Mathematical Biosciences and Engineering

MBE, 20(1): 93–127.
DOI: 10.3934/mbe.2023006
Received: 18 June 2022
Revised: 24 August 2022
Accepted: 01 September 2022
Published: 29 September 2022

http://www.aimspress.com/journal/MBE

Research article

Modified Marine Predators Algorithm hybridized with teaching-learning mechanism for solving optimization problems

Yunpeng Ma[1], Chang Chang[2], Zehua Lin[2], Xinxin Zhang[1], Jiancai Song[1] and Lei Chen[1,*]

[1] School of Information Engineering, Tianjin University of Commerce, Beichen, Tianjin 300134
[2] College of Science, Tianjin University of Commerce, Beichen, Tianjin 300134

* **Correspondence:** Email: chenlei@tjcu.edu.cn; Tel: +8613512999888.

Abstract: Marine Predators Algorithm (MPA) is a newly nature-inspired meta-heuristic algorithm, which is proposed based on the Lévy flight and Brownian motion of ocean predators. Since the MPA was proposed, it has been successfully applied in many fields. However, it includes several shortcomings, such as falling into local optimum easily and precocious convergence. To balance the exploitation and exploration ability of MPA, a modified marine predators algorithm hybridized with teaching-learning mechanism is proposed in this paper, namely MTLMPA. Compared with MPA, the proposed MTLMPA has two highlights. Firstly, a kind of teaching mechanism is introduced in the first phase of MPA to improve the global searching ability. Secondly, a novel learning mechanism is introduced in the third phase of MPA to enhance the chance encounter rate between predator and prey and to avoid premature convergence. MTLMPA is verified by 23 benchmark numerical testing functions and 29 CEC-2017 testing functions. Experimental results reveal that the MTLMPA is more competitive compared with several state-of-the-art heuristic optimization algorithms.

Keywords: meta-heuristics optimization; Marine Predators Algorithm; exploitation and exploration; modified Marine Predators Algorithm; teaching-learning-based optimization

1. Introduction

Lots of real-life engineering optimization problems have complex characteristics, such as multi-modality, high-dimensional and non-differentiable, so that they are not easy to solve by traditional optimization methods. For instance, if we use traditional optimization methods (such as steepest decent,

dynamic programming and linear programming) to address an optimization problem, we need to calculate its gradient. So the traditional method cannot solve non-differentiable problems. Luckily, as the meta-heuristic optimization algorithms are proposed and developed, many complicated optimization problems can be solved easily and efficiently.

During the last three decades, the research of meta-heuristics intelligent optimization algorithm has become a research hot-spot, so that many state-of-the-art swarm optimization algorithms were proposed and developed. For example, inspired by the foraging behavior of the birds, particle swarm optimization (PSO) was proposed [1]. Krill herds (KH) algorithm [2] was proposed based on the foraging behavior of krill herds, which shows fast convergence speed but poor convergence accuracy. Based on cuckoo parasitic brood behavior, Cuckoo Search (CS) [3] was proposed in 2009, which has good versatility and global searching ability. Grey wolf optimization algorithm (GWO) [4] was proposed based on the foraging behavior of wolves, which shows good performance and has been applied in many fields. Inspired by the teaching learning phenomenon, a kind of teaching learning based optimization algorithm (TLBO) [5,38–42,45] was proposed, which has good global searching ability but poor local searching ability. The whale optimization algorithm (WOA) [6] was proposed based on the foraging behavior of whales, which is good at solving large spatial gradient problem, but it cannot jump out of the local optima. Marine Predator algorithm (MPA) [7] was proposed based on the foraging behavior of ocean predators. Moreover, many researchers have proved that these swarm-inspired optimization algorithms are suitable to solve complex function problems and difficult real-life optimization problems [8–14]. In deep learning, the optimizer was used to find the optimal solution of the model. The improvement of the optimizer was also applied in all areas of life [15–17,36–37].

In this paper, we focus on studying the Marine Predator Algorithm (MPA). The MPA is a novel simple and efficient meta-heuristic optimization algorithm inspired by the survival of the fittest theory in the ocean. MPA has many advantages, including fewer parameters, simple configureuration, ease of implementation and high calculation accuracy. Therefore, since the MPA was proposed, it has caused researchers' attentions and applied in many fields successfully. Chen et al. proposed a rolling bearing fault diagnosis method based on the MPA based-support vector machine [18]. The MPA was utilized to fuse base layers by optimal parameters, allowing the output image to have good quality [19]. The optimum design of the controller was established by using MPA [20]. However, the MPA still exists several drawbacks, such as falling into local optima easily, poor balance ability of exploitation and exploration, weak convergence speed and solution quality. In order to enhance the performance of MPA, researchers have proposed many variants of MPA. In [21], MPA was compared with high-performance optimizer and other classical algorithms that recently developed. Elaziz et al. [22] proposed an improved MPA based on quantum theory to handle multi-level image segmentation problems. Ramezani et al. [23] noted that MPA is deficient in terms of local optimization of fast escape and exploration of space and enhanced the algorithm by incorporating the characteristics of opposition learning, chaos graphs and so on. The enhanced MPA [24] implemented a population enhancement strategy to improve solution quality, which was applied to the parameter estimation of the photovoltaic model. In [25], a multi-objective MPA was proposed. Optimal vehicle-to-grid and grid-to-vehicle scheduling strategy [26] using improved marine predator algorithm. An improved MPA was presented [27] for the optimal design of hybrid renewable energy systems. An enhanced multi-objective optimization algorithm of the MPA was proposed for minimizing the operating cost and emission [28].

The MPA has four phases: MPA formulation, MPA optimization scenarios, eddy formation and

FAD's effect, marine memory. The most core phase of the MPA is 'MPA optimization scenarios'. The MPA algorithm has been applied into many fields because of its superior performance. However, the MPA still includes some disadvantages, such as falling into local optima easily, slow convergence rate and poor solution quality. Therefore, this paper introduces teaching-learning group mechanisms in the 'MPA optimization scenarios' phase to improve the convergence accuracy and solution quality. Moreover, there are three phases in 'MPA optimization scenarios'. In the first phase, a kind of teaching group mechanism is introduced, which was proposed in MTLBO [29] by our team. Actually, the population individuals will be divided into two groups based on the fitness values of function. And the two group individuals have different position updating mechanism. In the third phase, another kind of learning group mechanism is introduced to update those population individuals. The proposed MTLMPA is verified by 23 benchmark numerical testing functions and several CEC-2017 functions. Experimental results reveal that the MTLMPA presents better performance on most testing functions compared with state-of-the-art heuristic optimization algorithms.

The main contributions of this paper can be summarized as follows:

1) Based on conventional MPA, a modified marine predators algorithm hybridized with teaching-learning group mechanism is proposed.

2) 23 benchmark numerical functions and several CEC 2017 testing functions are used to evaluate the performance of MTLMPA. Compared with other state-of-the-art algorithms, MTLMPA can provide competitive solutions on most testing functions.

The rest of the paper is organized as follows. Section 2 presents preliminaries of marine predator algorithm in detail. Section 3 proposes the MTLMPA algorithm. The performance of the proposed method is tested and analyzed in Section 4. Finally, Section 5 concludes the work and outlines several advises for future work.

2. Marine Predators Algorithm

Inspired by the widespread foraging strategy of ocean predators, a nature-inspired meta-heuristic optimization algorithm, called Marine Predators Algorithm (MPA), was proposed in 2019. During foraging, the predators obey the Brownian motion and Lévy flight [30–35,43–44]. In MPA, the prey and predators update their position based on Brownian motion or Lévy flight. The MPA has four basic phases, which are described in detail as follows.

2.1. Initialization phase

In this subsection, population individuals are generated randomly by uniform distributed method, denoted $X_0 = X_{min} + \text{rand}(X_{max} - X_{min})$. X_{min} and X_{max} are the lower limit and upper limit of variables, respectively. rand is a random vector in the range 0–1. Then, the fitness function value of every individual is calculated. Finally, Elite matrix and Prey matrix are constructed. Based on the Elite matrix and Prey matrix, population individuals update their positions.

The Elite matrix is constructed from the optimal solution being specified as the top predator. The second matrix is defined as the Prey matrix. The predator updates its position according to the Prey matrix.

$$Elite = \begin{bmatrix} X_{1,1}^I & X_{1,2}^I & \cdots & X_{1,d}^I \\ X_{1,1}^I & X_{1,1}^I & \cdots & X_{1,1}^I \\ \vdots & \vdots & \vdots & \vdots \\ X_{1,1}^I & X_{1,1}^I & \cdots & X_{1,1}^I \end{bmatrix}_{n\times d}$$

$$\text{Prey} = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix}_{n\times d}$$

X_{ij}^I is the vector representing the top predator. d is the number of dimensions. n is the number of search agents. X_{ij} is the jth dimension of the ith Prey.

2.2. MPA optimization scenarios

The optimization scenarios of MPA can be divided into three main phases based on different velocity ratio of predator and prey. In the first phase, the prey moves faster than predator. In the second phase, both predator and prey move at almost same pace. In the third phase, the predator moves faster than prey. Based on the movement rules of predator and prey, a specific period of iteration is specified and assigned for each phase.

Phase 1: In the initial stage of MPA, in high-velocity ratio ($v \geq 10$), the best strategy of the predator is not moving at all. The mathematical model of this phase can be presented as follows.

While $Iter < \frac{1}{3} Max\_Iter$,

$$\begin{aligned} \overrightarrow{stepsize}_i &= \vec{R}_B \otimes (\overrightarrow{Elite}_i - \vec{R}_B \otimes \overrightarrow{\text{Prey}}_i) \qquad i = 1,...,n \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{\text{Prey}}_i + P \cdot \vec{R} \otimes \overrightarrow{stepsize}_i \end{aligned} \tag{1}$$

$\vec{\mathrm{R}}$ is a vector of uniform random numbers in [0, 1]. Iter is the current iteration number. Max_Iter is the maximum iteration number. P = 0.5. $\vec{\mathrm{R}}_{\mathrm{B}}$ is a random vector based on Normal distribution representing the Brownian motion. The standard Brownian motion is a random process. The step size has the characteristics of zero mean ($\mu = 0$) and unit variance ($\sigma^2 = 1$). The symbol $\otimes$ represents entry-wise multiplications.

Phase 2: In unit speed ratio, predators try to make the transition from exploration to exploitation. Therefore, half of the organisms are earmarked for exploration and the other half of them for exploitation.

While $\frac{1}{3} Max\_Iter < Iter < \frac{2}{3} Max\_Iter$, the first half of individuals update their positions based on the following Eq (2):

$$\begin{aligned} \overrightarrow{stepsize}_i &= \vec{R}_L \otimes (\vec{R}_L \otimes \overrightarrow{Elite}_i - \overrightarrow{\text{Prey}}_i) \qquad i = 1,...,n/2 \\ \overrightarrow{\text{Prey}}_i &= \overrightarrow{Elite}_i + P \cdot \vec{R} \otimes \overrightarrow{stepsize}_i \end{aligned} \tag{2}$$

And the second half of individuals update their positions based on the following Eq (3):

$$\begin{aligned} &\overrightarrow{stepsize_i} = \vec{R}_B \otimes (\vec{R}_B \otimes \overrightarrow{Elite_i} - \overrightarrow{\text{Prey}_i}) \qquad i = n/2,...,n \\ &\overrightarrow{\text{Prey}_i} = \overrightarrow{Elite_i} + P \cdot CF \otimes \overrightarrow{stepsize_i} \\ &CF = (1 - \frac{Iter}{Max\_Iter})^{(2 \times \frac{Iter}{Max\_Iter})} \end{aligned} \tag{3}$$

$\vec{\text{R}}_{\text{L}}$ is a vector of random numbers based on Lévy distribution representing Lévy movement. CF is considered as an adaptive parameter to control the step size for predator movement. $\vec{\text{R}}_{\text{B}} \otimes$ Elite is the Brownian motion of a predator chasing its prey. The preys also update their position according to predators in Brownian motion.

Phase 3: Low-velocity ratio or when predator moves faster than prey.

While $Iter > \frac{2}{3} Max\_Iter$, individuals update their positions based on the following Eq (4).

$$\begin{aligned} &\overrightarrow{stepsize_i} = \vec{R}_L \otimes (\vec{R}_L \otimes \overrightarrow{Elite_i} - \overrightarrow{\text{Prey}_i}) \qquad i = 1,...,n \\ &\overrightarrow{\text{Prey}_i} = \overrightarrow{Elite_i} + P \cdot CF \otimes \overrightarrow{stepsize_i} \end{aligned} \tag{4}$$

$\vec{R}_L \otimes \overrightarrow{Elite_i}$ simulates the movement of predators in the Lévy strategy.

