mechanism · Differential evolution · Simulated annealing

1 Introduction

1.1 Background and Literature Review

With the progress of society, when making decisions to solve obstacles, people always find the best processing method to keep things in the optimal state. Metaheuristic algorithms have appeared in recent years as the general alternative in dealing with real-life problems, such as economics, industrial manufacturing, physical science research, civil engineering design, manufacturing systems, and communication network applications. When the problem's dimension increases extensively, traditional mathematical or physical optimization methods cannot solve these problems [1] due to the complex relationship between high dimensions and variables. Therefore, nature-inspired optimization algorithms came into being, broadly including evolutionary algorithms and swarm intelligence algorithms. With the growth of biology and AI, swarm intelligence algorithms have developed rapidly. They explore randomly between the solution space and update the optimal location through every iteration of the agents in the search range. Because of the effectiveness and convenience of the swarm algorithms [2–7], lots of researchers devoted themselves to their study of them.

Most metaheuristics are enlightened by the behavior and physical phenomena of biological groups in nature. For example, ant colony algorithm (ACO) [8] is encouraged by the phenomenon of ants' foraging strategy. That is,

✉ Shengsheng Wang
wss@jlu.edu.cn

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China

[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

[3] College of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin 132022, China

[4] 2012 Laboratories, Huawei Technology Co., Ltd., Beijing 100095, China

ants always search for the optimal route for food. Particle swarm optimization (PSO) [9] is encouraged by the swarming phenomenon of insects, animal herds, birds, etc. Grey wolf optimization algorithm (GWO) [10], the basic idea, is grey wolves' leadership level and predator rule. Moth-flame optimization (MFO) [11] mimics the laterally oriented moths' navigation mechanism. Multi-verse optimizer (MVO) [12] simulates the salps' chain phenomenon and is enlightened by the concepts in the cosmology of multiple universes. Salp swarm algorithm (SSA) [13], the main inspiration, is from the salp chain phenomenon, which could help the salps achieve better sailing and foraging in oceans. Many metaheuristics are applied to many domains [14–24], such as solar photovoltaic models, solid oxide fuel cell (SOFC), hydropower production, trusses optimization, and feature selection, and obtain excellent performance because of their efficiency, simplicity, and avoidance of local optimum. However, no algorithm declares it could tackle all the optimization problems, and the algorithm could exhibit outstanding performance on all the issues. Hence, the continued improvement of the optimization algorithms is essential to tackle more problems.

Marine Predators Algorithm (MPA) [25] has the same advantages as MAs, like efficiency, flexibility, simplicity, and excellent robustness, so MPA is effective in solving real-world problems. For example, Dinh PH et al. [26] introduce the MPA to generate the optimal parameters, which synthesize the low-frequency components. The results present that MPA performs excellently in improving the quality of fused images. To settle the problem of optimal power flow (OPF), Islam, M.Z. et al. [27] adopted the MPA on it and designed the objective function involving fuel cost, power loss, etc. They validated the outstanding performance of the IEEE 30-bus test system.

Besides, many scholars have devoted themselves to the study of MPA variants, and the proposed MPA variants perform outstandingly in global optimization problems. Yousri, D et al. [28] applied concept learning to MPA and then applied the improved MPA to deal with the feature selection problems. Elaziz, MA et al. [29] used the differential evolution operators to improve the original MPA's exploration stage. They adopted the improved MPA to the photovoltaic models. The stable and excellent results show that it is talented in PV modeling. Yousri, D et al. [30] proposed a new metaheuristic approach, CLMPA, combining comprehensive learning, and MPA. The proposed CLMPA is applied to distinguish the best parameters of the supercapacitor equivalent circuit. Sahlol AT et al. [31] introduced the fractional order into the MPA to put forward an improved algorithm, FO-MPA, for the hybrid classification of COVID-19 images while improving classification accuracy. Yousri, D et al. [32] combined the comprehensive learning and multi-swarm strategies with the MPA to generate an enhanced algorithm, and then, they used the CLDMMPA to solve the solid oxide fuel cell (SOFC) models. In order to address the problem of multilevel threshold image segmentation, Abd Elaziz, M et al. [33] proposed the developed quantum marine predators. The improved algorithm gets excellent results according to convergence and segmentation quality. Aiming at the problems of losses and voltage deviations in the distribution systems, Ahmad Eid et al. [34] introduced the MPA method into the distribution systems and got good results. To better deal with the Joint Regularized Semi-Supervised Extreme Learning Machine (JRSSELM) problem, Yang, WB et al. [35] introduced a multi-strategy MPA to optimize the parameter extraction process, and MSMPA-JRSSELM performs well in logging oil formation identification.

1.2 Novelty and Contributions

From the literature, the MPA-based variants are difficult to balance exploitation and exploration effectively, and most of them are only built on sharing historical experiences. However, the ability to jump out of the local optimum still needs to be improved. Although the above variants of MPA have improved the MPA's explosive ability, they still easily fall into the local optimal solution when tackling some practical problems [81], and there are still opportunities to improve their convergence, consistency, and reliability.

Aiming to address the shortcomings of MPA, an improved MPA was established based on two main mechanisms, the outpost mechanism [36] and differential evolution[37] mutation with simulated annealing [38] (DE-SA) mechanism. The convergence efficiency and accuracy of the algorithm are enhanced by the outpost mechanism, and the solution variety is increased by the DE-SA mechanism. For the original MPA, since the population initialization is chosen randomly, which may cause uneven distribution of solutions, so the tent chaotic map is introduced in the initialization stage. The better N individuals are selected from 2N individuals for iteration, which increases the diversity of the initial population and makes the population iterate toward the optimal direction. After the tent chaotic map [40] processes the initial population, the outpost mechanism is introduced, which consists of greedy selection and Gaussian mutation [36]. The outpost individuals performed exploration to decide whether the large group would move forward. This mechanism significantly improved the convergence efficiency and accuracy of MPA. Afterward, DE-SA mechanism is used to generate a new solution and decide whether to accept it. DE-SA mechanism greatly increases the solution variety and makes the solution less prone to fall into the local optima. This modified MPA is tested on thirty CEC2014 benchmark problems, and for real-world problems, the engineering design problems and photovoltaic models are chosen.

This paper's major contributions are as follows:

(1) An improved and efficient Marine Predators Algorithm variant ODMPA is proposed.
(2) Introduce the outpost mechanism and DE-SA strategy to improve the disadvantages of the premature phenomenon and slow convergence accuracy of MPA.
(3) The ODMPA is validated on the IEEE CEC2014 benchmark functions and ranks first in the conventional and advanced MAs by statistical analysis.
(4) The ODMPA is adopted to three classical engineering design problem and photovoltaic models, and ODMPA gets the lowest results compared to other algorithms.

The study is organized as follows. The next section briefly discusses the concise background and conception of MPA. The outpost mechanism, consisting of the mutation of Gaussian mutation and the greedy selection and differential evolution mutation mechanism, is presented in Sect. 3. It is worth noting that the simulated annealing enriches the DE mechanism. Furthermore, this section describes a detailed process for adding the strategies to the original MPA. Section 4 discusses the performance of ODMPA on IEEE CEC2014 benchmark functions, engineering design problems, and photovoltaic models. The last section is arranged for the conclusions and future work.

2 Overview of the Marine Predators Algorithm (MPA)

Marine Predators Algorithm (MPA) involved in this study is a recent meta-heuristic optimization algorithm. The inspiration for MPA mainly comes from the different motion modes of marine predators under different conditions. When predators encounter prey in nature, Levy and Brownian motions are suitable for predators to adopt encounter strategies. According to the information in the predation process, the different velocity ratios and motion patterns between the predator and prey, the predator will adopt Levy or Brownian motion to achieve the best predation strategy to capture the prey.

Like other nature-inspired metaheuristics, the initial solution of MPA is produced randomly between the threshold from the search range. The threshold indicates the solution space between the upper and lower bound. The search agents size is predefined, and each individual represents the search agent:

$$X_0 = X_{\min} + rand(X_{\max} - X_{\min}) \tag{1}$$

where $X_{\min}$ and $X_{\max}$ represent the search space's lower and upper bound, and $rand$ is a random number belonging to [0,1].

MPA considers the top predator, which has the most optimal fitness value and potential, so the best solution is nominated to construct the Elite matrix. Elite is the top predator, and it also has d dimensions. Moreover, the Elite and Prey matrices participate in each iteration of the predator's position migration.

$$Elite = \begin{bmatrix} X_{1,1}^t X_{1,2}^t ... X_{1,d}^t \\ X_{2,1}^t X_{2,2}^t ... X_{2,d}^t \\ \vdots \\ \vdots \\ X_{n,1}^t X_{n,2}^t ... X_{n,d}^t \end{bmatrix} \quad Prey = \begin{bmatrix} X_{1,1} X_{1,2} ... X_{1,d} \\ X_{2,1} X_{2,2} ... X_{2,d} \\ \vdots \\ \vdots \\ X_{n,1} X_{n,2} ... X_{n,d} \end{bmatrix} \tag{2}$$

where n represents the population size, t represents the current iteration, and d represents the problem dimension. X^t denotes the best predator, and Elite consists of n duplications of the best solution. $X_{i,j}$ denotes the i_{th} prey's j_{th} dimension. If the top predator is updated, the Elite will be refreshed by the substituted solution. MPA initializes the Prey matrix and chooses the optimal agent as the top predator to generate the Elite matrix.

MPA simulates the widespread foraging strategy in ocean predators with the consideration of three different velocity ratio of the predators and preys. And in the following phases, v represents the velocity ratio of prey to predator. Thus, the definition and formulas of three stages are given by: Thus, the definition and formulas of three stages are given by:

Phase 1: High Velocity

The prey swims more rapidly than the predators ($v \geq 10$) under the circumstance, and the predator's wisest foraging strategy is remaining stationary. The prey and predators are the searcher and target, respectively. That is to say, while $Iter < 1/3Max\_Iter$, the formula of the rule is:

$$\begin{aligned} &\overrightarrow{step_i} = \overrightarrow{R_B} \otimes (\overrightarrow{Elite_i} - \overrightarrow{R_B} \otimes \overrightarrow{Prey_i}) \\ &\overrightarrow{Prey_i} = \overrightarrow{Prey_i} + P.\overrightarrow{R} \otimes \overrightarrow{step_i} \qquad i = 1, \ldots, n \end{aligned} \tag{3}$$

where $\overrightarrow{R_B}$ is a normal distribution vector which denotes the Brownian motion. $\overrightarrow{R}$ is a vector of the random numbers between 0 and 1. $Iter$ is the present iteration, and Max_Iter is the maximum iteration number. $P = 0.5$ is a constant number.

Phase 2: Unit Velocity

The velocity is fundamentally the same for the prey and predator in this phase because they swim simultaneously ($v \approx 1$). The circumstance mimics the trail of prey conducts the prey and predator's movement. The roles of the predator and prey are not stationary. They can be the searcher or target. The exploration and exploitation operation are the same significant for this phase, so the population is divided into two parts. While $1/3Max\_Iter < Iter < 2/3Max\_Iter$, the formula of the first part:

$$\begin{aligned} \overrightarrow{step_i} &= \overrightarrow{R_L} \otimes (\overrightarrow{Elite_i} - \overrightarrow{R_L} \otimes \overrightarrow{Prey_i}) \\ \overrightarrow{Prey_i} &= \overrightarrow{Prey_i} + P.\overrightarrow{R} \otimes \overrightarrow{step_i} \qquad i = 1, \ldots, n/2 \end{aligned} \tag{4}$$

where $\overrightarrow{R_L}$ is a Levy movement vector, and $\overrightarrow{R_L} \otimes \overrightarrow{Prey}$ presents the prey motion. The formula of the second half:

$$\begin{aligned} \overrightarrow{step_i} &= \overrightarrow{R_B} \otimes (\overrightarrow{R_B} \otimes \overrightarrow{Elite_i} - \overrightarrow{Prey_i}) \\ \overrightarrow{Prey_i} &= \overrightarrow{Elite_i} + P.CF \otimes \overrightarrow{step_i} \qquad i = n/2, \ldots, n \end{aligned} \tag{5}$$

where $\overrightarrow{R_B} \otimes \overrightarrow{Elite_i}$ represents the movement of the predator, and CF as the step size control parameter for predator movement, the definition is as follows:

$$CF = \left(1 - \frac{Iter}{Max\_Iter}\right)^{\left(\frac{2 \cdot Iter}{Max\_Iter}\right)} \tag{6}$$

Phase 3: Low Velocity

The predators swim more rapidly than the prey (v = 0.1), and the situation occurs in the last phase of iteration. The prey is the target, so the predator's wisest foraging strategy is the Levy movement. While $2/3Max\_Iter < Iter < Max\_Iter$, this phase is presented as:

$$\begin{aligned} \overrightarrow{step_i} &= \overrightarrow{R_L} \otimes (\overrightarrow{R_L} \otimes \overrightarrow{Elite_i} - \overrightarrow{Prey_i}) \\ \overrightarrow{Prey_i} &= \overrightarrow{Elite_i} + P.CF \otimes \overrightarrow{step_i} \qquad i = 1, \ldots, n \end{aligned} \tag{7}$$

where $\overrightarrow{R_L} \otimes \overrightarrow{Elite_i}$ represents the levy movement of the predator.

Besides Levy and Brownian movements in three different phases, the most influential factor is environmental effects, fish aggregating devices (FADs). MPA uses long jumps to avoid getting stuck into the sub-optimal solution, and the jumping mode's formula is shown as follows:

$$\overrightarrow{Prey_i} = \begin{cases} \overrightarrow{Prey_i} + CF\left[\overrightarrow{X}_{\min} + \overrightarrow{R} \otimes (\overrightarrow{X}_{\max} - \overrightarrow{X}_{\min})\right] \\ \quad \otimes \overrightarrow{U} \qquad if\ r \leq FADs \\ \overrightarrow{Prey_i} + [FADs(1-r) + r] \\ \quad \times(\overrightarrow{Prey_{r1}} - \overrightarrow{Prey_{r2}}) \qquad if\ r > FADs \end{cases} \tag{8}$$

where r is a random number between 0 and 1, $\overrightarrow{X}_{\min}$ and $\overrightarrow{X}_{\max}$ represent the lower and upper bound of the search space, and $\overrightarrow{Prey_{r1}}$ and $\overrightarrow{Prey_{r2}}$ are randomly selected, mutually exclusive search agents in the Prey matrix. $\overrightarrow{U}$ denotes a random vector whose elements only can be 0 or 1, and $FADs$ = 0.2 is the probability of FADs effect.

Finally, the marine memory operation is used to update the position. This is similar to the greedy selection, which preserves the better solution and makes the population iterate toward a better direction. The principle is that after updating the prey matrix, the new solution should be evaluated to decide whether to update the Elite matrix. Ocean memory greatly improves the performance of MPA in population iteration. The main process of MPA is displayed in Algorithm 1.

Algorithm 1 Pseudo-code of MPA

```
Input: $N$, $D$, $Max\_iter$
Output: The optimal result
Initialization:
randomly generate Prey matrix by the lower and upper bound based on Eq. (1)
define $R$, $CF$, $\overrightarrow{R_B}$, $\overrightarrow{R_L}$, $FADs$=0.2, $P = 0.5$
Main loop:
while the current iteration is smaller than the maximum iteration do
    compute the fitness value from each individual, and construct Elite matrix by finding the best solution
    compare the previous and current fitness to accomplish memory saving
    if $Iter < 1/3Max\_Iter$ then
        update the prey's location by Eq. (3)
    else if $1/3Max\_Iter < Iter < 2/3Max\_Iter$ then
        for every individual from the first half ($i = 1, 2, \ldots, n/2$)
        update the prey's location by Eq. (4)
        for every individual from the second half (i=n/2, ..., n)
        update the prey's location by Eq. (5)
    else if $2/3Max\_Iter < Iter < Max\_Iter$ then
        update the prey's location by Eq. (7)
    end if
    compare the previous and current fitness to accomplish memory saving
    compute the fitness value from each individual, and construct Elite matrix by finding the best solution
    employ the Fish Aggregating Devices (FADs) effect to update the prey by Eq. (8)
end while
return result
```

3 Proposed ODMPA

Although the MPA is well known for its advantages of efficient foraging strategy and simple structure, the algorithm easily falls into the sub-optimum for complex optimization problems. In order to avoid the premature phenomenon of the algorithm and improve the diversity of solutions, the outpost mechanism and differential evolution mutation with simulated annealing (DE-SA) mechanism are introduced into the original MPA to propose ODMPA.

3.1 Outpost Mechanism

The outpost mechanism [36] includes the greedy selection and Gaussian mutation phases, and the process of the outpost mechanism is shown in Fig. 1. In the first phase, the prey

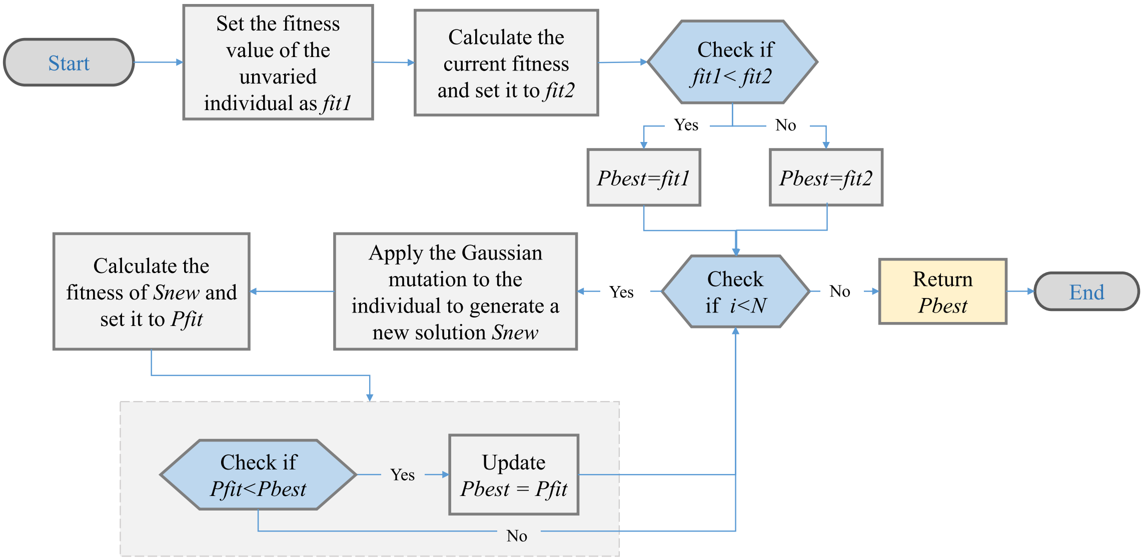

Start
Set the fitness value of the unvaried individual as *fit1*
Calculate the current fitness and set it to *fit2*
Check if *fit1< fit2*
Yes
No
Pbest=fit1
Pbest=fit2
Calculate the fitness of *Snew* and set it to *Pfit*
Apply the Gaussian mutation to the individual to generate a new solution *Snew*
Yes
Check if *i<N*
No
Return *Pbest*
End
Check if *Pfit<Pbest*
Yes
Update *Pbest = Pfit*
No

Fig. 1 Flowchart of outpost mechanism

compares the current fitness value with the last iteration. If the current fitness is more optimal than the last one, replace the prey's position with the current location; otherwise, keep the previous solution,

$$\begin{cases} [u] = min(fitness(Prey_{curr}), fitness(Prey_{last})) \\ Prey_i = Prey_u \end{cases}$$
$$i = 1, \ldots n \tag{9}$$

where u denotes the prey's location.

In the second phase, the position search distribution of prey could be simulated by Gaussian distribution. During the foraging process of ocean predators and prey, although the optimal individual will be found, the individual will continue to find a better position by conducting random searches near its position. The Gaussian distribution's definition is given by,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty \tag{10}$$

The normal distribution characteristics obtain the distribution density of individuals, when $\mu = 0$, $\sigma = 1$, and the normal distribution becomes the standard normal distribution. According to the literature [36], the standard normal distribution is also adopted. σ^2 and μ are the standard deviation of the search agents in solution space and the average value of total preys, respectively, so we also set $\sigma = 0$, $\mu = 1$. In the outpost mechanism, the muted individual is defined:

$$Prey_i{}' = Prey_i \otimes (1 + Gaus), \qquad i = 1, \ldots, n \tag{11}$$

and in equation (11), $Prey_i{}'$ and $Prey_i$ represent the i_{th} prey after Gaussian mutation and before mutation, respectively.

3.2 Differential Evolution Mutation with Simulated Annealing

3.2.1 Differential Evolution (DE)

Differential evolution (DE) algorithm is a classical and influential global optimization algorithm aiming at solving many optimization problems, proposed by R Storn and Price in 1997 [37]. The steps of DE are mutation, crossover, and selection, which are simple and effective. Because of the great power of DE, it has been successfully applied in some fields.

Like MPA, DE begins by generating an initial $N \times D$ matrix, and N and D denote the number of the population size and the dimension, respectively. The initialization formula of differential evolution is defined:

$$Z_i = Z_{min} + rand \times (Z_{max} - Z_{min}) \tag{12}$$

where the Z_{min} and Z_{max} mean the lower and upper bound of the individual. Z_i represents the i_{th} individual. $rand$ means the number randomly generated between 0 and 1. Then using the mutation operator, the solution is updated to M_i^t as the following equation:

$$M_i^t = Z_{r1}^t + F \times (Z_{r_2}^t - Z_{r_3}^t) \tag{13}$$

where Z_{r1}, Z_{r2}, and Z_{r3} are different individuals, and their selection applied to three randomly generated numbers in the range of 1 to n, where $r1 \neq r2 \neq r3 \neq i$. t represents the current iteration. F denotes the mutation factor, and is set to 0.5. Then, DE generates the solution V_i^t by introducing the crossover operator CR, and the equation is as follows:

$$V_i^t = \begin{cases} M_i^t & if(rand \leq CR) \\ Z_i^t & otherwise \end{cases} \tag{14}$$

where $rand \in [0, 1]$, and CR is the crossover probability, which determines whether the solution is mutated or not, and it is set to 0.2 according to the conventional DE. The last step to generate the solution by selection is dependent on the better fitness of Z_i^t and V_i^t, and the formula is shown as:

$$Z_i^{t+1} = \begin{cases} V_i^t & if(fit(V_i^t) < fit(Z_i^t)) \\ Z_i^t & otherwise \end{cases} \tag{15}$$

3.2.2 Simulated Annealing (SA)

The simulated annealing optimization is a classical and efficient algorithm that is easy to achieve, proposed by physicist S. Kirkpatrick et al. [38]. The inspiration for SA is from solid annealing. Firstly, to increase the internal energy of the solid, it should be heated to a very high temperature. As the temperature continues to grow, the solid turns into a disordered state. Then, when the temperature slowly decreases, the internal particles of the solid become orderly. Eventually, the solid achieves a state of internal energy equilibrium when the temperature drops to a minimum threshold.

SA simulates the slight decrease in the temperature of a high-temperature solid, aiming at reducing defects and minimizing the system energy. SA follows the Metropolis criterion [39]. That is, although the last solution's fitness value is more optimal, the current solution also has a certain probability of being accepted. The definition of simulated annealing is given by:

$$P = \begin{cases} 1 & if(fit(prey_{new}) \leq fit(prey)) \\ \exp(\frac{fit(prey - prey_{new})}{T}) & if(fit(prey_{new}) > fit(prey)) \end{cases} \tag{16}$$

where $fit(prey_{new})$ and $fit(prey)$ denotes the fitness of the newly generated solution after DE variation and present solution, respectively.

3.2.3 Process of the DE-SA Mechanism

DE-SA mechanism combines the differential evolution and the simulated annealing, and this hybrid improvement mechanism has the advantages of DE and SA. This mechanism can not only generate new competitive solutions but also has a certain probability of accepting the suboptimal solution, which significantly improves the MPA's exploration ability. Figure 2 depicts the DE-SA mechanism's detailed process for working in the MPA.

3.3 Tent Chaotic Map

Besides, ODMPA also introduces the chaotic tent map to increase the diversity of initialization population. Chaos is a common phenomenon in nonlinear systems. In a specific range, the change of chaotic variables has randomness, ergodicity, and regularity, and chaotic mapping operators significantly impact the optimization process. The tent map [40] has better ergodic uniformity than the logistic map and higher optimization efficiency. So the tent map is adopted to disturb the population after the initialization to generate a new $N \times D$ tent matrix and select the top N individuals from 2N individuals for iteration, increasing the initial population's variety. The mathematics formula for a tent map is as follows:

$$x_{i+1} = \begin{cases} x_i/\alpha & 0 \leq x_i \leq \alpha \\ (1 - x_i)(1 - \alpha) & \alpha < x_i \leq 1 \end{cases} \tag{17}$$

where $\alpha \in [0, 1]$, and the value of alpha affects the chaotic distribution, which in turn affects the generation of the initial population. Lots of experience shows that alpha taken between 0.55 and 0.56 may result in a more evenly distribution of chaotic values. Therefore, the cases were tested with the alpha of 0.55,0.555, and 0.56 in increments of 0.005, and the best results are achieved on CEC2014 benchmark functions when alpha is taken as 0.555.

3.4 The Proposed ODMPA

In this section, the outpost mechanism and DE-SA mechanism are introduced into the conventional MPA to construct an improved MPA, called ODMPA. Figure 3 demonstrates the detailed framework of ODMPA. In the improved ODMPA, there are three main improvements, tent chaotic map, outpost mechanism variation, and DE-SA variation. Firstly, the tent map to disturb the population after the initialization and then create a new matrix and select the top N individuals. This disturbance has a beneficial effect on the initial population and increases the diversity of solutions. Then, the outpost mechanism and DE-SA mechanism are sequentially introduced into the MPA. The outpost mechanism includes two main steps. Firstly, the prey compares the current fitness value with the last one; if it is worth moving forward, update the position; otherwise, keep the previous iteration's fitness. This step greatly enhances the convergence speed. Secondly, the prey should explore near the best value, and the motion is simulated by Gaussian mutation. After the variation of the outpost mechanism, the most important DE-SA mechanism is introduced. DE-SA variation could generate competitive new solutions by mutation, crossover, and selection. And because of the introduction of simulated annealing, it has also a certain probability of accepting the

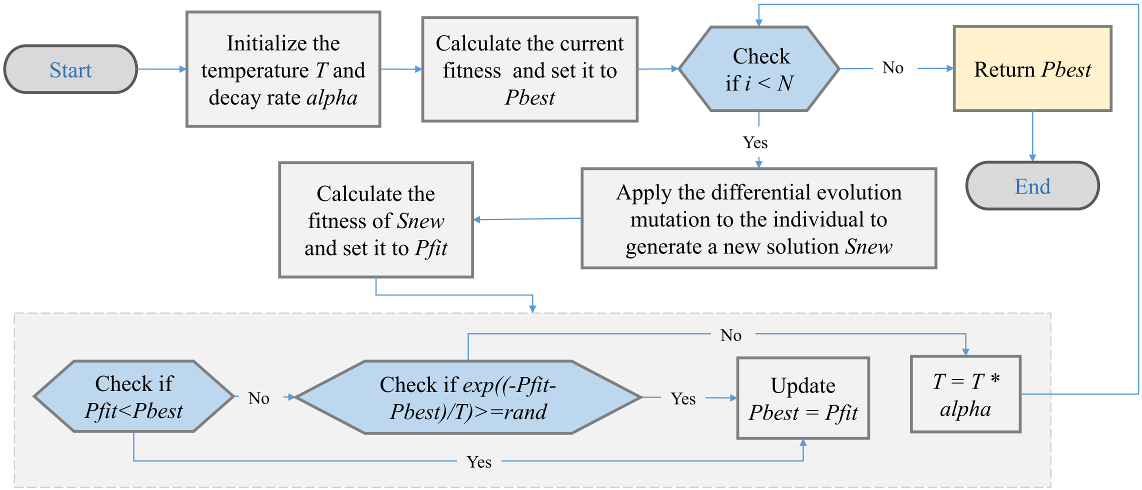

Start
Initialize the temperature *T* and decay rate *alpha*
Calculate the current fitness and set it to *Pbest*
Check if *i* < *N*
No
Return *Pbest*
End
Yes
Apply the differential evolution mutation to the individual to generate a new solution *Snew*
Calculate the fitness of *Snew* and set it to *Pfit*
No
Check if *Pfit*<*Pbest*
No
Check if *exp((-Pfit-Pbest)/T)>=rand*
Yes
Update *Pbest* = *Pfit*
T = *T* * *alpha*
Yes