2.3. Eddy formation and FADs' effect

The environmental problems can change the behavior of Marine predators, such as the eddy formation and FADs' effect. According to research [32], predators always move around FADs. The FADs are considered as local optima and their effect as trapping in these points. Consideration of these longer jumps during simulation avoids stagnation in local optima. Therefore, the FADs effect can be presented as a mathematical Eq (5).

$$\overrightarrow{\text{Prey}_i} = \begin{cases} \overrightarrow{\text{Prey}_i} + CF[\vec{X}_{\min} + \vec{R} \otimes (\vec{X}_{\max} - \vec{X}_{\min})] \otimes \vec{U} & if \quad r \le FADs \\ \overrightarrow{\text{Prey}_i} + [FADs(1-r) + r](\overrightarrow{\text{Prey}_{r1}} - \overrightarrow{\text{Prey}_{r2}}) & if \quad r > FADs \end{cases} \tag{5}$$

where FADs = 0.2 is the probability of FADs effect on the optimization process. $\vec{U}$ is the binary vector. r is a uniform random number between [0, 1]. $\vec{X}_{max}$ and $\vec{X}_{\min}$ are upper and lower bounds of containing dimensions. r1 and r2 subscripts denote random indexes of prey matrix.

2.4. Marine memory

Marine predators usually have good memories and remember places where they've been successful in foraging. After updating prey and performing FADs effect, this matrix is evaluated for

fitness to update the Elite. Each solution in the current iteration is compared to the previous iteration to determine its fitness. If the current solution is more fitted, it will be replaced. This process also improves the solution quality with the lapse successful foraging.

3. A modified teaching learning Marine Predators Algorithm

The MPA is a recently proposed population-based meta-heuristic algorithm that has been proven to be more competitive with other algorithms. However, the MPA still has several deficiencies to be addressed, such as falling into local optima easily, poor balance ability of exploitation and exploration, weak convergence speed and solution quality.

To solve above-mentioned problems, this paper proposes a modified teaching learning Marine Predators algorithm. In literature [29], we have proposed a kind of teaching learning group mechanism that were introduced in the conventional teaching-learning-based optimization algorithm (namely MTLBO) to enhance solution quality and balance exploration and exploitation. Thus, this paper combines MPA with the teaching learning group mechanism to increase its performance, called MTLMPA. Firstly, the population updating mechanism of the MTLBO in teacher phase is integrated into the first phase of MPA, which can make predators target their preys successfully. Simultaneously, the convergence speed and accuracy of MPA can be improved. Secondly, the population updating mechanism of the MTLBO in learner phase is integrated into the third phase of MPA. Marine predators can simulate students' learning method and optimize their position quickly. Thus, the exploitation and exploration ability of MPA can be enhanced. Now, the variant of MPA is described in the following subsection.

3.1. The population updating mechanism of MTLMPA in first phase

Phase 1: In this phase, a kind of group mechanism is introduced. Firstly, calculate the fitness function values of predators. Based on the fitness values, an elite predator is selected as the most knowledgeable individual in the population. Secondly, calculate the mean value of predators' position, noted as $\mathrm{Prey}_{\mathrm{mean}}$. Based on the mean value, predators are divided into two groups. One group contains superior predators, another group are poor predators. During the food capture period, these superior predators update their position mainly rely on their experience and elite predator. And those poor predators mainly follow the elite predator to capture prey. Finally, all predators still move in Brownian motion. Therefore, the specific model is presented as follows.

$$Prey_i = \begin{cases} Prey_i \otimes w + rand \otimes P.R \otimes R_B \otimes [\mathrm{Elite}_i - R_B \otimes Prey_i], \\ \qquad\qquad f(Prey_i) < f(Prey_{mean}) \\ [Prey_i + 2 \times (rand - 0.5) \otimes P.R \otimes R_B \otimes (Prey_{mean} - R_B \otimes Prey_i)] \times \sin(\frac{\pi}{2} \frac{iter}{Max\_Iter}) \\ \quad + diff \times \cos(\frac{\pi}{2} \frac{iter}{Max\_Iter}) \times P.R, \\ \qquad\qquad f(Prey_i) > f(Prey_{mean}) \end{cases} \tag{6}$$

where w is the inertia weight. $w = w_{start} - (w_{start} - w_{end}) \times \frac{iter}{Max\_Iter}$ The inertia weight is linear decreasing, which is helpful to improve the local development ability of the algorithm. $sin(\frac{\pi}{2} \frac{iter}{Max\_Iter})$,

$cos(\frac{\pi}{2}\frac{iter}{Max\_Iter})$ are two weight coefficients.$sin(\frac{\pi}{2}\frac{iter}{Max\_Iter})$ increases with increasing iteration and gradually tends towards 1. $cos(\frac{\pi}{2}\frac{iter}{Max\_Iter})$ decreases with increasing iteration and gradually decreases to 0.

3.2. The population updating mechanism of MTLMPA in third phase

Phase 3: This stage is the low speed ratio or when the predators is moving faster than the prey. This procedure is frequently associated with strong exploitation capability. In this phase, the Lévy flight mode is the best strategy for predators. When the new population updating mechanism is introduced, all predators are still divided into two groups based on their fitness function values. After sorting the fitness values, the first half of predators are regarded as the superior predator. Simultaneously, the rest of the predators are considered as inferior predator. Superior predators have strong predation capability, so they update their position rely on themself information and elite predator information during the predation. Moreover, superior predators can learn to hunt by themselves. Additionally, inferior predators have relatively weak predation ability, so that they only follow the elite predator to hunt. Based on the phenomenon, the specific mathematical model is summarized as follows.

$$Prey_i = \begin{cases} \begin{cases} Prey_i + P.CF \times \cos(\frac{\pi}{2}\frac{iter}{Max\_Iter}) \otimes R_L \otimes [(R_L \otimes Prey_{\text{neighbor}} - Prey_i)], \\ \qquad\qquad f(Prey_i) > f(Prey_{neighbor}) \\ Prey_i + P.CF \otimes R_L \otimes [2 \times (rand - 0.5) \otimes R_L \otimes (Prey_{upper\lim} - Prey_{lower\lim})], \\ \qquad\qquad f(Prey_i) < f(Prey_{neighbor}) \end{cases} \\ Prey_i + P.CF \times \cos(\frac{\pi}{2}\frac{iter}{Max\_Iter}) \otimes R_L \otimes (R_L \otimes Elite_i - Prey_i) \end{cases} \tag{7}$$

Seen from Eq (7), the superior predators randomly chose a nearby predator $Prey_{\text{neighbor}}$ to follow. If the $Prey_i$ is better than $Prey_{\text{neighbor}}$, the $Prey_i$ mainly updates its position by itself. Otherwise $Prey_i$ learns from $Prey_{\text{neighbor}}$. The weight coefficient $cos(\frac{\pi}{2}\frac{iter}{Max\_Iter})$ is introduced to improve convergence speed and local exploitation ability.

The flow chart of MTLMPA is presented as follows.

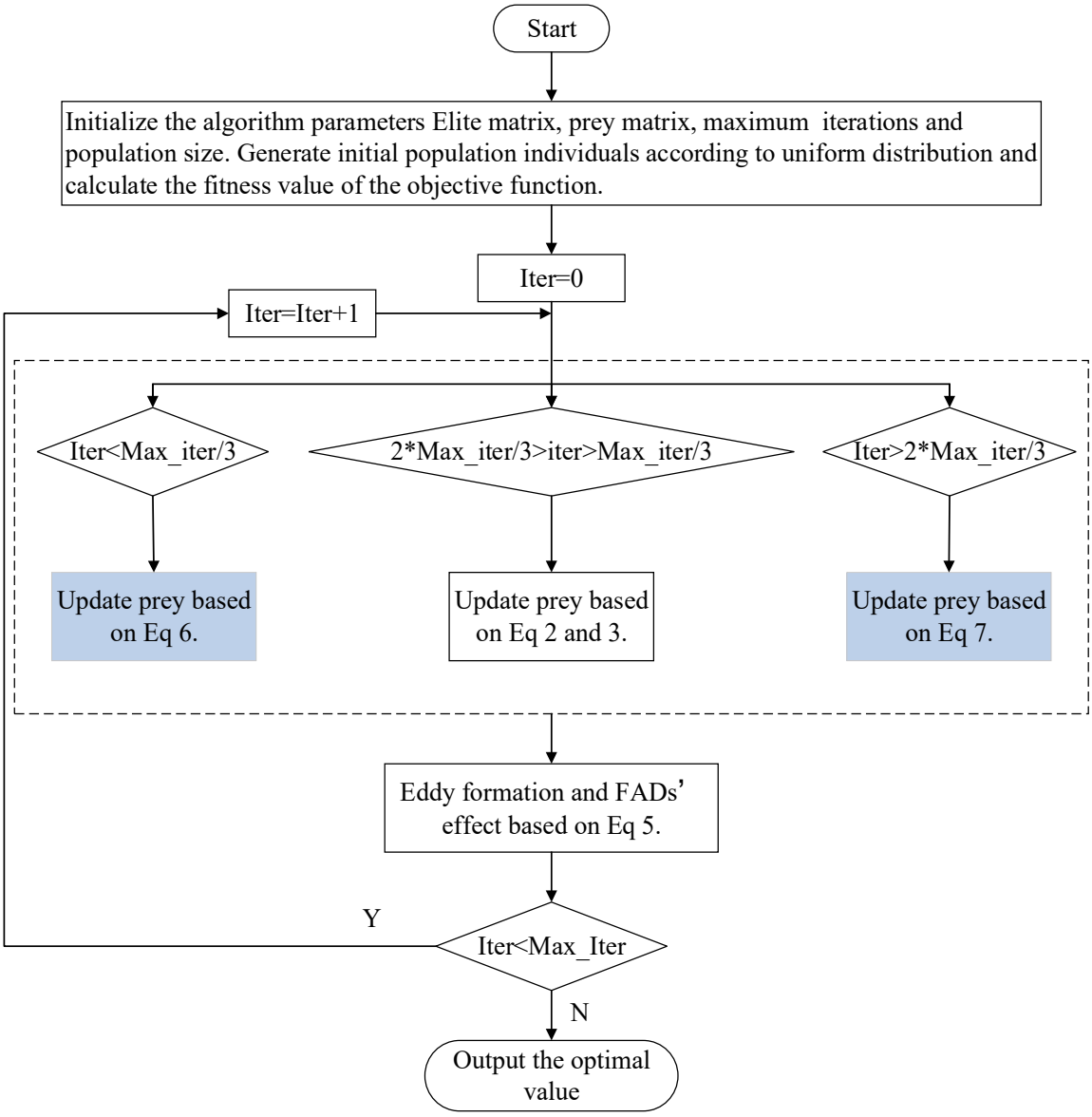

Figure 1. Flow chart of MTLMPA.

4. Performance testing

In order to verify the effectiveness of MTLMPA, 23 benchmark testing functions are applied to evaluate MTLMPA's performance in exploration, exploitation and minimization. The detailed description of these functions are presented in Table 1. Seen from Table 1, TF1–TF7 belong to the unimodal functions which are used to evaluate the exploitation capability of MTLMPA. TF8–TF13 simulate multi-modal functions to test the exploration performance of MTLMPA. The functions TF14–TF23 with fixed dimensions are used to test the algorithm's performance in low dimensions. In addition, 29 CEC-2017 testing functions are used to verify the performance of MTLMPA as well.

All testing were conducted on a single machine. CPU: Intel (R) Core (TM) i5-6300HQ CPU @ 2.30 GHz, Windows10 operating system and MATLAB R2016a. The population number is set as 40. The maximum iteration number is set as 500. In order to reduce statistical errors, each

algorithm is independently simulated 30 times.

Table 1. 23 Benchmark testing functions.