Fig. 2 The detailed process of differential evolution mutation with simulated annealing

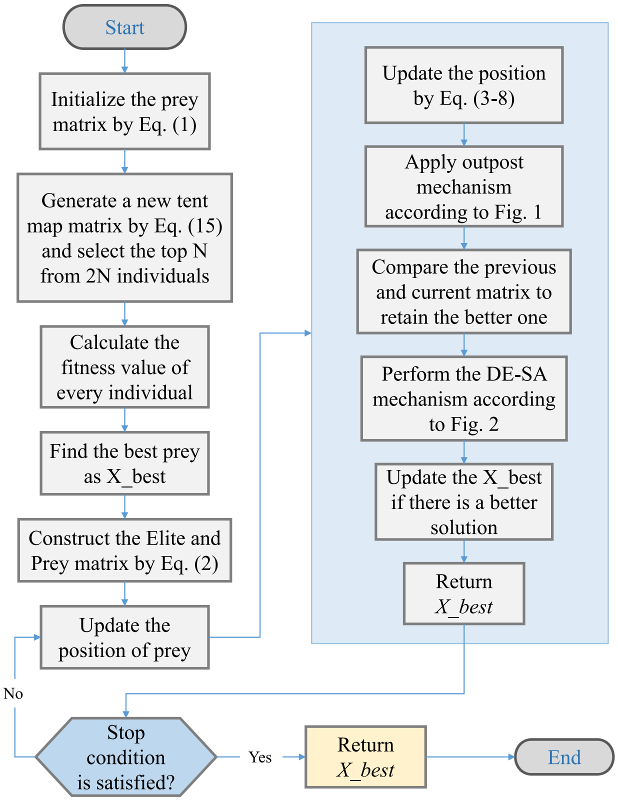

Start
Initialize the prey matrix by Eq. (1)
Generate a new tent map matrix by Eq. (15) and select the top N from 2N individuals
Calculate the fitness value of every individual
Find the best prey as X_best
Construct the Elite and Prey matrix by Eq. (2)
Update the position of prey
Update the position by Eq. (3-8)
Apply outpost mechanism according to Fig. 1
Compare the previous and current matrix to retain the better one
Perform the DE-SA mechanism according to Fig. 2
Update the X_best if there is a better solution
Return X_best
No
Stop condition is satisfied?
Yes
Return X_best
End

Fig. 3 The detailed framework of ODMPA

suboptimal solution, which enhances the solution's diversity and ability to escape off the optima.

4 Experiment Results and Analysis

In Sect. 4.1, ODMPA was compared to the CEC2014 [41] benchmark functions with other algorithms, including MPA, classical swarm intelligence algorithms, the latest swarm intelligence algorithms of the last two years, MPA variants, and advanced algorithms. The purpose of the comparison with the classical algorithms and improved MPA variants is to verify whether the ODMPA is improved or not. The comparison with the advanced algorithms is to demonstrate the effectiveness of ODMPA. In Sect. 4.2, ODMPA was validated on some engineering problems, including WBD [79], PVD [78], and SRD [80]. Moreover, in Sect. 4.3, ODMPA was adopted into the static and dynamic photovoltaic models. Engineering design problems and photovoltaic models are the real-world problems, compared to other counterparts, ODMPA could successfully tackle real-world problems in extracting the unknown parameters to get the minimum error. The results showed that ODMPA has great effectiveness in solving the practical problems.

For the results of the CEC2014 benchmark functions, the mean value result (Average) and standard deviation (Stdv) were adopted, validating the performance, and Table 4 lists the overall results. The Friedman test [42], the nonparametric statistical test, is used to assess the performance of ODMPA and the other swarm algorithms statistically. Moreover, the sign "+/=/−" results for each algorithm are calculated through the Friedman test. So through the overall information, the average ranking value (ARV) can be calculated, and ODMPA's ARV value is 4.55. The lowest ARV value indicates ODMPA's performance excels its counterparts. Moreover, all experiments were carried out on MATLAB R2016a, and the hardware platform is Core i7 CPU, 8 GB RAM.

4.1 IEEE CEC2014 Benchmark Functions

The benchmark functions are always a strong tool [82–86] to evaluate the effectiveness of optimization algorithms. IEEE

Table 1 Detail of the IEEE CEC2014 test functions

No	Class	Functions	$F_i^* = F_i(x^*)$
1	Unimodal	Rotated High Conditioned Elliptic Function	100
2	Unimodal	Rotated Bent Cigar Function	200
3	Unimodal	Rotated Discus Function	300
4	Multimodal	Shifted and Rotated Rosenbrock’s Function	400
5	Multimodal	Shifted and Rotated Ackley’s Function	500
6	Multimodal	Shifted and Rotated Weierstrass Function	600
7	Multimodal	Shifted and Rotated Griewank’s Function	700
8	Multimodal	Shifted Rastrigin’s Function	800
9	Multimodal	Shifted and Rotated Rastrigin’s Function	900
10	Multimodal	Shifted Schwefel’s Function	1000
11	Multimodal	Shifted and Rotated Schwefel’s Function	1100
12	Multimodal	Shifted and Rotated Katsuura Function	1200
13	Multimodal	Shifted and Rotated HappyCat Function	1300
14	Multimodal	Shifted and Rotated HGBat Function	1400
15	Multimodal	Shifted and Rotated Expanded Griewank’s plus Rosenbrock’s Function	1500
16	Multimodal	Shifted and Rotated Expanded Schaffer’s F6 Function	1600
17	Hybrid	Hybrid Function 1	1700
18	Hybrid	Hybrid Function 2	1800
19	Hybrid	Hybrid Function 3	1900
20	Hybrid	Hybrid Function 4	2000
21	Hybrid	Hybrid Function 5	2100
22	Hybrid	Hybrid Function 6	2200
23	Composition	Composition Function 1	2300
24	Composition	Composition Function 2	2400
25	Composition	Composition Function 3	2500
26	Composition	Composition Function 4	2600
27	Composition	Composition Function 5	2700
28	Composition	Composition Function 6	2800
29	Composition	Composition Function 7	2900
30	Composition	Composition Function 8	3000

CEC2014 benchmark functions is a standard test function set divided into four parts, which can well test the algorithm’s performance. Table 1 displays the details of CEC2014 benchmark functions. ODMPA is compared with the well-known advanced and original algorithms. The parameters were set to the same value to make sure the competition’s fairness. And the total experiments were carried on the same environment. The search agents scale and dimension are installed as 30, and the Max_Iter is 1000 based on the settings of the most experiments. And each algorithm was run for thirty independent trials to guarantee the fairness and reliability of the experiments. Table 2 displays the particular parameter settings of the conventional and advanced metaheuristics. Table 3 provides the average and standard deviation values of best-so-far solutions found in each run. The counterparts include MPA, classical swarm intelligence algorithms, the latest swarm intelligence algorithms of the last two years, MPA variants, and advanced algorithms. The comparison with these algorithms, especially with MPA variants and advanced algorithms, and the better results obtained by ODMPA prove the effectiveness of our improvements to MPA, all these algorithms are important.

- Differential Evolution Algorithm (DE)
- Gravitational Search Algorithm (GSA) [43]
- Multi-Verse Optimizer (MVO)
- Manta Ray Foraging Optimization (MRFO) [44]
- Marine Predators Algorithm (MPA)
- Harris Hawks Optimization (HHO) [45]
- Pathfinder Algorithm (PFA) [46]
- Sine Cosine Algorithm (SCA) [47]
- Slime Mould Algorithm (SMA) [48]

Table 2 The parameter settings of the conventional and advanced metaheuristics

Algorithm	Settings
ODMPA	$FADs$=0.2; fai=0.555; $T_0 = 0.025$; $\alpha = 0.99$; P=0.5; $CF \in [1, 0]$
DE	CR=0.2; Mutation factor=[0.2,0.8]
GSA	$\alpha = 20$; G_0=100; $T = 1000$; $K_0 \in [10001]$
GWO	$\alpha \in [20]$
HHO	$E_0 \in [-11]$; $u, v \in [01]$; $beta$=1.5
MPA	P=0.5; $FADs$=0.2; $CF \in [1, 0]$
MRFO	$beta, r, r_1, r_2, r_3 \in [01]$; T=1000; S=2
MVO	$WEP \in [0.21]$; $TDR \in [0.60]$
PFA	$\sim$
SCA	$A = 2$
SMA	$vc = [10]$; $z \in [00.1]$
SSA	$prodP = 0.2$; $threat = 0.1$; $ST = 0.8$
SSAsalp	$c_1, c_2 \in [01]$
TSA	$P_{\min} = 1$; $P_{\max} = 4$
WOA	$a_1 = [20]$; $a_2 = [-2-1]$; $b = 1$
AVOA	$p_1 = 0.6$; $\alpha = 0.8$; $\beta = 0.2$; $p_2 = 0.4$; $\gamma = 2.5$; $p_3 = 0.6$;
Ex-GWO	$\alpha \in [20]$
MMPA	$P = 0.5$; $FADs = 0.2$; $\mu = 4$
COOT	$R_1, R_2, R_3, R_4 \in [01]$; $R \in [-11]$
EMPA	$P = 0.5$; $FADs = 0.2$; $CF \in [1, 0]$
IMPA	$P = 0.5$; $FADs = 0.2$; $CF \in [1, 0]$

- Sparrow Search Algorithm (SSA) [49]
- Grey Wolf Optimizer (GWO)
- Tunicate Swarm Algorithm (TSA) [50]
- Salp Swarm Algorithm (SSAsalp)
- Whale Optimization Algorithm (WOA) [51]
- COOT Bird Optimization Method (COOT) [52]
- African Vultures Optimization Algorithm (AVOA) [53]
- Expanded GWO (Ex-GWO) [54]
- Modified Self-adaptive Marine Predators Algorithm (MMPA) [55]
- Enhanced Marine Predators Algorithm (EMPA)
- Improved Marine Predators Algorithm (IMPA) [56]

Table 3 displays the mean value (Average) and standard deviation (Stdv) results. The results show that ODMPA has obtained the most minimum value in general, which shows that ODMPA can perform better than other algorithms in the precision of the solution. The minimum values are bolded in the table. Friedman's rank test fully uses the complete information of the average fitness values of the obtained algorithms to perform nonparametric tests and generate a combined ranking of the ARV values of all algorithms. The ARV values are calculated by SPSS and are shown in Table 4. ODMPA ranks first and has the lowest ARV value, which verifies that ODMPA displays best in finding the optimal solution.

Figures 4, 5, and 6 show the iterative curve graphs. F1–F3 unimodal functions' properties decide they do not have the sub-optima, so they are suitable for evaluating the elementary performance of ODMPA. F1, F2, and F3 functions' optimal values by ODMPA are closer to the global optima than the original MPA. The numerical results show that the search ability and convergence speed of the ODMPA were vastly enhanced on unimodal functions. The differential evolution mutation with simulated annealing mechanism helps to extend new feasible solutions in the threshold range. The outpost mechanism enhances the convergence efficiency. Tent mapping in the initialization phase makes the initial randomization of the population closer to the optima. In F4 and F7 functions, almost all the algorithms' best values are close to the optima. Moreover, ODMPA's convergence speed is relatively faster than the original MPA and most algorithms. In F8 and F9 functions, while the convergence speed of ODMPA is slower in earlier iterations, as the iteration increases, ODMPA finally reaches the global optimal value. For the F12 function, ODMPA's convergence speed and optimal value are better than other counterparts.

F18 and F19 are taken as the representative functions for the hybrid functions. From the iterative graphs of F18 and F19, it can be derived that ODMPA has the best capacity to find the best optimal value. Moreover, for the composition functions, F23 and F24 also have the same phenomenon.