	Dimention	Range	Global solution
$TF_1(x)=\sum_{i=1}^{d}x_i^2$	10, 50, 100	[-100, 100]^n	0
$TF_2(x)=\sum_{i=1}^{d}\lvert x_i\rvert+\Pi_{i=1}^{d}\lvert x_i\rvert$	10, 50, 100	[-100, 100]^n	0
$TF_3(x)=\sum_{i=1}^{d}(\sum_{j=1}^{d}x_j)^2$	10, 50, 100	[-100, 100]^n	0
$TF_4(x)=Max\{\lvert x_i\rvert,1\le i\le d\}$	10, 50, 100	[-100, 100]^n	0
$TF_5(x)=\sum_{i=1}^{d-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	10, 50, 100	[-30, 30]^n	0
$TF_6(x)=\sum_{i=1}^{d-1}([x_i+0.5])^2$	10, 50, 100	[-100, 100]^n	0
$TF_7(x)=\sum_{i=1}^{d}ix_i^4+random\ [0,1]$	10, 50, 100	[-1.28, 1.28]^n	0
$TF_8(x)=\sum_{i=1}^{d}-x_i\sin(\sqrt{\lvert x_i\rvert})$	10, 50, 100	[-500, 500]^n	−418.98 × d
$TF_9(x)=\sum_{i=1}^{d}[x_i^2-10\cos(2\pi x_i)+10]$	10, 50, 100	[-5.12, 5.12]^n	0
$TF_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2})-\exp(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i))+20+e$	10, 50, 100	[-32, 32]^n	0
$TF_{11}(x)=-\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\Pi_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}})+1$	10, 50, 100	[-600, 600]^n	0
$TF_{12}(x)=-\frac{\pi}{d}\{10\sin(\pi y_1)+\sum_{i=1}^{d-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_d-1)^2\}+\sum_{i=1}^{d}u(x,10,100,4)$	10, 50, 100	[-50, 50]^n	0
$TF_{13}(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{d-1}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_d-1)^2[1+\sin^2(2\pi x_d)]\}+\sum_{i=1}^{d}u(x_i,5,100,4)$	10, 50, 100	[-50, 50]^n	0
$TF_{14}(x)=(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6})^{-1})$	2	[-65, 65]	1
$TF_{15}(x)=\sum_{i=1}^{11}[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$	4	[-5, 5]	0.00030
$TF_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5, 5]	−1.0316
$TF_{17}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	2	[-5, 5]	0.398
$TF_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	2	[-2, 2]	3
$TF_{19}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$	3	[1, 3]	-3.86
$TF_{20}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	[0, 1]	-3.32
$TF_{21}(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	−10.1532
$TF_{22}(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	−10.4028
$TF_{23}(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	[0, 10]	−10.536

4.1. Comparison with MPA

In this subsection, the performance comparisons between MTLMPA and MPA are given on 23 testing benchmark functions with 50 dimensions, including exploitation capability evaluation, exploration capability evaluation and algorithm convergence ability. For every testing function, the MTLMPA and MPA independently run 30 times to find the global optima solution, separately. And then, we find the best solution in the thirty times results, which is denoted as Best. Finally, we calculate the mean and standard deviation of the thirty results, separately. The mean value is recorded as Ave and the standard deviation is abbreviated as Std. The mean value represents convergence accuracy and the standard deviation represents stability of algorithm. Noted that: the smaller the mean and standard deviation, the better the algorithm performs.

4.1.1. Exploitation capability evaluation

Uni-modal functions are real-valued functions that have a single strictly local maximum in the interval. There is only one global optimal solution in each testing function. These functions are fitness to evaluate the exploitation ability of the optimization algorithm. Therefore, functions TF1–TF7 are used to investigate the exploitation capability of MPA and MTLPA. The experiment results are recorded in the Table 2. The best performance index is presented in bold font.

Table 2. Testing results of MTLMPA and MPA on seven uni-modal functions.

Functions	Performance Index	Method	
		MTLMPA	MPA
TF1	Best	$\mathbf{8.98 \times 10^{-86}}$	6.22×10^{-21}
	Ave	$\mathbf{1.70 \times 10^{-86}}$	1.25×10^{-20}
	Std	$\mathbf{5.43 \times 10^{-86}}$	1.42×10^{-20}
TF2	Best	$\mathbf{4.68 \times 10^{-47}}$	7.53×10^{-12}
	Ave	$\mathbf{1.86 \times 10^{-45}}$	5.04×10^{-12}
	Std	$\mathbf{4.14 \times 10^{-45}}$	5.00×10^{-12}
TF3	Best	$\mathbf{1.24 \times 10^{-41}}$	0.079
	Ave	$\mathbf{3.90 \times 10^{-41}}$	0.067
	Std	$\mathbf{1.03 \times 10^{-40}}$	0.102
TF4	Best	$\mathbf{1.11 \times 10^{-36}}$	2.23×10^{-8}
	Ave	$\mathbf{4.98 \times 10^{-38}}$	3.52×10^{-8}
	Std	$\mathbf{2.06 \times 10^{-37}}$	1.26×10^{-8}
TF5	Best	47.746	46.384
	Ave	47.454	46.045
	Std	0.925	0.369
TF6	Best	1.238	0.188
	Ave	1.463	0.296
	Std	0. 181642	0.160
TF7	Best	$\mathbf{1.54 \times 10^{-4}}$	0.002
	Ave	$\mathbf{2.51 \times 10^{-4}}$	0.001
	Std	$\mathbf{5.00 \times 10^{-5}}$	7.70×10^{-4}

Seen from Table 2, we can find that the proposed MTLMPA shows better performance than conventional MPA on five functions, including TF1, TF2, TF3, TF4, TF7. On functions TF5 and TF6, the MPA presents relatively better performance than MTLMPA. Therefore, these experiment results

reveal that the MTLMPA has stronger exploitation capability than the MPA.

4.1.2. Exploration ability evaluation

In general, multi-modal testing functions have a large number of local optimal values, so that these functions are fitness to verify the exploration ability of optimization algorithm. Functions TF8–TF13 are the multi-modal function with high dimensions. Functions TF14–TF23 are the multi-modal function with fixed (low) dimensions. Therefore, these multi-modal functions are used to evaluate the exploration ability of MTLMPA and MPA. Experiment results of MTLMPA and MPA are recorded in Tables 3 and 4, separately.

Shown in Table 3, for high-dimensional multi-modal functions, the proposed MTLMPA presents better performance than MPA on TF8–TF11. Moreover, the two optimization algorithms obtain similar results on TF12 and TF13. Seen from Table 4, for fixed dimensional multi-modal functions, MTLMPA and MPA can find the global optima solution on all testing functions. Although they have the similar convergence accuracy, the MTLMPA owns stronger algorithm stability than conventional MPA. In conclusion, the MTLMPA has better exploration capability than MPA on most testing functions.

Table 3. Testing results of MTLMPA and MPA on six multi-modal functions.

Functions	MTLMPA			MPA		
	Best	Ave	Std	Best	Ave	Std
TF8	-1.08×10^{4}	-1.16×10^{4}	8.80×10^{2}	-1.29×10^{4}	-1.36×10^{4}	8.35×10^{2}
TF9	0	0	0	0	0	0
TF10	8.88×10^{-16}	8.88×10^{-16}	0	1.13×10^{-11}	1.75×10^{-11}	1.15×10^{-11}
TF11	0	0	0	0	0	0
TF12	0.033	0.03	0.01	0.005	0.007	0.004
TF13	0.075	1.704	1.86	0.617	0.309	0.148

Table 4. Testing results of MTLMPA and MPA on ten fixed dimension functions.

Functions	MTLMPA			MPA		
	Best	Ave	Std	Best	Ave	Std
TF14	0.998	0.998	9.22×10^{-17}	0.998	0.998	1.89×10^{-16}
TF15	3.07×10^{-4}	3.07×10^{-4}	9.78×10^{-18}	0.998	0.998	1.89×10^{-16}
TF16	-1.032	-1.032	6.32×10^{-16}	-1.032	-1.032	4.70×10^{-16}
TF17	0.398	0.398	0	0.398	0.398	4.51×10^{-16}
TF18	3	3	1.35×10^{-15}	3	3	1.81×10^{-15}
TF19	-3.863	-3.863	2.67×10^{-15}	-3.863	-3.863	2.29×10^{-15}
TF20	-3.322	-3.322	9.85×10^{-11}	-3.322	-3.322	2.02×10^{-13}
TF21	-10.153	-9.813	1.29	-10.153	-10.153	3.08
TF22	-10.403	-10.226	9.70×10^{-12}	-10.403	-10.403	2.73×10^{-12}
TF23	-10.536	-10.536	1.37	-10.536	-10.536	-10.2

4.1.3. Analysis of convergence performance

In this subsection, several simulation Figures of testing functions are given to analyse the convergence accuracy and convergence speed of two algorithms. Seen from these figures, it is easy to contrast the convergence performance of the two algorithms. The blue line is the convergence curve of MPA. The green line is the convergence curve of MTLMPA. Seen from Figures 2 to 7, we can find that the MTLMPA has better convergence accuracy and convergence speed than MPA.

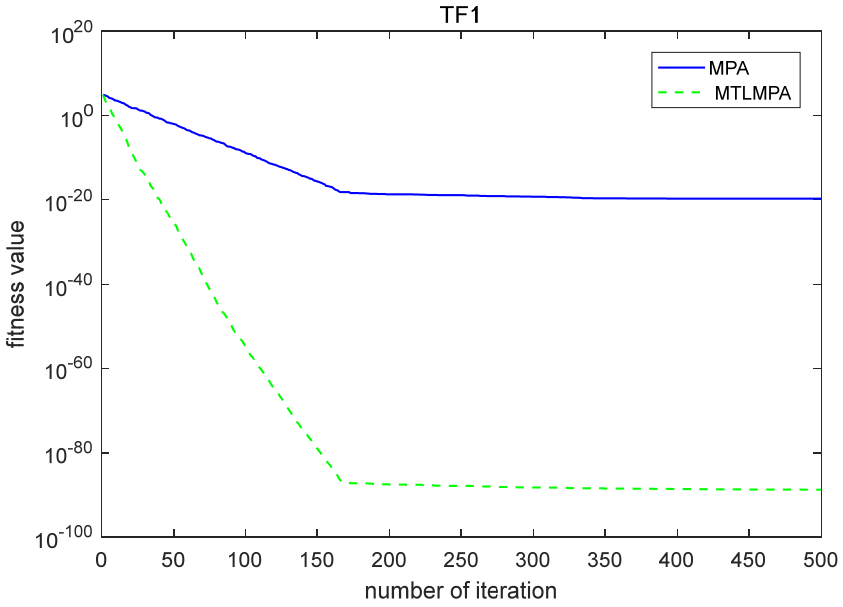

Figure 2. Convergence curves of two algorithms on TF1.

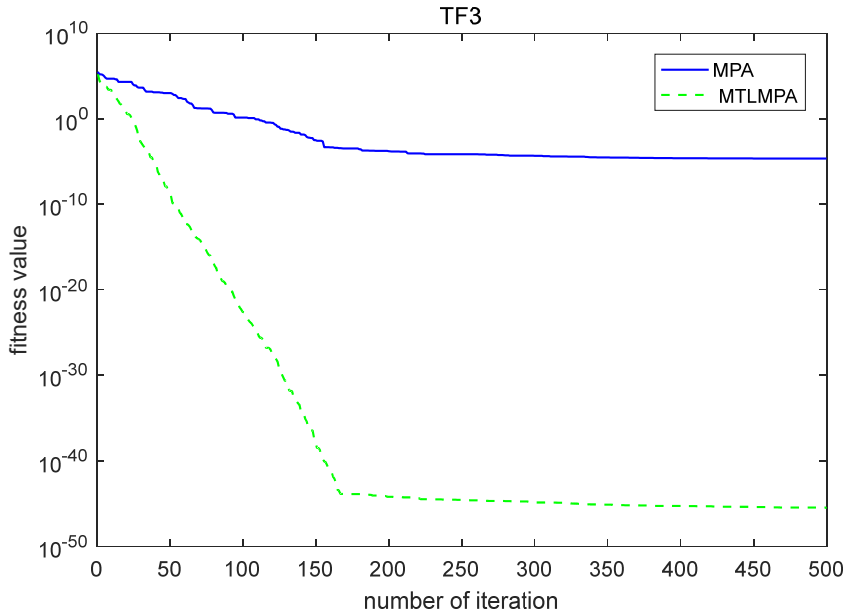

Figure 3. Convergence curves of two algorithms on TF3.

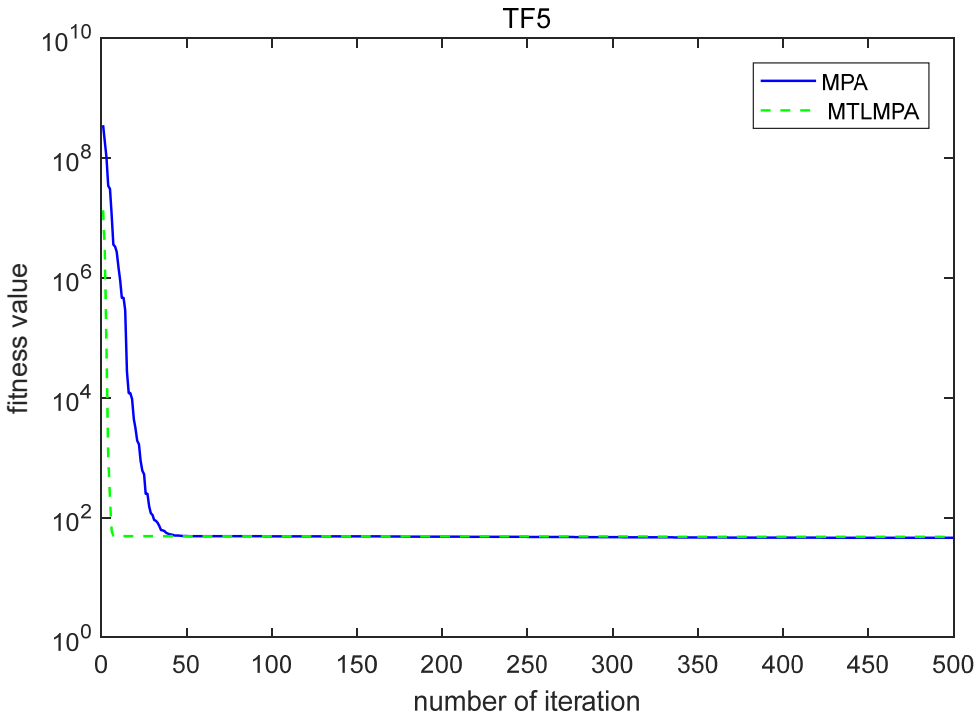

Figure4. Convergence curves of two algorithms on TF5.

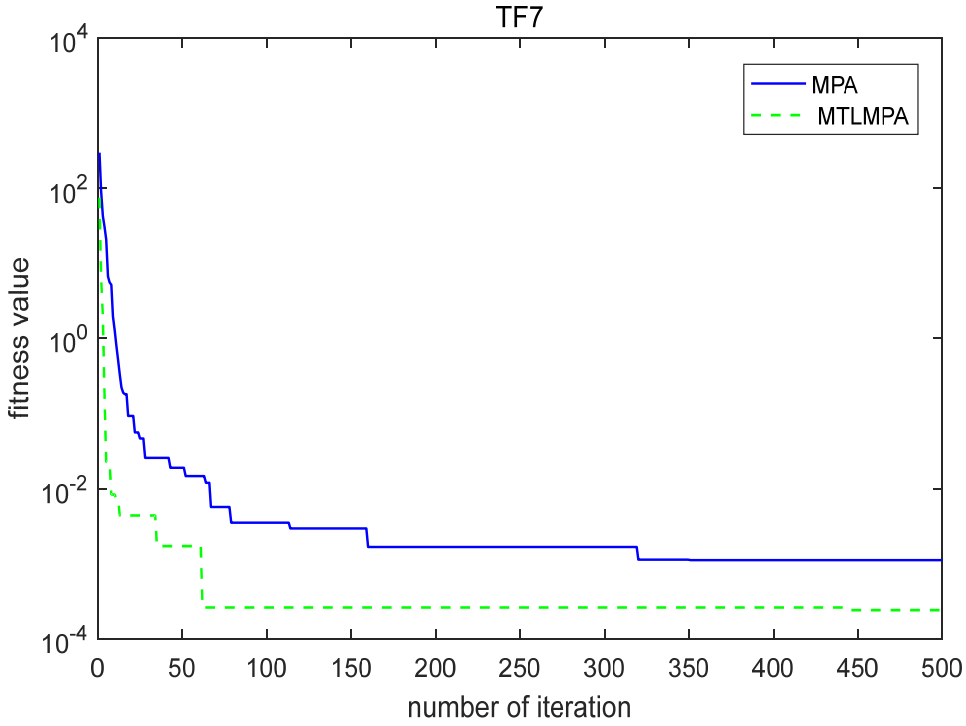

Figure5. Convergence curves of two algorithms on TF7.

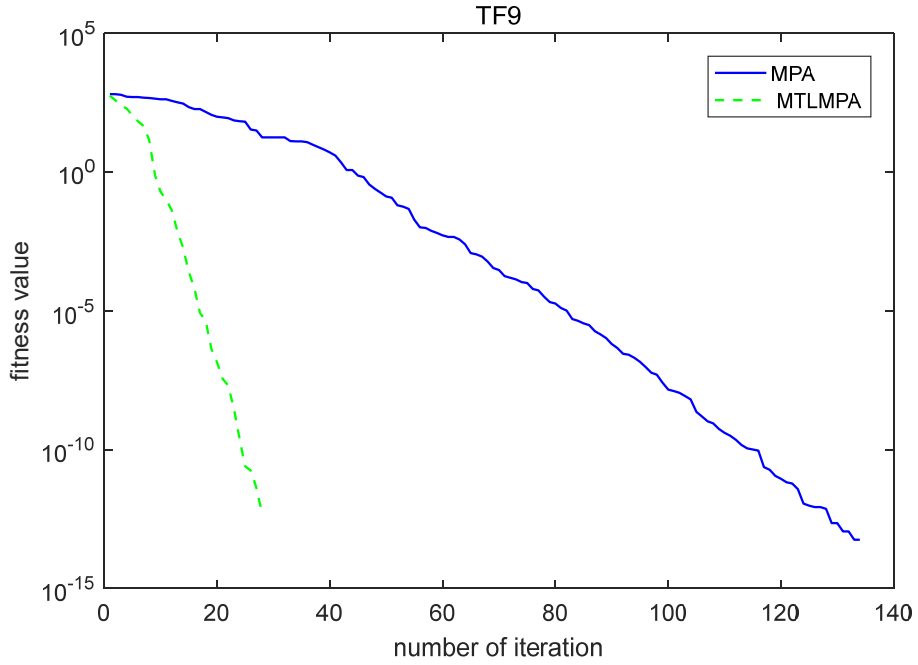

Figure6. Convergence curves of two algorithms on TF9.

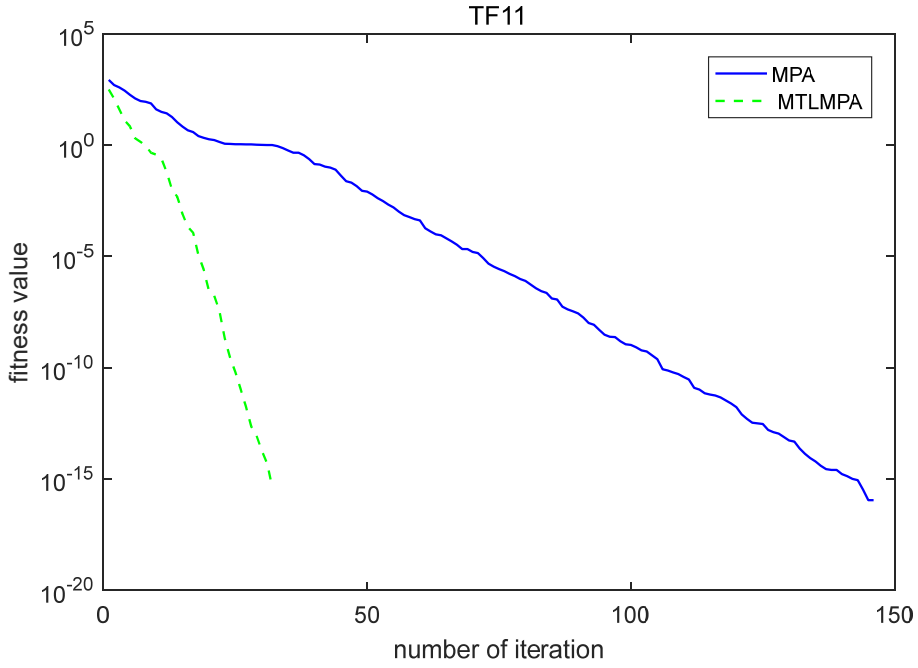

Figure7. Convergence curves of two algorithms on TF11.

4.1.4. Test on CEC-2017 functions

The CEC-2017 contains 29 benchmark functions for evaluating optimization problems. These functions can be divided into four categories: uni-modal function, multi-modal function, mixed

function and combined function. The uni-modal function and the multi-modal function would be much more complex in this section. The combined function and mixed function are considered in test. In the mixed function, the variable is randomly divided into several sub-components, and different basic functions are used for different sub-components. The composition function better integrates the properties of the sub-functions and maintains the continuity of the optimal solution. This section compared MPA with MTLMPA on CEC-2017. The specific results are listed in Table 5.

Table 5. Test results of CEC-2017.

Functions	MTLMPA			MPA		
	Best	Ave	Std	Best	Ave	Std
F1	100.069	100.172	0.117	100.001	443.759	432.866
F2	100.085	100.906	1.361	100.079	818.407	1177.726
F3	100.039	100.395	0.506	100.013	2223.573	3067.584
F4	100.059	109.633	29.517	100.003	1378.010	1949.167
F5	100.025	100.196	0.196	100.002	265.194	467.395
F6	100.042	100.779	1.787	100.007	921.372	1544.781
F7	100.063	102.485	6.005	100.018	1289.083	2380.034
F8	100.134	100.656	6.005	100.009	2304.526	2380.034
F9	100.062	100.194	0.123	100.004	643.250	759.905
F10	759.905	100.400	0.61	100.014	462.052	292.612
F11	100.040	100.147	0.073	100.001	231.227	261.141
F12	100.051	100.572	1.344	100.003	1051.582	1482.336
F13	100.079	100.348	0.257	100.001	245.897	400.007
F14	100.057	100.242	0.154	100.040	2505.454	3670.958
F15	100.031	100.225	0.178	100.008	1680.989	1947.772
F16	100.092	100.296	0.189	100.001	802.063	1461.193
F17	100.029	100.318	0.325	100.001	1093.011	2531.992
F18	100.073	108.614	25.923	100.001	679.102	721.708
F19	100.082	100.718	1.235	100.002	161.700	101.933
F20	100.042	100.250	0.250	100.220	382.136	617.024
F21	100.069	100.875	1.234	100.220	1392.539	1832.370
F22	100.061	100.209	0.122	100.003	1425.010	2170.169
F23	100.085	101.481	3.997	100.002	509.307	656.829
F24	100.023	100.622	1.101	100.001	662.150	1282.155
F25	100.001	462.784	1146.423	100.045	925.512	1746.087
F26	100.082	100.451	0.478	100.001	1567.273	1757.914
F27	100.043	100.363	0.384	100.021	1255.151	1238.848
F28	100.033	100.207	0.183	101.179	865.255	0.183
F29	100.107	101.253	1.971	111.470	1771.598	3096.588

As can be seen from the Table 5, the results of MPA and MTLMPA are similar in terms of optimal values. However, there are huge differences in mean and standard deviations. The MTLMPA has a strong stability. The MTLMPA converged to the optimal value each time with 500 iterations. The MPA did not converge to the optimum in the partial tests and the standard deviations are also large.

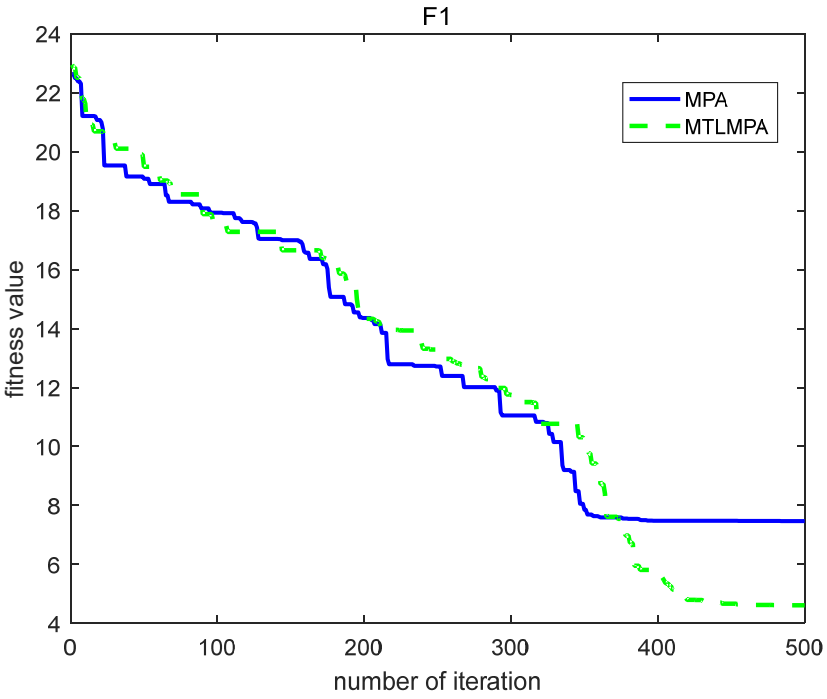

Figure 8. Convergence curves of MTLMPA and MPA on CEC2017 F1.

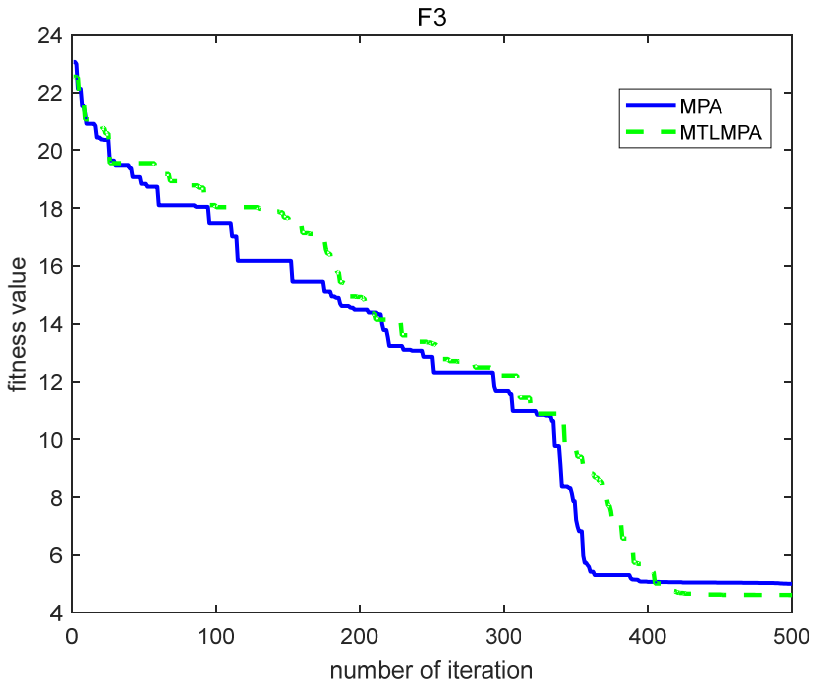

Figure 9. Convergence curves of MTLMPA and MPA on CEC2017 F3.