Table 3 Results of ODMPA and other algorithms on IEEE CEC2014 functions

	F1		F2		F3	
	Average	Stdv	Average	Stdv	Average	Stdv
ODMPA	4.477E+06	3.800E+06	4.460E+04	3.970E+04	1.789E+03	1.470E+03
DE	5.963E+07	1.390E+07	6.311E+05	2.010E+05	**8.637E+02**	3.910E+02
GSA	3.555E+08	8.290E+07	2.104E+10	4.770E+09	9.011E+04	8.420E+03
GWO	2.844E+08	1.340E+08	1.272E+10	3.570E+09	8.822E+04	2.030E+03
HHO	5.333E+07	2.170E+07	6.357E+07	1.890E+07	4.103E+04	7.860E+03
MPA	3.479E+07	1.970E+07	2.298E+05	2.220E+05	2.451E+04	1.940E+04
MRFO	**1.717E+06**	1.080E+06	**2.355E+03**	1.950E+03	3.472E+03	4.320E+03
MVO	8.896E+06	5.340E+06	6.343E+05	2.250E+05	2.335E+03	1.560E+03
PFA	3.494E+07	1.650E+07	4.046E+07	2.840E+07	4.293E+04	1.770E+04
SCA	1.059E+09	4.290E+08	4.205E+10	7.290E+09	6.998E+05	1.060E+06
SMA	6.874E+06	3.630E+06	7.695E+05	4.290E+05	7.332E+03	5.780E+03
SSA	3.930E+06	2.300E+06	1.043E+04	9.220E+03	1.354E+04	1.190E+04
SSAsalp	2.469E+07	9.790E+06	1.212E+04	1.180E+04	2.378E+05	7.080E+04
TSA	2.074E+09	1.060E+08	8.749E+10	3.440E+09	2.847E+05	3.000E+05
WOA	1.548E+08	6.380E+07	2.161E+09	9.720E+08	7.832E+04	4.340E+04
AVOA	1.385E+07	7.900E+06	1.524E+04	1.210E+04	2.172E+04	6.380E+03
COOT	1.279E+08	5.440E+07	1.077E+09	5.700E+08	5.463E+04	1.060E+04
Ex-GWO	1.983E+09	2.860E+04	1.026E+11	2.080E+03	8.989E+04	1.520E+03
MMPA	1.420E+07	6.590E+06	3.257E+05	2.880E+05	7.321E+03	4.610E+03
EMPA	3.486E+07	1.910E+07	1.792E+05	9.420E+04	2.116E+04	1.460E+04
IMPA	3.238E+07	2.280E+07	1.882E+05	3.010E+05	2.322E+04	1.860E+04
	F4		F5		F6	
	Average	Stdv	Average	Stdv	Average	Stdv
ODMPA	**4.968E+02**	2.400E+01	5.213E+02	3.060E-01	6.172E+02	3.890E+00
DE	5.485E+02	1.930E+01	5.207E+02	6.480E-02	6.261E+02	1.570E+00
GSA	2.649E+03	5.340E+02	5.200E+02	3.270E-04	6.306E+02	2.600E+00
GWO	1.264E+03	4.850E+02	5.210E+02	1.220E-01	6.302E+02	2.840E+00
HHO	6.316E+02	5.780E+01	5.207E+02	1.400E-01	6.342E+02	3.220E+00
MPA	5.811E+02	4.660E+01	5.201E+02	3.290E-02	6.260E+02	5.210E+00
MRFO	4.980E+02	3.840E+01	5.209E+02	1.750E-01	6.232E+02	4.650E+00
MVO	5.099E+02	3.610E+01	5.205E+02	1.010E-01	6.170E+02	3.260E+00
PFA	5.800E+02	5.310E+01	5.210E+02	5.700E-02	6.257E+02	4.220E+00
SCA	6.158E+03	1.970E+03	5.213E+02	8.490E-02	6.429E+02	2.500E+00
SMA	5.050E+02	3.630E+01	5.215E+02	9.840E-02	**6.130E+02**	2.680E+00
SSA	5.100E+02	3.290E+01	5.200E+02	4.960E-02	6.292E+02	4.200E+00
SSAsalp	5.273E+02	2.850E+01	**5.200E+02**	4.100E-05	6.162E+02	3.580E+00
TSA	1.682E+04	1.730E+03	5.216E+02	5.540E-02	6.493E+02	1.430E+00
WOA	8.759E+02	1.590E+02	5.211E+02	1.330E-01	6.354E+02	3.620E+00
AVOA	5.420E+02	3.560E+01	5.201E+02	9.680E-02	6.247E+02	3.040E+00
COOT	6.864E+02	6.470E+01	5.210E+02	1.040E-01	6.332E+02	4.290E+00
Ex-GWO	2.404E+04	1.090E-01	5.216E+02	1.920E-05	6.520E+02	1.630E-01
MMPA	5.350E+02	4.100E+01	5.203E+02	1.210E-01	6.158E+02	3.570E+00
EMPA	5.836E+02	8.640E+01	5.201E+02	4.290E-02	6.258E+02	3.270E+00
IMPA	5.990E+02	7.620E+01	5.201E+02	6.780E-02	6.263E+02	3.350E+00

Table 3 continued

	F7		F8		F9		F10		F11		F12	
	Average	Stdv	Average	Stdv	Average	Stdv	Average	Stdv	Average	Stdv	Average	Stdv
ODMPA	7.003E+02	1.350E-01	**8.511E+02**	1.660E+01	**9.523E+02**	1.730E+01	2.504E+03	4.710E+02	**3.985E+03**	5.110E+02	1.200E+03	2.470E-01
DE	7.004E+02	8.050E-02	8.818E+02	9.010E+00	1.115E+03	1.940E+01	3.555E+03	3.040E+02	6.656E+03	2.550E+02	1.201E+03	1.960E-01
GSA	9.252E+02	4.190E+01	9.578E+02	1.320E+01	1.083E+03	1.610E+01	5.176E+03	5.340E+02	5.697E+03	5.960E+02	**1.200E+03**	6.960E-03
GWO	7.902E+02	2.860E+01	1.088E+03	2.230E+01	1.186E+03	2.620E+01	6.883E+03	6.520E+02	7.448E+03	9.200E+02	1.202E+03	6.220E-01
HHO	7.019E+02	3.610E-01	9.448E+02	1.170E+01	1.099E+03	2.020E+01	4.443E+03	4.970E+02	5.746E+03	7.210E+02	1.202E+03	3.970E-01
MPA	7.004E+02	1.820E-01	9.189E+02	3.600E+01	1.083E+03	4.980E+01	4.674E+03	5.700E+02	5.239E+03	7.310E+02	1.200E+03	1.870E-01
MRFO	7.000E+02	1.730E-02	9.415E+02	2.280E+01	1.072E+03	2.370E+01	3.836E+03	8.800E+02	5.048E+03	6.960E+02	1.202E+03	8.190E-01
MVO	7.007E+02	9.390E-02	8.581E+02	1.500E+01	1.050E+03	3.930E+01	2.717E+03	3.810E+02	4.508E+03	7.110E+02	1.201E+03	2.660E-01
PFA	7.014E+02	2.520E-01	9.679E+02	3.940E+01	1.138E+03	6.630E+01	4.468E+03	7.310E+02	5.372E+03	6.100E+02	1.203E+03	5.460E-01
SCA	1.100E+03	6.340E+01	1.164E+03	3.320E+01	1.295E+03	3.640E+01	9.790E+03	5.220E+02	1.042E+04	4.640E+02	1.206E+03	1.140E+00
SMA	7.012E+02	9.970E-02	8.556E+02	1.300E+01	1.000E+03	2.640E+01	2.168E+03	3.870E+02	4.012E+03	4.230E+02	1.201E+03	1.990E-01
SSA	7.000E+02	1.340E-02	9.546E+02	2.290E+01	1.097E+03	1.510E+01	4.568E+03	6.560E+02	5.620E+03	8.520E+02	1.202E+03	7.850E-01
SSAsalp	**7.000E+02**	1.110E-02	8.680E+02	1.550E+01	9.904E+02	2.570E+01	4.233E+03	7.060E+02	4.473E+03	6.530E+02	1.200E+03	1.710E-01
TSA	1.529E+03	5.160E+01	1.319E+03	7.140E+00	1.333E+03	2.610E+01	1.133E+04	3.210E+02	1.246E+04	5.040E+02	1.206E+03	1.290E+00
WOA	7.115E+02	2.880E+00	1.016E+03	5.030E+01	1.163E+03	5.870E+01	5.876E+03	6.960E+02	7.005E+03	6.740E+02	1.202E+03	6.920E-01
AVOA	7.000E+02	3.180E-02	8.919E+02	2.240E+01	1.061E+03	2.710E+01	3.058E+03	5.460E+02	5.029E+03	7.730E+02	1.201E+03	2.260E-01
COOT	7.110E+02	3.980E+00	9.846E+02	3.420E+01	1.124E+03	3.780E+01	5.552E+03	6.950E+02	6.905E+03	7.220E+02	1.203E+03	7.040E-01
Ex-GWO	1.619E+03	8.910E-04	1.205E+03	3.120E+00	1.378E+03	1.590E-05	1.122E+04	7.650E+01	1.290E+04	2.730E+02	1.208E+03	3.990E-01
MMPA	7.005E+02	1.950E-01	8.627E+02	1.580E+01	1.040E+03	2.480E+01	**2.130E+03**	3.670E+02	4.343E+03	8.140E+02	1.200E+03	1.430E-01
EMPA	7.004E+02	1.760E-01	9.064E+02	2.590E+01	1.072E+03	5.560E+01	4.544E+03	8.040E+02	5.196E+03	7.100E+02	1.200E+03	1.350E-01
IMPA	7.004E+02	1.680E-01	9.276E+02	3.490E+01	1.077E+03	4.860E+01	4.776E+03	7.800E+02	5.108E+03	1.010E+03	1.200E+03	1.050E-01

Table 3 continued

	F13		F14		F15	
	Average	Stdv	Average	Stdv	Average	Stdv
ODMPA	**1.301E+03**	6.380E-02	**1.400E+03**	3.490E-02	**1.508E+03**	2.080E+00
DE	1.301E+03	1.890E+00	1.400E+03	2.630E-01	1.520E+03	1.770E+00
GSA	1.304E+03	4.420E-01	1.492E+03	1.530E+01	2.797E+03	1.030E+03
GWO	1.302E+03	6.540E-01	1.432E+03	9.690E+00	4.554E+03	3.470E+03
HHO	1.301E+03	1.170E-01	1.400E+03	6.360E-02	1.547E+03	9.810E+00
MPA	1.301E+03	1.250E-01	1.401E+03	3.740E-01	1.516E+03	5.930E+00
MRFO	1.301E+03	1.110E-01	1.400E+03	1.930E-01	1.516E+03	6.080E+00
MVO	1.301E+03	1.270E-01	1.401E+03	3.340E-01	1.512E+03	3.050E+00
PFA	1.301E+03	1.340E-01	1.401E+03	3.240E-01	1.531E+03	7.470E+00
SCA	1.305E+03	4.950E-01	1.545E+03	2.160E+01	2.180E+05	1.980E+05
SMA	1.301E+03	1.250E-01	1.401E+03	3.730E-01	1.513E+03	2.290E+00
SSA	1.301E+03	1.490E-01	1.400E+03	1.630E-01	1.542E+03	1.900E+01
SSAsalp	1.301E+03	1.030E-01	1.400E+03	2.050E-01	1.509E+03	2.960E+00
TSA	1.309E+03	4.610E-01	1.686E+03	2.460E+01	4.442E+05	1.230E+05
WOA	1.301E+03	2.860E-01	1.403E+03	4.440E+00	1.860E+03	7.840E+02
AVOA	1.301E+03	1.080E-01	1.400E+03	1.000E-01	1.512E+03	4.000E+00
COOT	1.301E+03	1.490E-01	1.401E+03	1.050E+00	1.605E+03	8.530E+01
Ex-GWO	1.311E+03	2.230E-07	1.706E+03	4.360E-03	2.763E+05	1.800E+01
MMPA	1.301E+03	1.370E-01	1.400E+03	2.610E-01	1.511E+03	2.820E+00
EMPA	1.301E+03	1.430E-01	1.401E+03	3.650E-01	1.518E+03	5.780E+00
IMPA	1.301E+03	1.120E-01	1.401E+03	3.630E-01	1.521E+03	1.260E+01

	F16		F17		F18	
	Average	Stdv	Average	Stdv	Average	Stdv
ODMPA	1.615E+03	4.970E-01	**2.625E+04**	2.620E+04	**2.412E+03**	1.270E+03
DE	1.612E+03	3.250E-01	3.050E+06	1.060E+06	8.277E+04	5.270E+04
GSA	1.614E+03	2.400E-01	2.821E+07	8.030E+06	1.062E+07	5.090E+07
GWO	1.614E+03	3.350E-01	1.456E+07	7.380E+06	2.181E+08	4.300E+07
HHO	1.613E+03	3.400E-01	6.026E+06	4.150E+06	3.063E+05	1.620E+05
MPA	1.613E+03	4.440E-01	3.414E+06	2.360E+06	2.671E+04	5.090E+03
MRFO	**1.612E+03**	5.650E-01	2.101E+05	1.410E+05	6.979E+03	6.260E+03
MVO	1.612E+03	6.070E-01	6.764E+05	5.270E+05	9.742E+03	6.130E+03
PFA	1.613E+03	2.690E-01	1.881E+06	1.450E+06	4.628E+04	6.210E+04
SCA	1.614E+03	2.710E-01	1.138E+08	6.780E+07	1.301E+09	4.820E+08
SMA	1.613E+03	5.860E-01	7.205E+05	4.740E+05	1.612E+04	1.100E+04
SSA	1.613E+03	4.160E-01	8.682E+05	6.190E+05	5.287E+03	4.210E+03
SSAsalp	1.612E+03	7.360E-01	1.639E+06	9.690E+05	1.260E+04	8.440E+03
TSA	1.615E+03	1.810E-01	4.033E+08	1.210E+08	8.787E+09	1.650E+09
WOA	1.613E+03	4.240E-01	1.587E+07	1.120E+07	1.188E+06	1.180E+06
AVOA	1.613E+03	5.900E-01	2.219E+06	1.630E+06	5.481E+03	4.220E+03
COOT	1.613E+03	3.070E-01	6.259E+06	4.660E+06	2.470E+05	2.840E+05
Ex-GWO	1.615E+03	3.170E-01	3.889E+08	8.250E+03	1.478E+10	3.590E+04
MMPA	1.613E+03	5.300E-01	1.216E+06	9.870E+05	4.070E+03	2.480E+03
EMPA	1.613E+03	5.020E-01	2.772E+06	1.410E+06	2.690E+04	3.800E+03
IMPA	1.613E+03	6.620E-01	3.411E+06	2.200E+06	2.667E+04	5.180E+03

Table 3 continued

	F19		F20		F21	
	Average	Stdv	Average	Stdv	Average	Stdv
ODMPA	**1.910E+03**	2.280E+00	**2.213E+03**	1.800E+02	**5.187E+03**	3.450E+03
DE	1.912E+03	8.950E-01	1.199E+04	4.280E+03	6.267E+05	3.170E+05
GSA	2.082E+03	3.540E+01	4.404E+05	2.780E+05	1.376E+07	5.060E+06
GWO	2.027E+03	2.480E+01	3.361E+05	8.110E+04	1.032E+07	6.760E+06
HHO	1.937E+03	3.650E+01	3.199E+04	1.040E+04	1.079E+06	1.050E+06
MPA	1.942E+03	3.940E+01	3.593E+04	1.980E+04	8.561E+05	8.410E+05
MRFO	1.921E+03	2.290E+01	1.485E+04	9.820E+03	1.080E+05	1.250E+05
MVO	1.917E+03	1.760E+01	5.290E+03	3.510E+03	2.584E+05	2.270E+05
PFA	1.920E+03	1.780E+01	2.747E+04	1.620E+04	5.185E+05	3.810E+05
SCA	2.219E+03	8.820E+01	4.237E+06	3.760E+06	3.436E+07	2.070E+07
SMA	1.912E+03	1.990E+00	2.900E+04	1.750E+04	5.202E+05	4.380E+05
SSA	1.923E+03	2.630E+01	2.768E+04	1.970E+04	4.634E+05	3.730E+05
SSAsalp	1.917E+03	1.220E+01	1.088E+05	4.570E+04	3.898E+05	2.730E+05
TSA	2.418E+03	1.390E+02	5.416E+07	1.030E+08	3.493E+08	2.380E+08
WOA	1.985E+03	3.950E+01	1.528E+05	3.590E+05	8.380E+06	8.690E+06
AVOA	1.922E+03	2.450E+01	2.685E+04	1.330E+04	5.863E+05	4.640E+05
COOT	1.950E+03	3.310E+01	5.634E+04	3.160E+04	1.983E+06	2.090E+06
Ex-GWO	2.803E+03	2.080E-02	1.628E+07	6.720E+04	1.758E+09	4.450E+04
MMPA	1.913E+03	1.060E+01	2.076E+04	1.000E+04	5.556E+05	3.830E+05
EMPA	1.950E+03	3.530E+01	3.190E+04	1.880E+04	7.906E+05	8.220E+05
IMPA	1.941E+03	3.810E+01	3.318E+04	1.910E+04	9.539E+05	8.980E+05