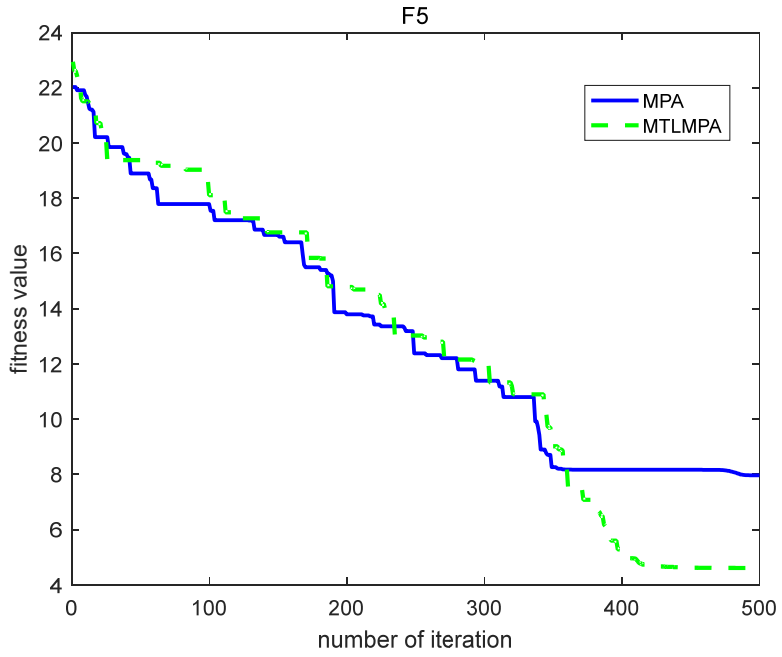

Figure 10. Convergence curves of MTLMPA and MPA on CEC2017 F5.

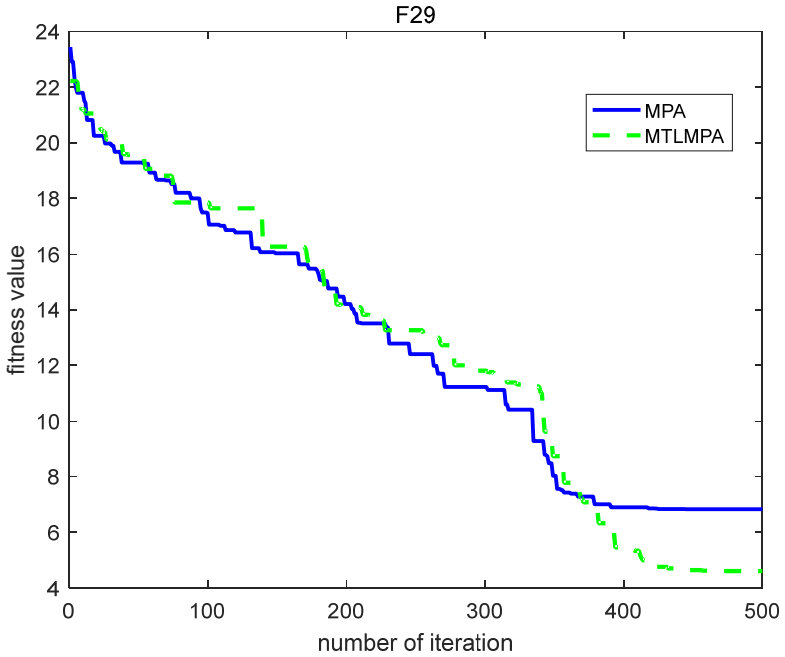

Figure 11. Convergence curves of MTLMPA and MPA on CEC2017 F2.

The figures show that the convergence value of MTLMPA is significantly better than MPA. And there exist no convergence in some results of MPA. In conclusion, MTLMPA has strong stability and can accurately converge to the optimal values. Its application value is high compared with MPA.

4.2. Performance comparison with other algorithms

In order to further evaluate the performance of MTLMPA, 23 benchmark testing functions with different dimensions (including 10, 50, 100 dimensions) are used. Moreover, several state-of-the-art optimization algorithms are applied and considered as comparison algorithm, including PSO, GWO, SCA, WOA, CS, KH, MPA and TLBO etc. For the fairness of the comparison, the number of population individual is set as 40 for every algorithm, the maximum iteration number is set as 500, each algorithm is independently simulated 30 times. The unique parameters of every algorithm are set based on their requirements.

Table 6. Mean of function fitness value on TF1–TF13 with 10 dimensions.

Functions	PSO	CS	GWO	KH	SCA	WOA	TLBO	MPA	MTLMPA
TF1	4.92×10^{-23}	1.25×10^{-4}	1.95×10^{-64}	0	3.09×10^{-14}	3.50×10^{-83}	1.16×10^{-108}	6.81×10^{-30}	2.13×10^{-102}
TF2	1.17×10^{-12}	0.0118	9.89×10^{-37}	7.43×10^{-172}	4.27×10^{-10}	8.57×10^{-56}	2.60×10^{-58}	5.87×10^{-17}	4.94×10^{-55}
TF3	2.73×10^{-7}	0.177	0.115	0	0.005	1.39×10^{2}	2.90×10^{-55}	1.72×10^{-14}	1.89×10^{-65}
TF4	5.48×10^{-6}	0.136	8.10×10^{-21}	6.36×10^{-170}	4.09×10^{-4}	1.932	5.62×10^{-45}	1.50×10^{-12}	3.13×10^{-47}
TF5	5.895	5.886	6.531	8.587	7.292	6.721	8.516	1.4805	1.295
TF6	3.14×10^{-23}	1.49×10^{-4}	0.008	0.999	0.419	3.08×10^{-4}	0.594	1.16×10^{-12}	7.76×10^{-8}
TF7	0.007	0.002	4.75×10^{-4}	1.21×10^{-4}	0.002	0.002	0.002	6.14×10^{-4}	1.76×10^{-4}
TF8	-2.38×10^{3}	-2.85×10^{23}	-2.76×10^{3}	-1.45×10^{3}	-2.23×10^{3}	-3.53×10^{3}	-2.79×10^{3}	-3.68×10^{3}	-3.69×10^{3}
TF9	4.562	3.7427	0.650	0	0.640	1.252	0.961	9.19×10^{-14}	0
TF10	3.86×10^{-12}	0.068	6.81×10^{-15}	1.70×10^{-15}	2.67×10^{-6}	4.20×10^{-15}	5.27×10^{-15}	4.91×10^{-15}	8.88×10^{-16}
TF11	0.191	0.245	0.037	0	0.107	0.064	0.018	3.30×10^{-4}	0
TF12	5.21×10^{-24}	0.003	0.003	0.450	0.081	0.007	0.101	6.14×10^{-13}	1.87×10^{-8}
TF13	3.12×10^{-23}	3.98×10^{-4}	0.016	0.833	0.271	0.021	0.324	2.73×10^{-12}	3.24×10^{-6}

Table 7. Std of function fitness value on TF1–TF13 with 10 dimensions.

Functions	PSO	CS	GWO	KH	SCA	WOA	TLBO	MPA	MTLMPA
TF1	2.09×10^{-22}	1.08×10^{-4}	5.35×10^{-64}	0	1.06×10^{-13}	1.91×10^{-82}	6.25×10^{-108}	1.68×10^{-29}	1.07×10^{-10}
TF2	2.31×10^{-12}	0.006	2.22×10^{-36}	0	6.42×10^{-10}	4.34×10^{-55}	4.99×10^{-58}	7.07×10^{-17}	1.18×10^{-54}
TF3	2.64×10^{-7}	0.115	3.35×10^{-28}	0	0.017	2.36×10^{2}	1.47×10^{-54}	3.40×10^{-14}	7.00×10^{-65}
TF4	8.27×10^{-6}	0.044	1.06×10^{-20}	0	0.001	4.398	1.88×10^{-44}	1.92×10^{-12}	1.66×10^{-46}
TF5	5.372	3.302	0.541	0.026	0.401	0.318	0.375	0.417	0.526
TF6	7.64×10^{-23}	8.35×10^{-5}	0.046	0.263	0.143	2.71×10^{-4}	0.37025	7.78×10^{-13}	1.28×10^{-7}
TF7	0.003	0.00174	3.53×10^{-4}	9.73×10^{-5}	0.002	0.002	0.002	3.32×10^{-4}	1.30×10^{-14}
TF8	4.44×10^{2}	1.20×10^{-24}	3.30×10^{2}	2.48×10^{2}	1.77×10^{2}	5.53×10^{2}	2.18×10^{2}	1.55×10^{2}	2.11×10^{2}
TF9	2.391	3.342	1.785	0	3.501	6.859	3.389	5.03×10^{-13}	0
TF10	3.21×10^{-12}	0.037	1.70×10^{-15}	0	1.18×10^{-5}	2.79×10^{-15}	1.53×10^{-15}	1.23×10^{-15}	0
TF11	0.132	0.060	0.057	0	0.199	0.096	0.029	0.002	0
TF12	1.18×10^{-23}	0.004	0.007	0.330	0.024	0.013	0.134	7.36×10^{-13}	2.90×10^{-8}
TF13	3.12×10^{-23}	3.98×10^{-4}	0.016	0.833	0.271	0.021	0.324	2.73×10^{-12}	3.24×10^{-6}

As illustrated in Table 6, MTLMPA can still converge to an accurate value in the situation of low dimension. MTLMPA has a significantly higher accuracy than MPA. MTLMPA and KH algorithms continue to have significant benefits over other algorithms in TF1, TF2, TF3, TF4, TF7, TF9, TF11, TF13. In particular, MTLMPA performs well than KH in TF5, TF6, TF10, TF12. At the same time, the MTLMPA is stable. And the standard deviation of 30 cycles is small. It can be seen that MTLMPA converges faster than other algorithms in TF2, TF3, TF4, TF6, TF7, TF9, TF10, TF11, TF12. And the algorithm is stable. It can find the optimal solution to the maximum extent within its own allowable range.

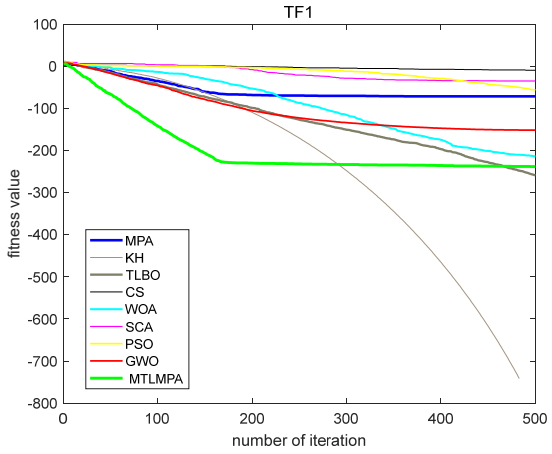

Figure 12. Convergence curves on TF1, dim = 10.

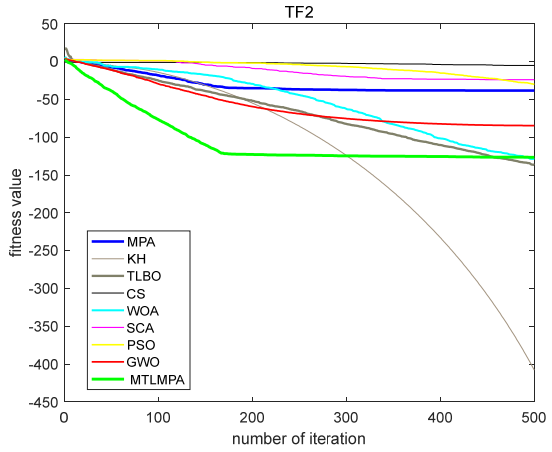

Figure 13. Convergence curves on TF2, dim = 10.

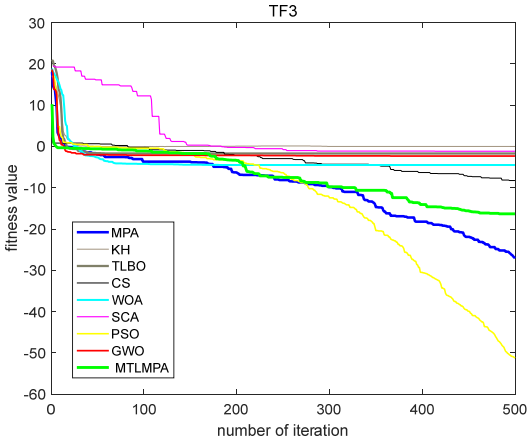

Figure 14. Convergence curves on TF3, dim = 10.