	F22		F23		F24	
	Average	Stdv	Average	Stdv	Average	Stdv
ODMPA	**2.462E+03**	1.560E+02	**2.500E+03**	0.000E+00	**2.600E+03**	0.000E+00
DE	2.623E+03	1.300E+02	2.616E+03	7.350E-02	2.634E+03	1.780E+00
GSA	3.659E+03	4.140E+02	2.631E+03	1.100E+02	2.621E+03	7.280E+00
GWO	3.256E+03	3.040E+02	2.701E+03	1.490E+01	2.600E+03	6.630E-04
HHO	3.134E+03	2.480E+02	2.500E+03	0.000E+00	2.600E+03	0.000E+00
MPA	3.120E+03	3.000E+02	2.630E+03	9.910E+00	2.648E+03	8.730E+00
MRFO	2.864E+03	1.850E+02	2.500E+03	0.000E+00	2.600E+03	0.000E+00
MVO	2.859E+03	2.290E+02	2.617E+03	1.130E+00	2.634E+03	6.960E+00
PFA	2.850E+03	2.190E+02	2.617E+03	8.030E-01	2.641E+03	1.030E+01
SCA	4.157E+03	3.710E+02	2.953E+03	8.770E+01	2.622E+03	3.050E+01
SMA	2.690E+03	2.330E+02	2.500E+03	0.000E+00	2.600E+03	0.000E+00
SSA	3.102E+03	3.530E+02	2.500E+03	2.590E-11	2.600E+03	1.190E-02
SSAsalp	2.656E+03	1.880E+02	2.632E+03	9.740E+00	2.642E+03	5.470E+00
TSA	3.988E+05	2.000E+05	2.500E+03	0.000E+00	2.600E+03	0.000E+00
WOA	3.088E+03	2.940E+02	2.682E+03	2.030E+01	2.606E+03	3.200E+00
AVOA	3.003E+03	2.180E+02	2.500E+03	0.000E+00	2.600E+03	0.000E+00
COOT	3.090E+03	2.310E+02	2.610E+03	6.220E+01	2.600E+03	1.460E-06
Ex-GWO	4.457E+06	6.850E+02	2.500E+03	0.000E+00	2.600E+03	0.000E+00
MMPA	3.051E+03	2.150E+02	2.500E+03	1.720E-03	2.600E+03	1.830E-03
EMPA	3.188E+03	2.670E+02	2.634E+03	1.010E+01	2.649E+03	7.860E+00
IMPA	3.142E+03	2.870E+02	2.632E+03	9.570E+00	2.649E+03	8.750E+00

Table 3 continued

	F25		F26		F27	
	Average	Stdv	Average	Stdv	Average	Stdv
ODMPA	2.700E+03	9.190E-01	2.760E+03	4.960E+01	**2.900E+03**	0.000E+00
DE	2.715E+03	2.020E+00	2.701E+03	1.320E-01	3.152E+03	5.410E+01
GSA	2.708E+03	1.920E+00	2.800E+03	2.020E-02	6.186E+03	9.440E+02
GWO	2.700E+03	1.690E-13	2.751E+03	5.020E+01	3.639E+03	2.150E+02
HHO	**2.700E+03**	0.000E+00	2.777E+03	4.280E+01	2.900E+03	0.000E+00
MPA	2.714E+03	4.630E+00	2.701E+03	1.210E-01	3.610E+03	2.820E+02
MRFO	2.700E+03	0.000E+00	2.701E+03	1.070E-01	2.900E+03	0.000E+00
MVO	2.708E+03	2.740E+00	2.779E+03	7.120E+01	3.455E+03	1.770E+02
PFA	2.716E+03	5.170E+00	2.749E+03	9.240E+01	3.512E+03	2.710E+02
SCA	2.758E+03	3.980E+01	2.707E+03	2.780E+00	4.184E+03	8.230E+01
SMA	2.700E+03	0.000E+00	2.701E+03	1.390E-01	2.900E+03	0.000E+00
SSA	2.700E+03	1.920E-12	2.764E+03	4.870E+01	2.900E+03	1.450E-11
SSAsalp	2.719E+03	6.650E+00	2.701E+03	1.210E-01	3.458E+03	7.390E+01
TSA	2.700E+03	0.000E+00	2.800E+03	0.000E+00	2.917E+03	9.540E+01
WOA	2.718E+03	2.010E+01	2.727E+03	4.480E+01	3.774E+03	3.340E+02
AVOA	2.700E+03	0.000E+00	2.744E+03	5.020E+01	2.900E+03	0.000E+00
COOT	2.700E+03	0.000E+00	2.757E+03	5.010E+01	3.490E+03	4.180E+02
Ex-GWO	2.700E+03	0.000E+00	2.800E+03	0.000E+00	2.900E+03	0.000E+00
MMPA	2.700E+03	1.610E-05	2.787E+03	3.440E+01	2.900E+03	2.340E-04
EMPA	2.715E+03	5.980E+00	**2.701E+03**	1.230E-01	3.648E+03	2.540E+02
IMPA	2.718E+03	5.340E+00	2.701E+03	1.230E-01	3.550E+03	2.920E+02

	F28		F29		F30	
	Average	Stdv	Average	Stdv	Average	Stdv
ODMPA	**3.000E+03**	0.000E+00	3.267E+03	5.650E+02	4.742E+03	1.640E+03
DE	3.795E+03	3.020E+01	1.329E+04	4.780E+03	1.176E+04	2.040E+03
GSA	5.980E+03	8.260E+02	1.319E+08	1.610E+08	1.999E+06	4.730E+05
GWO	5.387E+03	1.080E+03	4.148E+07	3.040E+07	2.081E+06	6.990E+05
HHO	3.000E+03	0.000E+00	2.279E+05	1.230E+06	2.641E+04	3.900E+04
MPA	4.109E+03	2.740E+02	6.965E+06	7.670E+06	5.957E+04	5.160E+04
MRFO	3.000E+03	0.000E+00	3.293E+03	6.170E+02	8.012E+03	1.090E+04
MVO	4.539E+03	6.220E+02	1.331E+06	3.440E+06	9.007E+03	2.210E+03
PFA	4.058E+03	4.240E+02	1.957E+06	4.540E+06	1.734E+04	7.390E+03
SCA	9.140E+03	9.720E+02	2.228E+08	8.060E+07	3.341E+06	1.600E+06
SMA	3.000E+03	0.000E+00	3.130E+05	1.680E+06	6.860E+03	1.380E+03
SSA	3.000E+03	2.540E-11	3.100E+03	2.280E-04	**3.200E+03**	6.950E-03
SSAsalp	3.777E+03	5.930E+01	2.249E+04	6.870E+03	3.530E+04	1.530E+04
TSA	3.000E+03	0.000E+00	**3.100E+03**	0.000E+00	3.365E+06	6.940E+06
WOA	5.394E+03	5.140E+02	1.466E+07	1.520E+07	2.896E+05	1.470E+05
AVOA	3.000E+03	0.000E+00	5.231E+03	4.490E+03	1.407E+04	1.250E+04
COOT	4.621E+03	7.410E+02	6.534E+06	7.660E+06	8.692E+04	4.190E+04
Ex-GWO	3.000E+03	0.000E+00	3.100E+03	0.000E+00	3.200E+03	0.000E+00
MMPA	3.000E+03	6.870E-04	4.020E+03	1.930E+03	3.229E+03	8.450E+01
EMPA	4.014E+03	1.910E+02	7.504E+06	5.850E+06	6.408E+04	5.640E+04
IMPA	4.133E+03	3.130E+02	7.236E+06	6.830E+06	5.331E+04	5.320E+04

Bold values indicate the minimum values calculated by all the algorithms of CEC2014 test set

Table 4 Over rank off ODMPA and other algorithms by Friedman assessment

Algorithm	Rank	+/−/=	ARV
ODMPA	**1**	∼	**4.55**
DE	9	24/0/6	9.2
GSA	17	11/15/04	15.92
GWO	18	11/17/02	16.12
HHO	10	18/0/12	10.72
MPA	11	1/13/16	10.98
MRFO	2	23/0/7	5.63
MVO	8	1/8/21	7.58
PFA	14	3/14/13	11.83
SCA	21	0/15/15	18.8
SMA	5	2/4/24	6.82
SSA	7	3/6/21	7.45
SSAsalp	6	19/0/11	7.4
TSA	20	6/16/8	17.35
WOA	16	6/18/06	15.65
AVOA	4	25/0/5	6.35
COOT	15	2/20/08	13.85
Ex-GWO	19	7/15/8	16.65
MMPA	3	1/4/25	5.72
EMPA	12	1/11/18	11.02
IMPA	13	2/11/17	11.42

Bold value indicates the lowest ARV

These results prove that ODMPA can obtain excellent results on complex optimization problems.

4.2 ODMPA for Engineering Design Problems

The engineering application problem is also the mainstream test problem of the proposed algorithm, which can better test the ability of the algorithm to solve real-world problems. It is vital to optimizing the problems with a constrained search space, and these problems always belong to the engineering domain. ODMPA is adapted to three real-life engineering design problems with constraints, including pressure vessel design (PVD) problem [78], welded beam design (WBD) [79] problem, and speed reducer design (SRD)[80] problem. There are numerous constraints in the mathematics model of engineering problems, so choosing a proper approach to deal with them is essential. Therefore, the penalty function is added to the original objective function to obtain an augmented objective function. The work [57] mentioned many types of penalty functions, including segregated GA functions, death penalty functions, static penalty functions, etc. In solving engineering problems, a simple and practical approach should be taken to ensure there is no infeasible solution in the search range. Thus, the death penalty function is the most suitable for engineering design problems. Moreover, the penalty factor weighs the loss and is usually specified as a large positive number.

4.2.1 Welded Beam Design Problem

The welded beam is a continuous engineering optimization problem, and whose main target is to minimize the cost of the welded beam. The point is to find an optimal set of dimensions: the height of beam (t), the width of beam (b), the thickness of beam (h), the length of beam (l). The seven constraints, the definition of WBD problem, are given by Eq. 18.

$$
\begin{aligned}
&\overrightarrow{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b],\\
&\text{Minimize} : f(\overrightarrow{x}) = 1.10471x_2x_1^2 + 0.04811x_3x_4(14 + x_2),\\
&\text{Subjectto} :\\
&g_1(\overrightarrow{x}) = \tau(\overrightarrow{x}) - \tau_{\max} \le 0\\
&g_2(\overrightarrow{x}) = \sigma(\overrightarrow{x}) - \sigma_{\max} \le 0\\
&g_3(\overrightarrow{x}) = \delta(\overrightarrow{x}) - \delta_{\max} \le 0\\
&g_4(\overrightarrow{x}) = x_1 - x_4 \le 0\\
&g_5(\overrightarrow{x}) = P - P_c(\overrightarrow{x}) \le 0\\
&g_6(\overrightarrow{x}) = 0.125 - x_1 \le 0\\
&g_7(\overrightarrow{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0,\\
&Variablerange 0.1{\le}x_1{\le}2,\ 0.1{\le}x_2{\le}10,\ 0.1{\le}x_3{\le}10,\ 0.1{\le}x_4{\le}2,\\
&where \tau(\overrightarrow{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\tfrac{x_2}{2R} + (\tau'')^2},\\
&\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P(L + \tfrac{x_2}{2}),\\
&R = \sqrt{\tfrac{x_2^2}{4} + \left(\tfrac{x_1+x_3}{2}\right)^2},\\
&J = 2\left\{\sqrt{2}x_1x_2\left[\tfrac{x_2^2}{4} + \left(\tfrac{x_1+x_3}{2}\right)^2\right]\right\},\\
&\sigma(\overrightarrow{x}) = \frac{6PL}{x_4x_3^2}, \delta(\overrightarrow{x}) = \frac{4PL^3}{Ex_4x_3^2},\\
&P_c(\overrightarrow{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),\\
&P = 6000lb, L = 14in., \delta_{\max} = 0.25in., E = 30 \times 10^6 psi,\\
&G = 12 \times 10^6 psi, \tau_{\max} = 13600 psi, \sigma_{\max} = 30000 psi.
\end{aligned}
\tag{18}
$$

A lot of algorithms have solved the WBD problem and obtained good results. The comparison results to other famous traditional and advanced algorithms, including BA [58], RO [59], GSA, IHS [60], hHHO-SCA [61], MTSA [62], CPSO [63], are displayed in Table 5. The results demonstrate that ODMPA has a more excellent performance than other algorithms, and the best cost is 1.7017.

4.2.2 Pressure Vessel Design Problem

The main target of this engineering problem is aimed at minimizing the total cost pf the pressure vessel. The PVD problem has four variables: thickness of the shell (T_s), thickness of the head (T_h), internal range (R), and length of the shell (L). The four constraints and the structural formulation are given by Eq. 19.