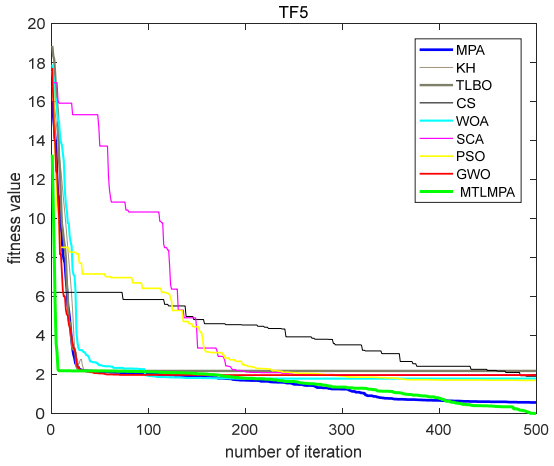

Figure 15. Convergence curves on TF5, dim = 10.

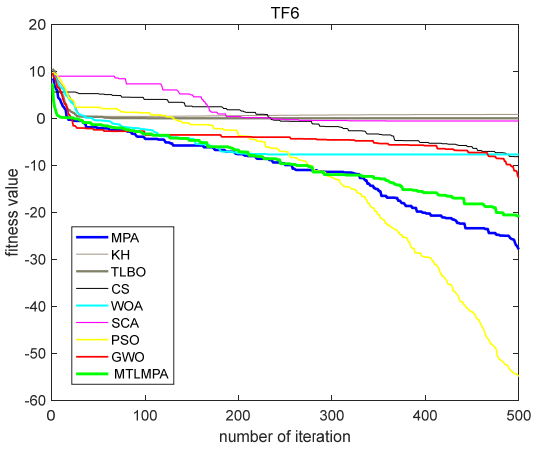

Figure 16. Convergence curves on TF6, dim = 10.

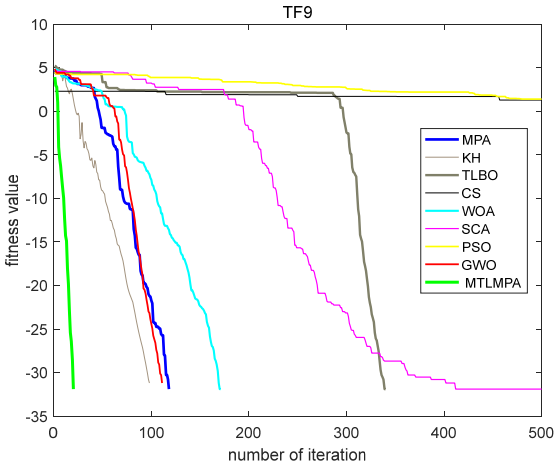

Figure 17. Convergence curves on TF9, dim = 10.

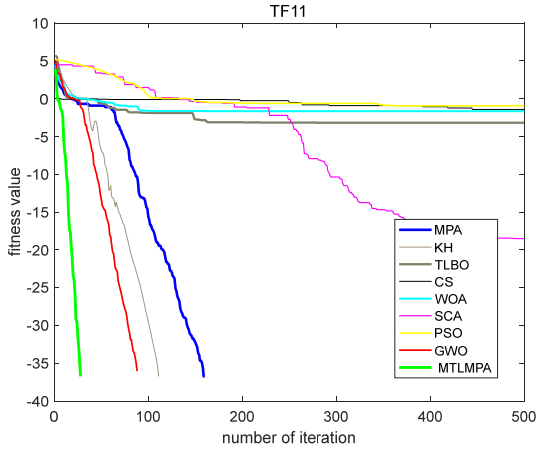

Figure 18. Convergence curves on TF11, dim = 10.

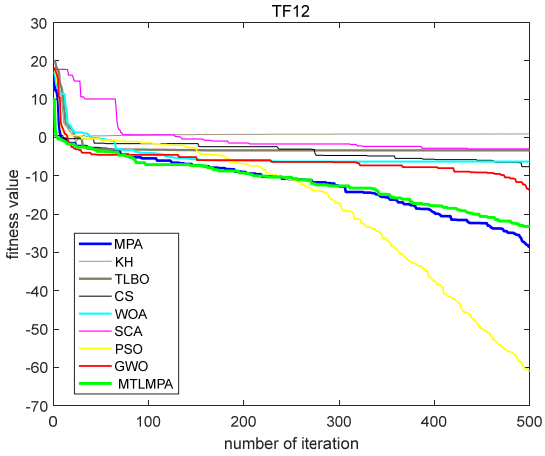

Figure 19. Convergence curves on TF12, dim = 10.

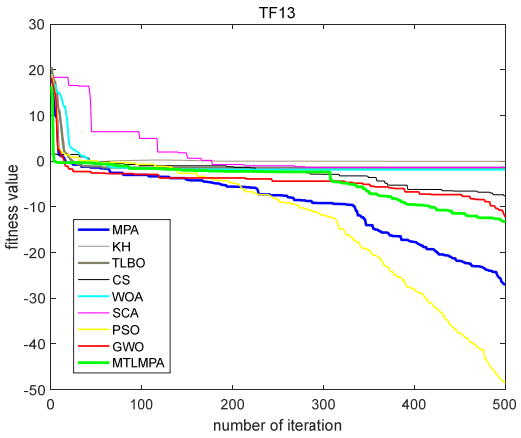

Figure 20. Convergence curves on TF13, dim = 10.

Table 8. Mean of function fitness value on TF1–TF13 with 50 dimensions.

Functions	PSO	CS	GWO	KH	SCA	WOA	TLBO	MPA	MTLMPA
TF1	0.079	1.39×10^{-56}	3.41×10^{-22}	0.000	6.51×10^{2}	3.85×10^{-78}	3.37×10^{-101}	1.44×10^{-20}	1.02×10^{-88}
TF2	0.818	6.45×10^{-5}	1.45×10^{-13}	6.03×10^{-170}	0.566	6.62×10^{-51}	2.52×10^{-57}	3.98×10^{-12}	6.33×10^{-47}
TF3	1.18×10^{3}	2.59×10^{3}	0.051	0.000	4.21×10^{2}	1.68×10^{5}	4.78×10^{-19}	0.034	2.88×10^{-39}
TF4	3.064	2.435	1.02×10^{-4}	1.24×10^{-18}	65.602	67.693	9.45×10^{-34}	3.62×10^{-8}	7.36×10^{-39}
TF5	2.98×10^{2}	4.35×10^{2}	47.073	48.460	3.85×10^{6}	47.885	48.910	46.256	45.728
TF6	0.071	35.000	2.218	10.656	6.70×10^{2}	0.672	10.337	0.239	1.428
TF7	1.655	0.007	0.002	1.06×10^{-4}	3.262	0.002	0.003	0.002	2.67×10^{-4}
TF8	-8.48×10^{3}	-1.34×10^{21}	-9.25×10^{3}	-3.40×10^{3}	-5.00×10^{3}	-1.86×10^{4}	-4.18×10^{3}	-1.34×10^{4}	-1.19×10^{4}
TF9	1.48×10^{2}	35.353	4.529	0.000	1.07×10^{2}	0.000	0.000	0.000	0.000
TF10	1.346	2.600	3.10×10^{-12}	8.88×10^{-16}	15.235	4.91×10^{-15}	8.88×10^{-16}	1.63×10^{-11}	8.88×10^{-16}
TF11	0.008	1.290	0.004	0.000	6.542	0.000	0.000	0.000	0.000
TF12	0.058	0.367	0.087	1.019	1.33×10^{7}	0.020	4.71×10^{-31}	0.007	0.026
TF13	0.081	2.825	1.794	4.944	2.19×10^{7}	0.758	1.35×10^{-32}	0.324	1.075

As can be observed from the Table 8, MTLMPA performs the best performance on functions TF3, TF5, TF9, TF10, TF11, TF14, TF15, TF16, TF17, TF18, TF19 and TF20. Although others functions are not optimal, it is not an order of magnitude away from optimal accuracy. In general, MTLMPA is an algorithm that can accurately converge to the optimal value.

At the same time, all the standard deviations of each algorithm are compared in Table 9. MPA has strong stability. MTLMPA also has excellent performance in standard deviation. It surpassed most of the algorithms on TF2, TF3, TF4, TF7, TF9, TF10, TF11, TF13. To better illustrate each algorithm's convergence performance, the simulating curves are produced as illustrated in the image below. Seen from these figures, the convergence speed and convergence accuracy are better than other algorithms on most functions.

Table 9. Std of function fitness value on TF1–TF13 with 50 dimensions.

Function	PSO	CS	GWO	KH	SCA	WOA	TLBO	MPA	MTLMPA
TF1	0.097	1.10×10^{-38}	3.04×10^{-22}	0.000	6.09×10^{2}	1.71×10^{-77}	1.16×10^{-100}	1.44×10^{-20}	4.81×10^{-8}
TF2	0.460	1.99×10^{-2}	7.64×10^{-14}	0.000	0.630	3.51×10^{-50}	1.07×10^{-56}	4.25×10^{-12}	1.10×10^{-46}
TF3	3.02×10^{2}	0.000	0.140	0.000	1.14×10^{4}	3.61×10^{4}	1.48×10^{-18}	0.042	8.73×10^{-39}
TF4	0.380	2.282	9.20×10^{-5}	0.000	7.882	23.616	2.26×10^{-33}	2.01×10^{-8}	1.91×10^{-38}
TF5	2.14×10^{2}	6.43×10^{2}	0.659	0.013	4.20×10^{6}	0.392	0.035	0.522	8.627
TF6	0.074	32.410	0.642	0.530	7.66×10^{2}	0.344	0.674	0.133	0.426
TF7	1.148	0.011	8.97×10^{-4}	1.05×10^{-4}	5.016	0.002	0.002	8.61×10^{-4}	1.41×10^{-4}
TF8	2.33×10^{3}	4.09×10^{21}	1.43×10^{3}	6.33×10^{2}	3.36×10^{2}	2.74×10^{3}	6.32×10^{2}	7.81×10^{2}	7.23×10^{2}
TF9	22.100	25.959	4.107	0.000	68.200	0.000	0.000	0.000	0.000
TF10	0.517	0.956	1.56×10^{-12}	0.000	7.642	2.42×10^{-15}	0.000	8.53×10^{-12}	0.000
TF11	0.012	0.507	0.010	0.000	7.894	0.000	0.000	0.000	0.000
TF12	0.131	0.446	0.062	0.115	1.71×10^{7}	0.012	8.91×10^{-47}	0.004	0.012
TF13	0.049	2.839	0.296	2.01×10^{-4}	3.41×10^{7}	0.313	5.57×10^{-48}	0.126	1.781

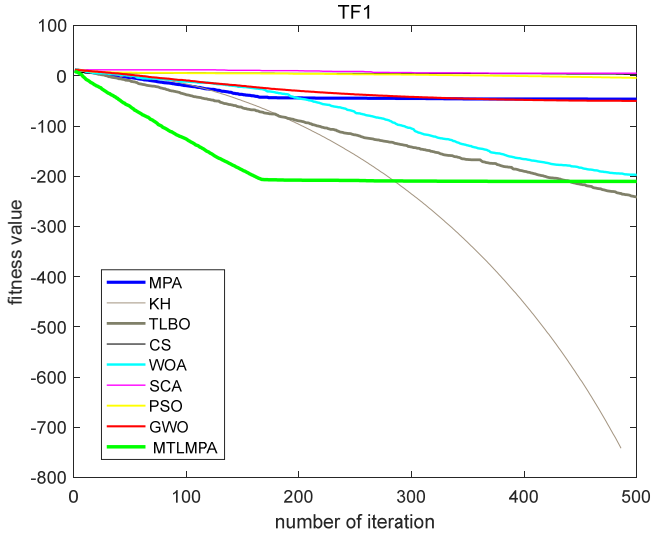

Figure 21. Convergence curves on TF1, dim = 50.

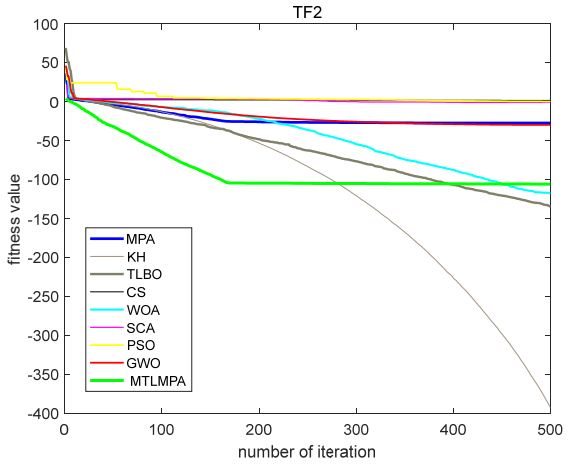

Figure 22. Convergence curves on TF2, dim = 50.