Fig. 4 The iteration curve of ODMPA on CEC2014 benchmark functions

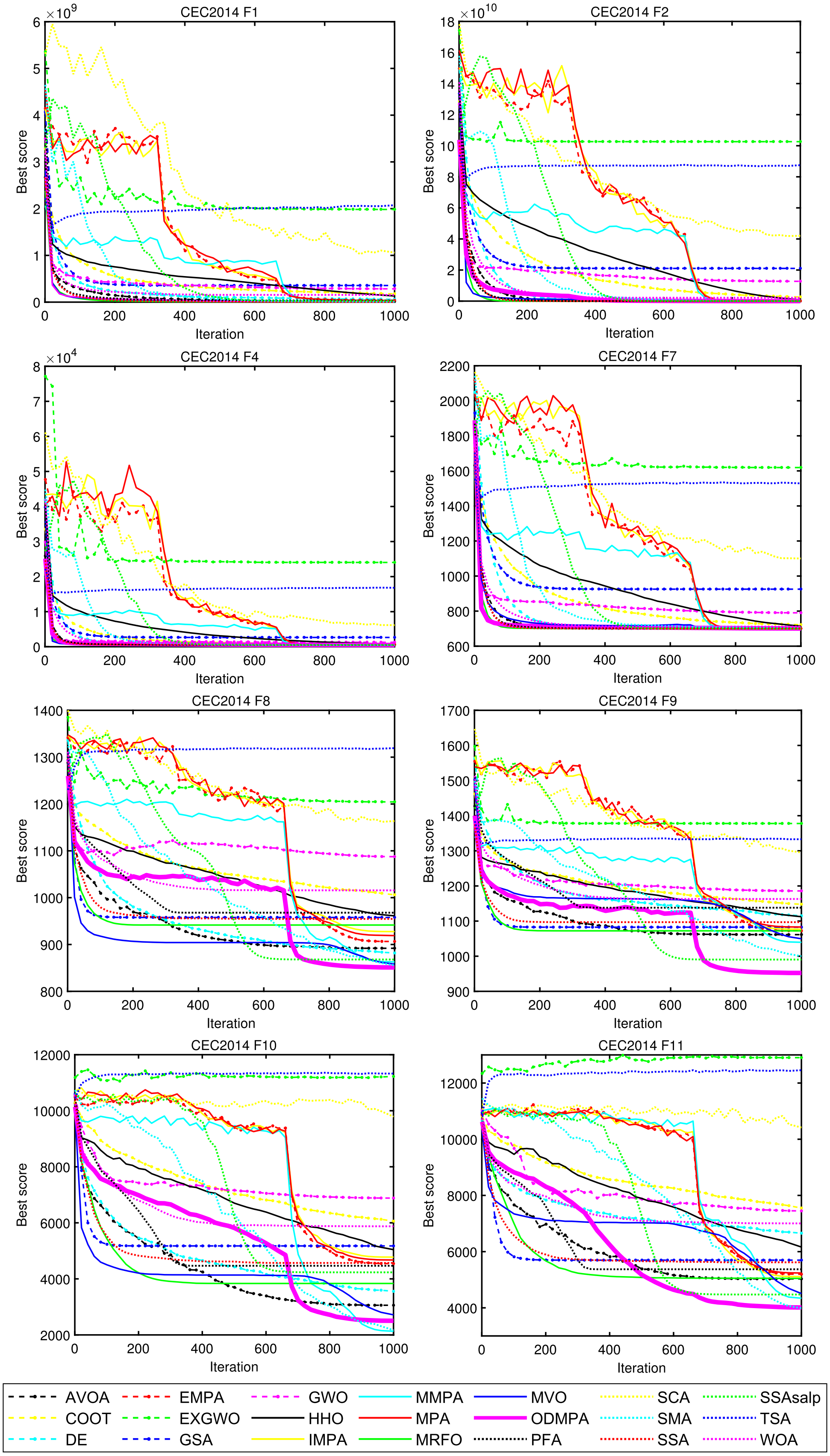
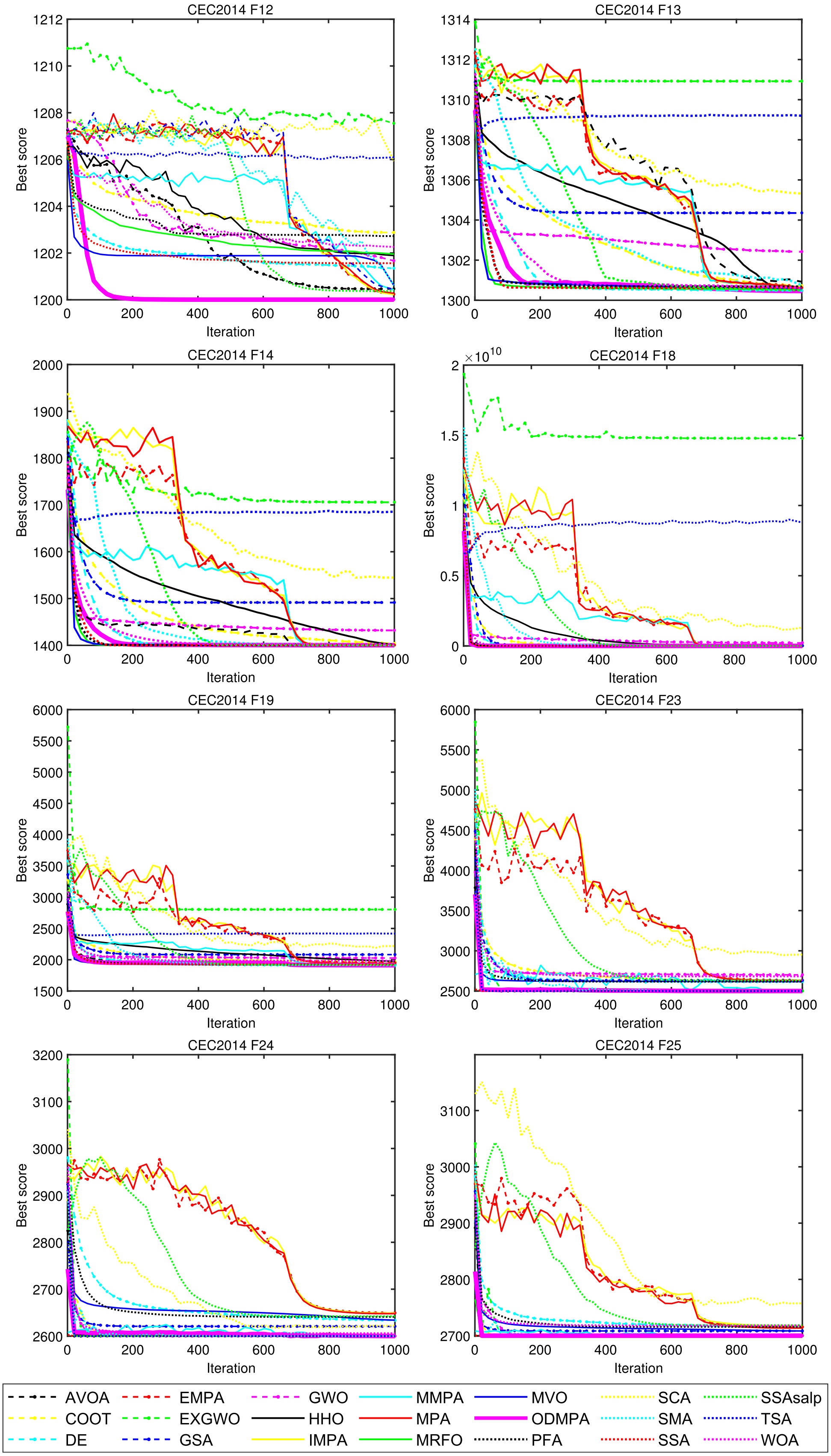


Fig. 5 The iteration curve of ODMPA on CEC2014 benchmark functions



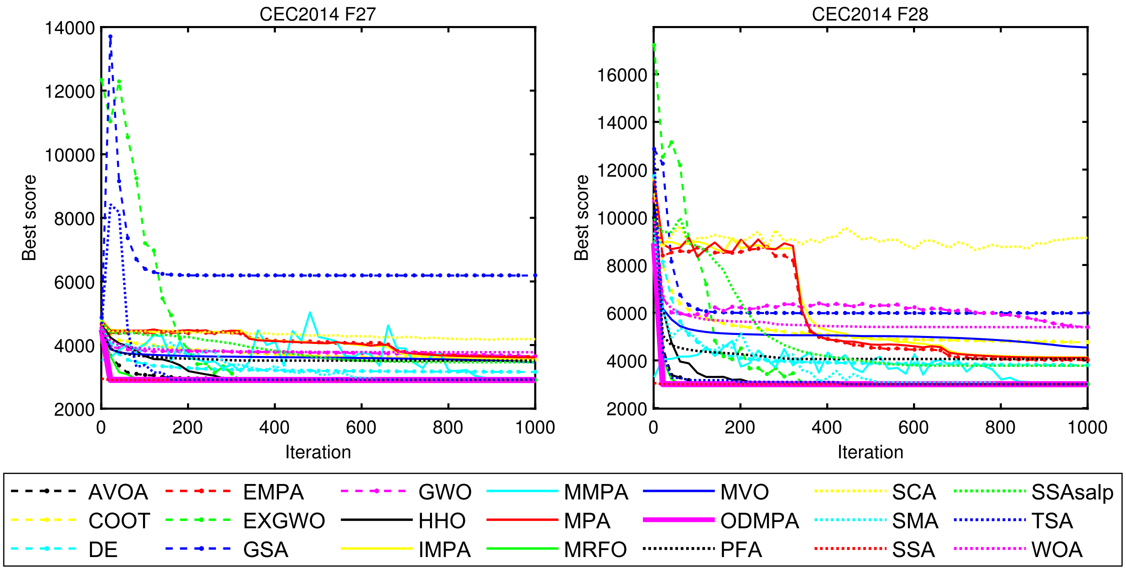


Fig. 6 The iteration curve of ODMPA on CEC2014 benchmark functions

Table 5 WBD problem's results on different algorithms

	Optimal solution variables				Optimal cost
	h	*l*	*t*	*b*	
ODMPA	**0.24011**	**2.8581**	**9.0366**	**0.20573**	**1.7017**
BA	2	0.1	3.174303	2	1.818138
RO	0.203687	3.528467	9.004233	0.207241	1.735344
GSA	0.20573	3.47049	9.03662	0.20573	1.7248
IHS	0.20573	3.47049	9.03662	0.20573	1.7248
hHHO-SCA	0.190086	3.696496	9.386343	0.204157	1.779032249
MTSA	0.24415742	6.22306595	8.29555011	0.24440474	2.38241101
CPSO	0.202369	3.544214	9.048210	0.205723	1.728024

Bold results are the minimum fitness values calculated by ODMPA

$$\begin{aligned}
&\overrightarrow{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L],\\
&\text{Minimize}: f(\overrightarrow{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2\\
&\quad + 3.1661x_1^2x_4 + 19.84x_1^2x_3,\\
&\text{Subjectto}:\\
&g_1(\overrightarrow{x}) = -x_1 + 0.0193x_3 \leq 0,\\
&g_2(\overrightarrow{x}) = -x_2 + 0.00954x_3 \leq 0,\\
&g_3(\overrightarrow{x}) = -\pi x_4 x_3^2 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0,\\
&g_4(\overrightarrow{x}) = x_4 - 240 \leq 0,\\
&\text{where}\\
&0.0625 \leq x_1 \leq 6.1875,\ 0.0625 \leq x_2\\
&\quad \leq 6.1875,\ 10 \leq x_3 \leq 200,\ 10 \leq x_4 \leq 200. \qquad (19)
\end{aligned}$$

The comparison results of ODMPA and other famous traditional and advanced algorithms for PVD problem, including BA, IHS, PSO, GA, ES [64], SGOA [2], and MMPA [55], are shown in Table 6. The results demonstrate that ODMPA performs more excellent than other algorithms, and the best cost is 5835.5822.

4.2.3 Speed Reducer Design Problem

The model of this engineer problem is aiming at minimizing the cost of the speed reducer. The SRD problem has seven variables: teeth number of the pinion (z), width of reducer (b), first shaft's length (l_1), diameter of shaft 1 (d_1), module of teeth (m), second shaft's length (l_2), diameter of shaft 2 (d_2). The eleven constraints and the mathematical formula of SRD are shown in Eq. 20.

$$\begin{aligned}
&\overrightarrow{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]\\
&\text{Minimize}: f(\overrightarrow{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3\\
&\quad - 43.0934) - 1.508x_1(x_6^2 + x_7^2)\\
&\quad + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2),
\end{aligned}$$

Table 6 PVD problem's results on different algorithms

	Optimal solution variables				Optimal cost
	T_s	T_h	R	L	
ODMPA	**0.51375**	**0.25013**	**45.2658**	**200**	**5835.5822**
BA	97.8015	98.10897	10.98606	200	7258.5640
IHS	1.125000	0.625000	58.29015	43.69268	7197.7300
PSO	0.812500	0.437500	42.091266	176.746500	6061.0777
GA	0.937500	0.500000	48.329000	112.679000	6410.3811
ES	0.812500	0.437500	42.098087	176.640518	6059.7456
SGOA	0.816637	0.403014	42.209181	175.779778	5977.4951
MMPA	0.778168	0.38464899	40.319629	199.999897	5885.3326

Bold results are the minimum fitness values calculated by ODMPA

$$\begin{aligned}
&\text{Subjectto}:\\
&g_1(\overrightarrow{x}) = \frac{27}{x_1x_2^2x_3} - 1 \le 0,\\
&g_2(\overrightarrow{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0,\\
&g_3(\overrightarrow{x}) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0,\\
&g_4(\overrightarrow{x}) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0,\\
&g_5(\overrightarrow{x}) = \frac{\left[\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9\times10^6\right]^{1/2}}{110x_6^3} - 1 \le 0,\\
&g_6(\overrightarrow{x}) = \frac{\left[\left(\frac{745x_4}{x_2x_3}\right)^2 + 157.5\times10^6\right]^{1/2}}{85x_7^3} - 1 \le 0,\\
&g_7(\overrightarrow{x}) = \frac{x_2x_3}{40} - 1 \le 0,\\
&g_8(\overrightarrow{x}) = \frac{5x_2}{x_1} - 1 \le 0,\\
&g_9(\overrightarrow{x}) = \frac{x_1}{12x_2} - 1 \le 0,\\
&g_{10}(\overrightarrow{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0,\\
&g_{11}(\overrightarrow{x}) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0,\\
&\text{where}\\
&2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3\\
&\quad \le 28, 7.3 \le x_4 \le 8.3,\\
&7.3 \le x_5 \le 8.3, 2.9 \le x_6 \le 3.9, 5.0 \le x_7 \le 5.5. \qquad (20)
\end{aligned}$$

In this problem, ODMPA is compared with other algorithms, including WCA [65], HEAA [66], MDE [67], m--HHO [68], EHHO, GLF-GWO [69], and m-SSA [70]. The results displayed in Table 7 demonstrate that ODMPA performs more excellent than other algorithms, and the optimal cost is 2753.9866.

4.3 Static and Dynamic Photovoltaic Models

The performance of the whole PV system now depends on its effective modeling, and the swarm intelligence algorithm can effectively solve the problem of PV identification parameters. Compared with other algorithms, ODMPA performs better in solving the problem of PV system identification parameters and also shows its superior performance. The model of this problem is nonlinearly optimized, and the main target is to extract the parameters of photovoltaic models. The static and dynamic models are chosen in this paper. An objective function was designed to set up accurate photovoltaic models for efficient reproduction. The minimization objective function involves the parameters needed to distinguish. Considering the sensitivity of the model parameters, the root-mean-square error is introduced into these experiments as the final evaluation standard, and the value of RMSE is calculated among the measured and estimated current.

4.3.1 Single-Diode Model

Modeling of photovoltaic modules is usually based on an equivalent circuit model with total parameters. The theory of SDM is established in the Shockley equation [71]. The formula of the output current is given by Eqs. (21–22):

$$\begin{aligned}
&I_L = I_{ph} - I_d - I_{sh}\\
&I_d = I_{sd}\cdot[\exp\left(\frac{V_L + R_S\cdot I_L}{n\cdot V_t}\right) - 1]\\
&I_{sh} = \frac{V_L + R_S\cdot I_L}{R_{sh}}\\
&where\\
&V_t{=}\frac{kT}{q}, k{=}1.3806503{\times}10^{-23},
\end{aligned}$$

Table 7 SRD problem's results on different algorithms

	Optimal solution variables							Optimal cost
	b	m	z	l_1	l_2	d_1	d_2	
ODMPA	**3.4764**	**0.7000**	**17.0000**	**8.2922**	**8.2999**	**3.3509**	**5.2781**	**2753.9866**
WCA	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2866	2994.4711
HEAA	3.5000	0.7000	17.0000	7.3004	7.7153	3.3502	5.2867	2994.4991
MDE	3.5000	0.7000	17.0000	7.3002	7.8000	3.3502	5.28687	2996.3567
m-HHO	3.5000	0.7000	17.0000	7.3000	7.8000	3.3512	5.2867	2996.6162
EHHO	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2866	2994.4710
GLF-GWO	3.5000	0.7000	17.0000	7.3000	7.8000	3.3502	5.2867	2996.3680
m-SSA	3.5000	0.7000	17.0000	7.3470	7.8032	3.3517	5.2867	2997.2376

Bold results are the minimum fitness values calculated by ODMPA

$$q{=}1.60217646{\times}10^{-19}. \tag{21}$$

According to the above equations, the output current expression can be derived:

$$\begin{aligned}&\overrightarrow{x} = [x1, x2, x3, x4, x5] = [I_{ph}, I_{sd}, R_S, R_{sh}, n],\\ &I_L = I_{ph} - I_{sd}\cdot\left[\exp\left(\tfrac{V_L+R_S\cdot I_L}{n\cdot V_t}\right)-1\right]-\tfrac{V_L+R_S\cdot I_L}{R_{sh}}.\end{aligned} \tag{22}$$

4.3.2 Double-Diode Model

For the SDM in harsh external environments, producing the compound current loss is easy. DDM considers this situation and makes up for it. The formula of the DDM's output current is given by Eq. (23):

$$\begin{aligned}\overrightarrow{x} &= [x_1, x_2, x_3, x_4, x_5, x_6, x_7]\\ &= [I_{ph}, I_{sd_1}, I_{sd_2}, R_S, R_{sh}, n_1, n_2],\\ I_L &= I_{ph} - I_{sd_1}\cdot\left[\exp\left(\frac{V_L + I_L\cdot R_S}{n_1\cdot V_t}\right)-1\right]\\ &\quad - I_{sd_2}\cdot\left[\exp\left(\frac{V_L + I_L\cdot R_S}{n_2\cdot V_t}\right)-1\right]\\ &\quad -\frac{V_L + I_L\cdot R_S}{R_{sh}},\\ &\quad where\\ V_t &= \frac{kT}{q}, k = 1.3806503\times10^{-23},\\ q &= 1.60217646\times10^{-19}.\end{aligned} \tag{23}$$

4.3.3 PV Module Model

In practical cases, the PV module is made up of some sunshine-based cells in series and parallel. The current formula of SDM for the PV module is expressed in Eq. (24).

$$\begin{aligned}\overrightarrow{x} &= [x_1, x_2, x_3, x_4, x_5, x_6, x_7]\\ &= [N_p, N_s, I_{ph}, I_{sd}, R_S, R_{sh}, n],\\ I_L &= I_{ph}N_p - I_{sd}N_p\left[\exp\left(\frac{V_L + (N_sR_SI_L/N_p)}{nN_sV_t}\right)-1\right]\\ &\quad -\frac{V_L + (N_sR_SI_L/N_p)}{(N_sR_{sh}/N_p)},\\ &\text{where}\\ V_t &= \frac{kT}{q}, k = 1.3806503\times10^{-23},\\ q &= 1.60217646\times10^{-19}.\end{aligned} \tag{24}$$

where N_p and N_s denote the number of sun-oriented cells in parallel and arrangement, respectively. The parameters setting is the same as the SDM model.

4.3.4 Objective Function

The target of algorithms to optimize the photovoltaic models is minimizing the gap of the estimated and measured current. In order to achieve the goal of validating the performance of photovoltaic models, the choice of evaluation standard is crucial. Root-mean-square error (RMSE) is appropriate, and the original problem is turned into the minimization of RMSE. The mathematical formula is given by Eq. (25):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(I_{simulated} - I_{measured})^2}{N}} \tag{25}$$

where the $I_{measured}$ represents the actual measured current, $I_{simulated}$ is the current calculated by formulas (21)–(24), and N indicates the number of groups of the current.

Two literature datasets [72] are used to extract the parameters of PV models, and the detailed information of the datasets is listed as follows. The RTC France cell consists of twenty-six pairs of data of the current and voltage, and its irradiance and temperature are 1000 W/m2 and 33°C, respectively, while the Photowatt-PWP201 consists of twenty-five pairs of data with thirty-six polysilicon cells, and the temperature is 45°C.

Table 8 The value range of photovoltaic models

	SDM/DDM		PV module	
Parameters	Lower bound	Upper bound	Lower bound	Upper bound
$I_{ph}(A)$	0	1	0	2
$I_{sd}(\mu A)$	0	1	0	50
$R_s(\Omega)$	0	0.5	0	2
$R_{sh}(\Omega)$	0	100	0	2000
n	1	2	1	50
$I_{sd1}(\mu A)$	0	1	–	–
$I_{sd1}(\mu A)$	0	1	–	–
n_1	1	2	–	–
n_2	1	2	–	–

Table 9 Comparison results for SDM problem (data: RTC France)

Algorithm	$I_{ph}(A)$	$I_{sd}(\mu A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n	RMSE
ODMPA	**0.76078**	**0.32242**	**0.036384**	**53.6349**	**1.481**	**9.86E-04**
SFLA [73]	0.7602	0.3212	0.0374	75.362	1.4802	1.8791E-03
IJAYA [74]	0.7607	0.7607	0.0363	55.429	1.4832	9.8871E-04
GOTLBO [75]	0.7608	0.3102	0.0365	51.971	1.4771	9.8997E-04
WOA	0.76034	0.62123	0.032216	61.7541	1.5505	2.71E-03
MPA	0.76057	0.46231	0.034933	66.9487	1.5182	1.20E-03

Bold results are the minimum fitness values calculated by ODMPA

Table 10 Comparison results for DDM problem (data: RTC France)

Algorithm	$I_{ph}(A)$	$I_{sd1}(\mu A)$	$I_{sd2}(\mu A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n_1	n_2	RMSE
ODMPA	**0.76078**	**0.31048**	**0.25477**	**0.036563**	**54.5384**	**1.8749**	**1.462**	**9.84E-04**
MPA	0.76012	0.9656	0.34761	0.034917	89.0604	2	1.4953	1.39E-03
AVOA	0.76103	0.12458	0.18621	0.037681	46.0647	1.4128	1.5879	1.14E-03
PSO	0.75777	0.16188	0.34265	0.03461	44.0048	1.7683	1.4929	3.63E-03
COOT	0.76708	0.11858	0.37383	0.03402	22.3821	1.7436	1.5018	4.62E-03
MVO	0.7601	0.09545	0.83353	0.034821	95.9077	1.4177	1.6861	1.50E-03

Bold results are the minimum fitness values calculated by ODMPA

ODMPA and other counterparts are run on MATLAB R2016a, and each algorithm is executed thirty times in case of contingency. The search agents size and dimension are 30, and the maximum number of evaluations is 300000. PV models' parameter setting is displayed in Table 8.

4.3.5 Identify SDM and DDM on the RTC France

Tables 9 and 10 record the RMSE values obtained by running SDM and DDM models with various algorithms and record the parameter groups corresponding to the optimal values. It can be seen that ODMPA shows optimal results, which indicates that ODMPA is more reliable and effective.

4.3.6 Identify the Photovoltaic Cell Module on the Photo-Watt-PWP201

Table 11 records the RMSE values obtained by running SDM and DDM models with various algorithms and the parameter groups corresponding to the optimal values. The results show that ODMPA obtains the lowest RMSE values.

These practical experiments prove that when extracting different models' unknown parameters, ODMPA can always have a solid ability to get optimal results.

Table 11 Comparison results for PV module problem (data: Photo-watt-PWP201)

Algorithm	$I_{ph}(A)$	$I_{sd}(\mu A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n	RMSE
ODMPA	**1.0304**	**3.521**	**1.2002**	**995.0763**	**48.6851**	**2.43E-03**
DE	1.0297	3.9448	1.1882	1141.3032	49.125	2.45E-03
EMPA	1.0288	4.7905	1.1677	1513.2231	49.8943	2.58E-03
GA	1.0303	4.0485	1.1843	1075.9085	49.2293	2.47E-03
HHO	1.0274	4.7219	1.1704	1993.3263	49.8301	2.62E-03
WOA	1.0274	4.9109	1.1648	1999.6029	49.9901	2.64E-03

Bold results are the minimum fitness values calculated by ODMPA

5 Conclusion and Future Direction

This study proposes a new MPA variant: ODMPA. The improved algorithm has stronger search capability and can jump out quickly when it is trapped in a local optimum. In order to retain the advantages of the original MPA and try to overcome the shortcomings, the main framework of MPA is reserved. Firstly, the outpost mechanism is introduced after the three-stage velocity ratio, which improves the exploitation ability in the neighborhood of feasible solutions and enhances the convergence accuracy. Then the DE-SA mechanism is introduced into MPA after the FADs stage to generate a new solution. The best candidate is selected among the traditional MPA variant and the DE-SA variant. The DE-SA strategy helps to improve the variety of solutions, which increases the spaces that can be explored and avoid premature phenomenon. The improved ODMPA is employed to tackle CEC2014 benchmark problems, engineering design problems, and photovoltaic models. Based on the Friedman assessment, the ODMPA ranks first in the 30 CEC2014 functions test set. Combining the excellent convergence speed and accuracy of ODMPA's convergence curve, the empirical results have demonstrated that ODMPA outperforms other algorithms in terms of search capability and convergence speed. Moreover, in complex real-life problems like engineering design problems and parameter extraction of PV cells, the application of ODMPA also shows it can perform outstandingly to find optimal results.

Accordingly, the ODMPA is an optimizer with great potential, ODMPA can be applied in information fusion [76], recommender systems [77], and machine learning. Moreover, ODMPA has the hope to be improved to tackle discrete or multi-objective optimization problems. Based on the excellent results of combining MPA with DE, combining other optimizers with MPA is also a valuable research direction in the future.