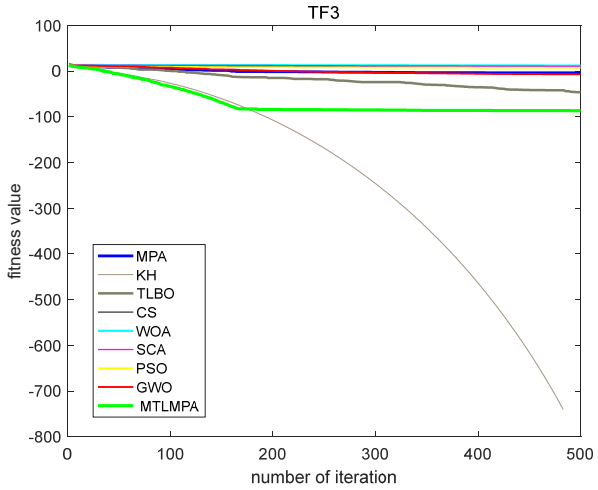

Figure 23. Convergence curves on TF3, dim = 50.

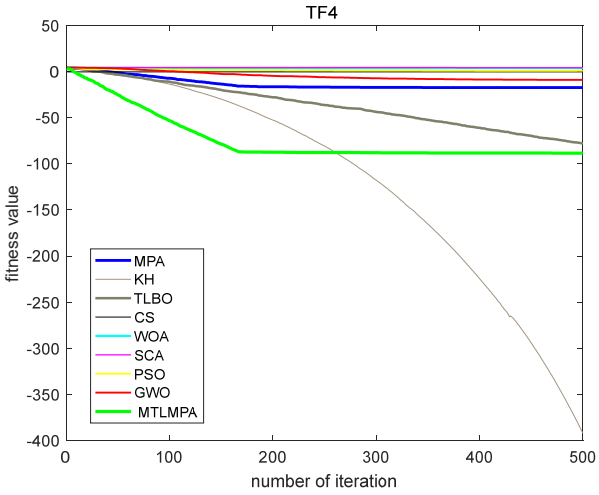

Figure 24. Convergence curves on TF4, dim = 50.

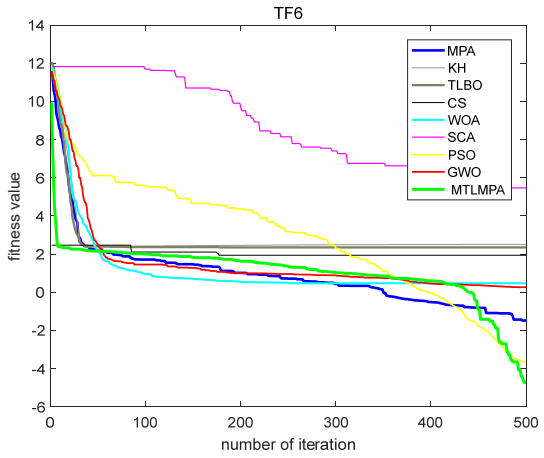

Figure 25. Convergence curves on TF6, dim = 50.

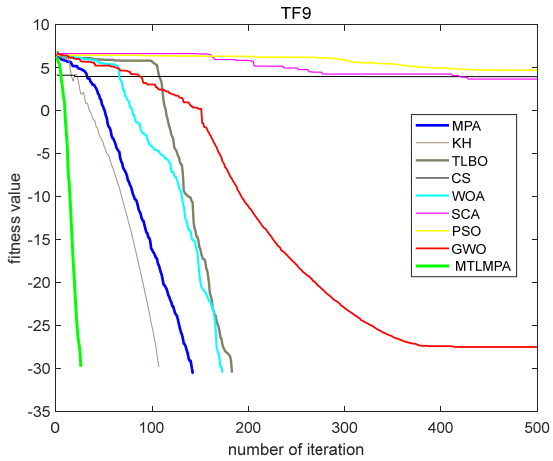

Figure 26. Convergence curves on TF9, dim = 50.

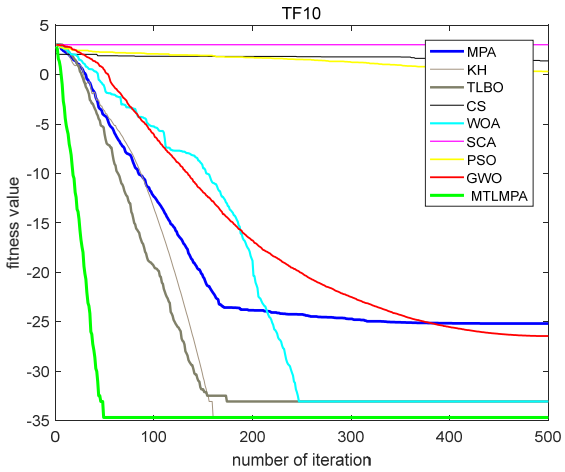

Figure 27. Convergence curves on TF10, dim = 50.

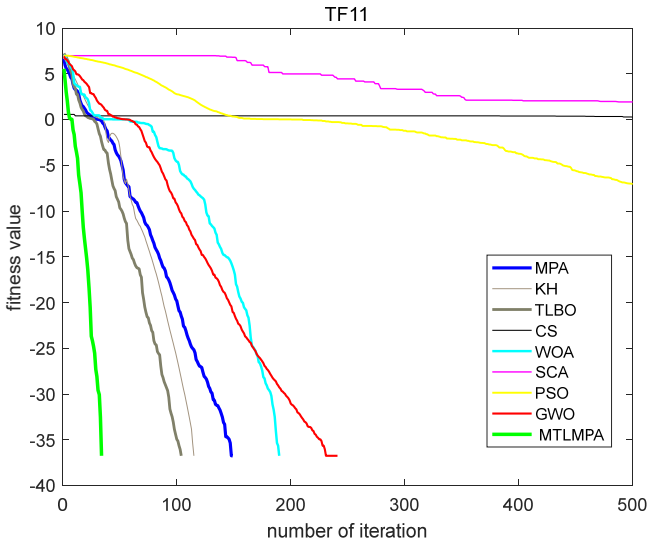

Figure 28. Convergence curves on TF11, dim = 50.

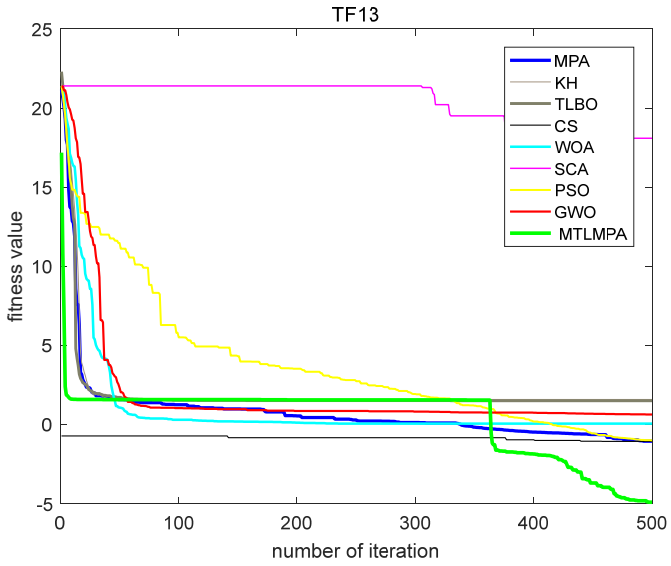

Figure 29. Convergence curves on TF13, dim = 50.

The following Tables 10 and 11 summarize the mean and standard deviation of all the algorithms for 100 dimensions functions. As can be seen from the two tables, MTLMPA algorithm still performs well on TF1, TF2, TF3, TF4, TF7, TF9, TF10, TF11, TF13. And the MTLMPA algorithm is stable and performs well on TF1, TF2, TF3, TF4, TF7, TF9, TF10, TF11, TF12.

Table 10. Mean of function fitness value on TF1–TF13 with 100 dimensions.

Functions	PSO	CS	GWO	KH	SCA	WOA	TLBO	MPA	MTLMPA
TF1	14.783	1.48×10^{2}	5.45×10^{-14}	0.000	9.37×10^{3}	2.04×10^{-78}	6.37×10^{-97}	8.38×10^{-19}	2.35×10^{-85}
TF2	28.100	8.153	5.84×10^{-9}	7.55×10^{-164}	10.500	6.80×10^{-51}	2.48×10^{-58}	3.26×10^{-11}	1.22×10^{-44}
TF3	1.43×10^{4}	1.89×10^{4}	2.32×10^{2}	0.000	2.39×10^{5}	9.13×10^{5}	1.70×10^{-9}	10.900	10.900
TF4	10.413	2.137	0.406	1.84×10^{-166}	88.896	78.251	5.32×10^{-32}	3.00×10^{-7}	5.39×10^{-37}
TF5	1.08×10^{4}	2.18×10^{3}	97.771	98.153	9.96×10^{7}	97.973	98.914	97.188	87.013
TF6	14.729	87.806	8.906	23.426	9.42×10^{3}	2.698	22.635	4.680	4.009
TF7	1.51×10^{3}	2.86×10^{2}	5.75×10^{-3}	1.09×10^{-4}	1.08×10^{2}	2.78×10^{-3}	4.13×10^{-3}	1.51×10^{-3}	2.59×10^{-4}
TF8	-1.18×10^{4}	-4.72×10^{19}	-1.55×10^{4}	-4.71×10^{3}	-6.98×10^{3}	-3.64×10^{4}	-9.33×10^{3}	-2.38×10^{4}	-1.85×10^{4}
TF9	5.61×10^{2}	61.785	7.482	0.000	2.58×10^{2}	0.000	0.000	0.000	0.000
TF10	3.441	2.569	2.65×10^{-8}	8.88×10^{-16}	19.633	4.80×10^{-15}	6.22×10^{-15}	8.67×10^{-11}	8.88×10^{-16}
TF11	0.271	2.450	0.007	0.000	1.04×10^{2}	0.000	3.10×10^{-9}	0.000	0.000
TF12	0.030	0.491	0.223	1.128	2.679×10^{8}	0.029	0.930	0.056	0.057
TF13	43.576	6.913	6.437	9.908	5.34×10^{8}	2.104	9.943	7.054	0.675

Table 11. Std of function fitness value on TF1–TF13 with 100 dimensions.

Functions	PSO	CS	GWO	KH	SCA	WOA	TLBO	MPA	MTLMPA
TF1	5.083	1.61×10^{2}	4.10×10^{-14}	0.000	5.96×10^{3}	8.37×10^{-78}	2.83×10^{-96}	6.71×10^{-19}	1.02×10^{-84}
TF2	5.770	4.506	1.98×10^{-9}	0.000	9.680	3.22×10^{-50}	7.85×10^{-58}	2.83×10^{-11}	2.54×10^{-44}
TF3	3.10×10^{3}	1.76×10^{4}	2.78×10^{2}	0.000	5.55×10^{4}	2.52×10^{5}	3.12×10^{-10}	15.400	1.83×10^{-33}
TF4	1.364	1.982	0.676	0.000	2.928	19.362	1.73×10^{-31}	1.56×10^{-7}	1.51×10^{-36}
TF5	4.21×10^{3}	2.80×10^{3}	0.660	0.014	5.26×10^{7}	0.305	0.040	0.633	28.601
TF6	4.654	83.186	0.863	0.400	6.69×10^{3}	0.660	0.769	1.069	1.536
TF7	2.70×10^{2}	5.33×10^{2}	2.61×10^{2}	8.59×10^{-5}	66.000	3.01×10^{-3}	1.95×10^{-3}	6.82×10^{-4}	1.47×10^{-4}
TF8	3.52×10^{3}	1.04×10^{20}	2.72×10^{3}	8.43×10^{2}	4.87×10^{2}	5.62×10^{3}	7.91×10^{2}	1.13×10^{2}	1.52×10^{3}
TF9	92.517	57.271	6.498	0.000	1.16×10^{2}	0.000	0.000	0.000	0.000
TF10	0.337	1.290	1.28×10^{-8}	0.000	3.343	2.53×10^{-15}	1.81×10^{-15}	3.79×10^{-11}	0.000
TF11	0.075	1.442	0.011	0.000	67.897	0.000	1.70×10^{-8}	0.000	0.000
TF12	1.606	0.690	0.050	0.073	1.11×10^{8}	0.016	0.094	0.013	0.030
TF13	18.505	9.169	0.468	0.002	2.27×10^{8}	0.697	0.127	2.516	2.497

In order to clearly observe the convergence performance of MTLMPA, Figures 30–38 are given. The horizontal axis is the iteration number. And the vertical axis represents the fitness values of testing functions for all optimization algorithms. The green line presents the performance of MTLMPA. Seen from these figures, the convergence speed and convergence accuracy are better than other algorithms on most functions.