Acknowledgements This work is supported by the Innovation Capacity Construction Project of Jilin Province Development and Reform Commission (2021FGWCXNLJSSZ10, 2019C053-3), the National Key Research and Development Program of China (No. 2020YFA0714103), and the Fundamental Research Funds for the Central Universities, JLU.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Mohapatra, P.; et al.: A modified competitive swarm optimizer for large scale optimization problems. Appl. Soft Comput. **59**, 340–362 (2017). https://doi.org/10.1016/j.asoc.2017.05.060
2. Yu, C.; et al.: SGOA: annealing-behaved grasshopper optimizer for global tasks. Eng. Comput. **2**, 1–28 (2021). https://doi.org/10.1007/s00366-020-01234-1
3. Zhao, S.; et al.: An enhanced cauchy mutation grasshopper optimization with trigonometric substitution: engineering design and feature selection. Eng. Comput. (2021). https://doi.org/10.1007/s00366-021-01448-x
4. Krishna, A.B.; et al.: hSMA-PS: a novel memetic approach for numerical and engineering design challenges. Eng. Comput. (2021). https://doi.org/10.1007/s00366-021-01371-1
5. Abbasi, A.; et al.: Multi-strategy Gaussian Harris hawks optimization for fatigue life of tapered roller bearings. Eng. Comput. (2021). https://doi.org/10.1007/s00366-021-01442-3
6. Nautiyal, B.; et al.: Improved Salp Swarm Algorithm with mutation schemes for solving global optimization and engineering problems. Eng. Comput. (2021). https://doi.org/10.1007/s00366-020-01252-z
7. Dong, R.Y.; et al.: Boosted kernel search: framework, analysis and case studies on the economic emission dispatch problem. Knowl.-Based Syst. **233**, 107529 (2021). https://doi.org/10.1016/j.knosys.2021.107529
8. Dorigo, M.; et al.: Ant colony optimization theory: a survey. Theor. Comput. Sci. **344**(2–3), 243–278 (2005). https://doi.org/10.1016/j.tcs.2005.05.020
9. Eberhart, R.; et al.: A new optimizer using particle swarm theory. In: Mhs95 Sixth International Symposium on Micro Machine & Human Science. IEEE (2002)
10. Sm, A.; et al.: Grey Wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
11. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl.-Based Syst. **89**, 228–249 (2015). https://doi.org/10.1016/j.knosys.2015.07.006
12. Mirjalili, S.; et al.: Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput. Appl. **27**(2), 495–513 (2016). https://doi.org/10.1007/s00521-015-1870-7

13. Mirjalili, S.; et al.: Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. **114**, 163–191 (2017). https://doi.org/10.1016/j.advengsoft.2017.07.002
14. Wang, D.; et al.: Heterogeneous differential evolution algorithm for parameter estimation of solar photovoltaic models. Energy Rep. **8**, 4724–4746 (2022). https://doi.org/10.1016/j.egyr.2022.03.144
15. Wang, Y.; et al.: The utilization of adaptive African vulture optimizer for optimal parameter identification of SOFC. Energy Rep. **8**, 551–560 (2022). https://doi.org/10.1016/j.egyr.2021.11.257
16. Xu, S.; et al.: A modified stochastic fractal search algorithm for parameter estimation of solar cells and PV modules. Energy Rep. **8**, 1853–1866 (2022). https://doi.org/10.1016/j.egyr.2022.01.008
17. Dang, K.D.; et al.: A single step optimization method for topology, size and shape of trusses using hybrid differential evolution and symbiotic organisms search. Comput. Struct. **270**, 106846 (2022). https://doi.org/10.1016/j.compstruc.2022.106846
18. Zhang, Y.N.; et al.: Boosted binary Harris hawks optimizer and feature selection. Eng. Comput. **37**(4), 3741–3770 (2021). https://doi.org/10.1007/s00366-020-01028-5
19. Zhang, Y.N.; et al.: Towards augmented kernel extreme learning models for bankruptcy prediction: algorithmic behavior and comprehensive analysis. Neurocomputing **430**, 185–212 (2021). https://doi.org/10.1016/j.neucom.2020.10.038
20. Zeng, G.Q.; et al.: Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems. Swarm Evol. Comput. **44**, 320–334 (2019). https://doi.org/10.1016/j.swevo.2018.04.008
21. Zhao, D.; et al.: Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy. Knowl.-Based Syst. **216**, 106510 (2021). https://doi.org/10.1016/j.knosys.2020.106510
22. Pang, J.H.; et al.: A scatter simulated annealing algorithm for the bi-objective scheduling problem for the wet station of semiconductor manufacturing. Comput. Ind. Eng. **123**, 54–66 (2018). https://doi.org/10.1016/j.cie.2018.06.017
23. Zeng, G.Q.; et al.: Modified extremal optimization for the hard maximum satisfiability problem. J. Zhejiang Univ.-Sci. C-Comput. Electron. **12**(7), 589–596 (2011). https://doi.org/10.1631/jzus.C1000313
24. Zhao, X.H.; et al.: Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients. Comput. Biol. Chem. **78**, 481–490 (2019). https://doi.org/10.1016/j.compbiolchem.2018.11.017
25. Faramarzi, A.; et al.: Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst. Appl. **152**, 113377 (2020)
26. Dinh, P.H.; et al.: An improved medical image synthesis approach based on marine predators algorithm and maximum Gabor energy. Neural Comput. Appl. **34**(6), 4367–4385 (2022). https://doi.org/10.1007/s00521-021-06577-4
27. Islam, M.Z.; et al.: Marine predators algorithm for solving single-objective optimal power flow. PLoS ONE **16**(8), e0256050 (2021). https://doi.org/10.1371/journal.pone.0256050
28. Yousri, D.; et al.: Fractional-order comprehensive learning marine predators algorithm for global optimization and feature selection. Knowl.-Based Syst. **235**, 107603 (2022)
29. Elaziz, M.A.; et al.: Enhanced marine predators algorithm for identifying static and dynamic photovoltaic models parameters. Energy Convers. Manag. **236**, 113971 (2021)
30. Yousri, D.; et al.: A new comprehensive learning marine predator algorithm for extracting the optimal parameters of supercapacitor model. J. Energy Storage **42**, 103035 (2021)
31. Sahlol, A.T.; et al.: Covid-19 image classification using deep features and fractional-order marine predators algorithm. Sci. Rep. **10**(1), 15364 (2020)
32. Yousri, D.; et al.: Parameters identification of solid oxide fuel cell for static and dynamic simulation using comprehensive learning dynamic multi-swarm marine predators algorithm. Energy Convers. Manag. **228**, 113692 (2021)
33. Abd Elaziz, M.; et al.: Quantum marine predators algorithm for addressing multilevel image segmentation. Appl. Soft Comput. **110**, 107598 (2021)
34. Eid, A.; et al.: Marine predators algorithm for optimal allocation of active and reactive power resources in distribution networks. Neural Comput. Appl. **33**(21), 14327–14355 (2021). https://doi.org/10.1007/s00521-021-06078-4
35. Yang, W.B.; et al.: A multi-strategy marine predator algorithm and its application in joint regularization semi-supervised ELM. Mathematics **9**(3), 291 (2021)
36. Chen, H.L.; et al.: Efficient multi-population outpost fruit fly-driven optimizers: framework and advances in support vector machines. Expert Syst. Appl. **142**, 112999 (2020)
37. Storn, R.; et al.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
38. Kirpatrick, S.; et al.: Optimization by simulated annealing. Read. Comput. Vis. **220**, 606–615 (1983)
39. Metropolis, A.; et al.: Teller, and E. Teller. J. Chem. Phys. **95**, 2048–2055 (1953)
40. He, Y.; et al.: A precise chaotic particle swarm optimization algorithm based on improved tent map. Fourth International Conference on Natural Computation, 2008. ICNC '08. (2008)
41. Erlich, I.; et al.: Evaluating the mean-variance mapping optimization on the IEEE-CEC 2014 test suite. In: 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE. (2014)
42. Joaquín, D.A.; et al.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evolut. Comput. **1**(1), 3–18 (2011)
43. Rashedi, E.; et al.: GSA: a gravitational search algorithm. Inf. Sci. **179**, 2232–2248 (2009)
44. Zhao, W.; et al.: Manta ray foraging optimization: an effective bio-inspired optimizer for engineering applications. Eng. Appl. Artif. Intell. **87**(1), 103300.1-103300.25 (2020)
45. Aaha, B.; et al.: Harris hawks optimization: algorithm and applications. Futur. Gener. Comput. Syst. **97**, 849–872 (2019)
46. Yapici, H.; et al.: A new meta-heuristic optimizer: pathfinder algorithm. Appl. Soft Comput. **78**, 545–568 (2019)
47. Mirjalili, S.; et al.: SCA: a sine cosine algorithm for solving optimization. Probl. Knowl.-Based Syst. **96**, 120–133 (2016)
48. Li, S.; et al.: Slime mould algorithm: a new method for stochastic optimization. Future Gener. Comput. Syst. **111**, 300–323 (2020)
49. Xue, J.; et al.: A novel swarm intelligence optimization approach: sparrow search algorithm. Syst. Sci. Control Eng. **8**(1), 22–34 (2020). https://doi.org/10.1080/21642583.2019.1708830
50. Kaur, S.; et al.: Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Eng. Appl. Artif. Intell. **90**, 103541 (2020)
51. Mirjalili: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016)
52. Naruei, I.; et al.: A new optimization method based on coot bird natural life model. Expert Syst. Appl. **183**(2), 115352 (2021)
53. Abdollahzadeh, B.; et al.: African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput. Ind. Eng. **158**, 107408 (2021)
54. Seyyedabbasi, A.; et al.: I-GWO and Ex-GWO: improved algorithms of the grey Wolf optimizer to solve global optimization problems. Eng. Comput. **37**, 509–532 (2019)
55. Fan, Q.; et al.: A modified self-adaptive marine predators algorithm: framework and engineering applications. Eng. Comput. (2021). https://doi.org/10.1007/s00366-021-01319-5

56. Abdel-Basset, M.; et al.: A hybrid COVID-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy. IEEE Access **PP**(99), 1–1 (2020)
57. Coello, C.A.C.: Use of a self-adaptive penalty approach for engineering optimization problems. Comput. Ind. **41**(2), 113–127 (2000). https://doi.org/10.1016/s0166-3615(99)00046-9
58. Yang, X.S.: A new metaheuristic bat-inspired algorithm. Studies in Computational Intelligence [Nicso 2010: nature inspired cooperative strategies for optimization]. International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO 2008), Tenerife, SPAIN (2008)
59. Kaveh, A.; et al.: A new meta-heuristic method: ray Optimization. Comput. Struct **112**, 283–294 (2012). https://doi.org/10.1016/j.compstruc.2012.09.003
60. Mahdavi, M.; et al.: An improved harmony search algorithm for solving optimization problems. Appl. Math. Comput. **188**(2), 1567–1579 (2007). https://doi.org/10.1016/j.amc.2006.11.033
61. Kamboj, V.K.; et al.: An intensify Harris Hawks optimizer for numerical and engineering optimization problems. Appl. Soft Comput. **89**, 106018 (2020). https://doi.org/10.1016/j.asoc.2019.106018
62. Babalik, A.; et al.: A modification of tree-seed algorithm using Deb's rules for constrained optimization. Appl. Soft Comput. **63**, 289–305 (2018). https://doi.org/10.1016/j.asoc.2017.10.013
63. He, Q.; et al.: An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng. Appl. Artif. Intell. **20**(1), 89–99 (2007). https://doi.org/10.1016/j.engappai.2006.03.003
64. Mezura-Montes, E.; et al.: An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int. J. Gen. Syst. **37**(4), 443–473 (2008). https://doi.org/10.1080/03081070701303470
65. Eskandar, H.; et al.: Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput. Struct. **110**, 151–166 (2012). https://doi.org/10.1016/j.compstruc.2012.07.010
66. Wang, Y.; et al.: Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct. Multidiscip. Optim. **37**(4), 395–413 (2009). https://doi.org/10.1007/s00158-008-0238-3
67. Mezura-Montes, E.; Velázquez-Reyes, J.; Coello Coello, C.A.: Modifed diferential evolution for constrained optimization. In: 2006 IEEE Congress on Evolutionary Computation. IEEE (2006)
68. Gupta, S.; et al.: Opposition-based learning Harris hawks optimization with advanced transition rules: principles and analysis. Expert Syst. Appl. **158**, 113510 (2020). https://doi.org/10.1016/j.eswa.2020.113510
69. Gupta, S.; et al.: Enhanced leadership-inspired grey wolf optimizer for global optimization problems. Eng. Comput. **36**(4), 1777–1800 (2020). https://doi.org/10.1007/s00366-019-00795-0
70. Gupta, S.; et al.: Harmonized salp chain-built optimization. Eng. Comput. **37**(2), 1049–1079 (2021). https://doi.org/10.1007/s00366-019-00871-5
71. Shockley, W.; et al.: Electrons and holes in semiconductors. Phys. Today **5**(12), 18 (1952)
72. Easwarakhanthan, T.; et al.: Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers. Int. J. Solar Energy **4**(1), 1–12 (1986)
73. Eusuff, M.; et al.: Shuffled frog-leaping algorithm: a memetic metaheuristic for discrete optimization. Eng. Optim. **38**(2), 129–154 (2006). https://doi.org/10.1080/03052150500384759
74. Yu, K.J.; et al.: Parameters identification of photovoltaic models using an improved JAYA optimization algorithm. Energy Convers. Manag. **150**, 742–753 (2017). https://doi.org/10.1016/j.enconman.2017.08.063
75. Chen, X.; et al.: Parameters identification of solar cell models using generalized oppositional teaching learning based optimization. Energy **99**, 170–180 (2016). https://doi.org/10.1016/j.energy.2016.01.052
76. Chen, Z.G.; et al.: Information synergy entropy based multi-feature information fusion for the operating condition identification in aluminium electrolysis. Inf. Sci. **548**, 275–294 (2021). https://doi.org/10.1016/j.ins.2020.07.031
77. Wang, D.H.; et al.: A content-based recommender system for computer science publications. Knowl.-Based Syst. **157**, 1–9 (2018). https://doi.org/10.1016/j.knosys.2018.05.001
78. Chen, H.; et al.: A balanced whale optimization algorithm for constrained engineering design problems. Appl. Math. Modell. **71**(JUL.), 45–59 (2019)
79. Zhang; et al.: Chaos-induced and mutation-driven schemes boosting salp chains-inspired optimizers. IEEE Access **7**, 31243–31261 (2019)
80. Gandomi, A.H.; Yang, X.S.: Benchmark problems in structural optimization. In: Koziel, S., Yang, X.S. (eds.) Chapter 12 in Computational Optimization, Methods and Algorithms, pp. 267–291. Springer, Berlin (2011)
81. Sun, X.; et al.: Optimal performance of a combined heat-power system with a proton exchange membrane fuel cell using a developed marine predators algorithm. J. Clean. Prod. **284**, 124776 (2020)
82. Zhao, S.; et al.: Elite dominance scheme ingrained adaptive salp swarm algorithm: a comprehensive study. Eng. Comput. **38**, 4501–4528 (2022). https://doi.org/10.1007/s00366-021-01464-x
83. Liang, X.; et al.: Chaotic oppositional sine-cosine method for solving global optimization problems. Eng. Comput. **38**(2), 1223–1239 (2022). https://doi.org/10.1007/s00366-020-01083-y
84. Wang, J.; et al.: Hybrid seagull optimization algorithm and its engineering application integrating yin-yang pair idea. Eng. Comput. **38**(3), 2821–2857 (2022)
85. Zhang, H.; et al.: Differential evolution assisted salp swarm algorithm with chaotic structure for real-world problems. Eng. Comput. (2022). https://doi.org/10.1007/s00366-021-01545-x
86. Hao, C.; et al.: Advanced orthogonal learning-driven multi-swarm sine cosine optimization: framework and case studies–sciencedirect. Expert Syst. Appl. **144**, 113113 (2019)