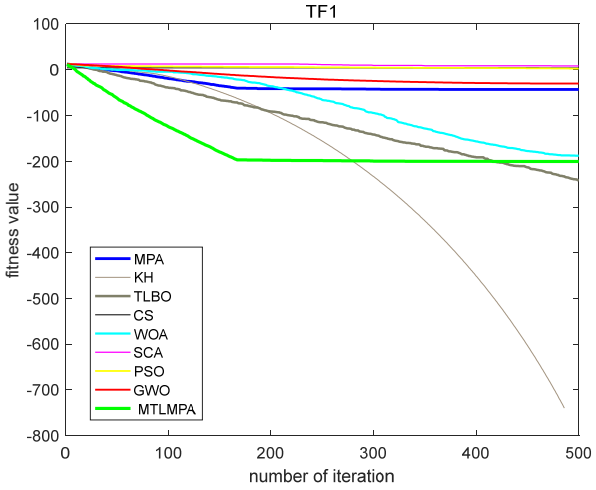

Figure 30. Convergence curves on TF1, dim = 100.

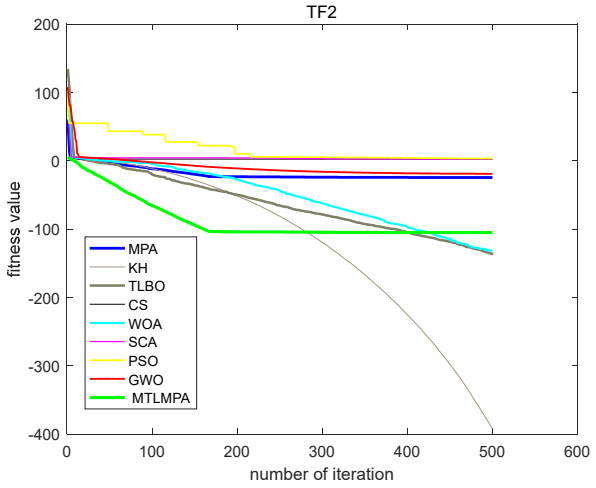

Figure 31. Convergence curves on TF2, dim = 100.

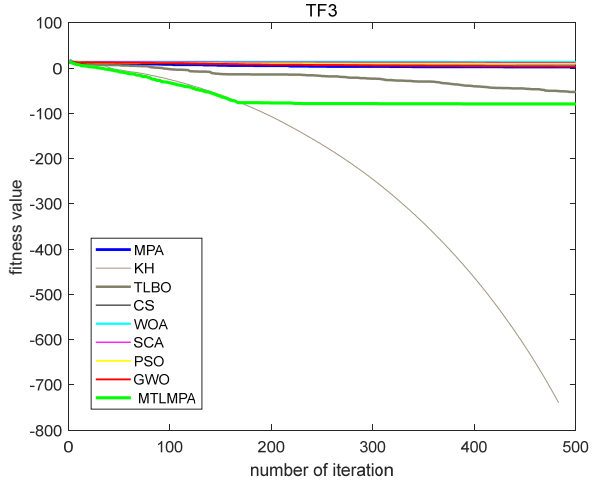

Figure 32. Convergence curves on TF3, dim = 100.

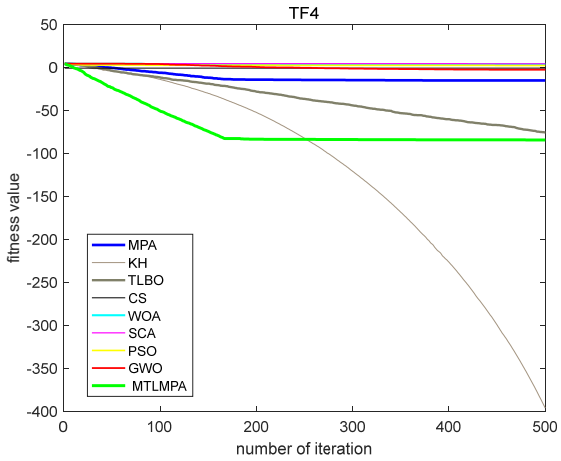

Figure 33. Convergence curves on TF4, dim = 100.

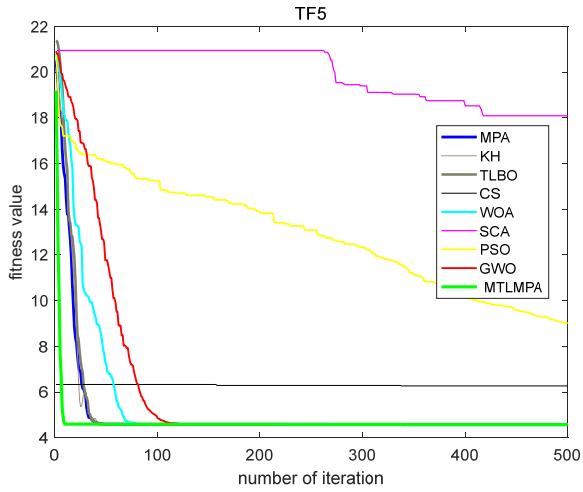

Figure 34. Convergence curves on TF5, dim = 100.

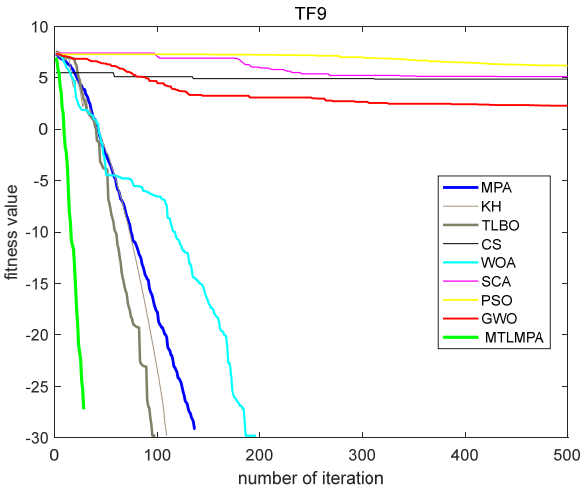

Figure 35. Convergence curves on TF9, dim = 100.

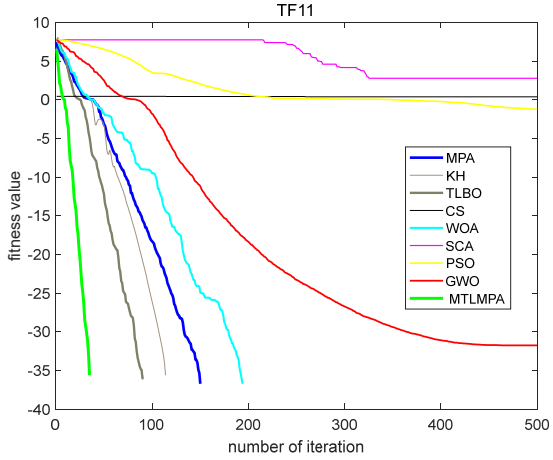

Figure 36. Convergence curves on TF11, dim = 100.

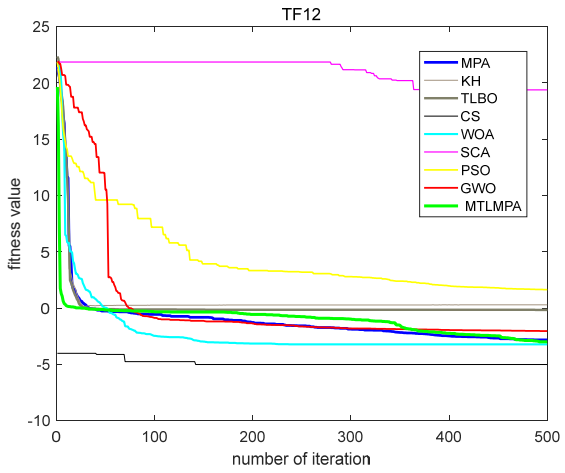

Figure 37. Convergence curves on TF12, dim = 100.

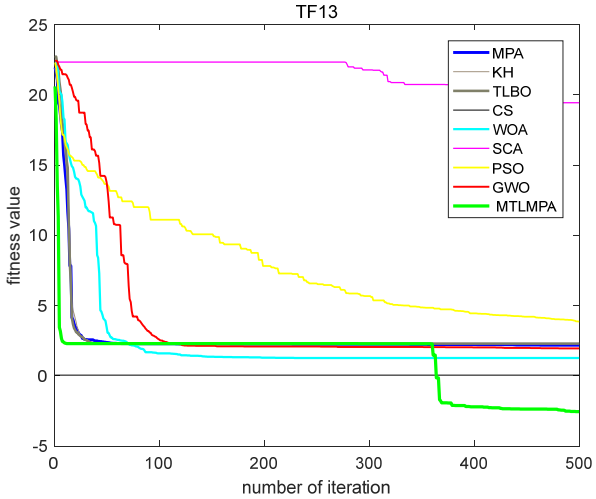

Figure 38. Convergence curves on TF13, dim = 100.

5. Conclusions

To improve the convergence performance and balance the exploitation and exploration ability of MPA, a kind of modified marine predators algorithm hybridized with teaching-learning mechanism (MTLMPA) is proposed. Compared with the conventional MPA, two different population group mechanisms are separately introduced in the first phase and third phase of MTLMPA to update the individuals' position. The new population individual mechanisms can improve the solution quality and balance the exploration and exploitation. To verify the performance of MTLMPA, 23 benchmark testing functions and 29 CEC-2017 functions are used. Experimental results show that the proposed MTLMPA has better convergence performance and stronger stability than other state-of-the-art heuristic optimization algorithms on most functions.

In the future, we will focus on the following tasks:

Based on the MTLMPA, a novel multi-objective MTLMPA will be designed to solve multi-objective optimization problems. The MTLMPA will be further improved to address dynamic constrained optimization problems.

Acknowledgments

Funding: This work is supported by the National Natural Science Foundation of China (Grant No. 62203332) and the Natural Science Foundation of Tianjin (Grant No. 20JCQNJC00430) and The Special fund Project of Tianjin Technology Innovation Guidance (Grand No. 21YDTPJC00370) and the Technical Innovation Guide Foundation of Tianjin (Grant No. 20YDTPJC00320) and College Students' Innovative Entrepreneurial Training Plan Program (Grant Nos. 202010069066, 202110069034, 202110069003).

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95 - International Conference on Neural Networks*, **4** (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
2. A. H. Gandomi, A. H. Alavi, Krill herd: a new bio-inspired optimization algorithm, *Commun. Nonlinear Sci. Numer. Simul.*, **17** (2012), 4831–4845. https://doi.org/10.1016/j.cnsns.2012.05.010
3. A. H. Gandomi, X. Yang, A. H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Eng. Comput.*, **29** (2013), 17–35. https://doi.org/10.1007/s00366-011-0241-y
4. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
5. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching – learning-based optimization: a novel method for constrained mechanical design optimization problems, *Comput.-Aided Des.*, **43 (**2011), 303–315. https://doi.org/10.1016/j.cad.2010.12.015
6. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

7. A. Faramarzi, M. Heidarinejad, S. Mirjalili, A. Gandomi, Marine predators algorithm: a nature-inspired metaheuristic, *Expert Syst. Appl.*, **152** (2020), 113377. https://doi.org/10.1016/j.eswa.2020.113377
8. J. Sarvaiya, D. Singh, Selection of the optimal process parameters in friction stir welding/processing using particle swarm optimization algorithm, *Mater. Today: Proc.*, **62** (2022), 896–901. https://doi.org/10.1016/j.matpr.2022.04.062
9. Z. Hu, H. Norouzi, M. Jiang, S. Dadfar, T. Kashiwagi, Novel hybrid modified krill herd algorithm and fuzzy controller based MPPT to optimally tune the member functions for PV system in the three-phase grid-connected mode, *ISA trans.*, **2022** (2022). https://doi.org/10.1016/j.isatra.2022.02.009
10. Q. Bai, H. Li, The application of hybrid cuckoo search-grey wolf optimization algorithm in optimal parameters identification of solid oxide fuel cell, *Int. J. Hydrogen Energy*, **47** (2022), 6200–6216. https://doi.org/10.1016/j.ijhydene.2021.11.216
11. C. Song, X. Wang, Z. Liu, H. Chen, Evaluation of axis straightness error of shaft and hole parts based on improved grey wolf optimization algorithm, *Measurement*, **188** (2022), 110396. https://doi.org/10.1016/j.measurement.2021.110396
12. H. Abaeifar, H. Barati, A. R. Tavakoli, Inertia-weight local-search-based TLBO algorithm for energy management in isolated micro-grids with renewable resources, *Int. J. Electr. Power Energy Syst.*, **137** (2022), 107877. https://doi.org/10.1016/j.ijepes.2021.107877
13. V. K. Jadoun, G. R. Prashanth, S. S. Joshi, K. Narayanan, H. Malik, F. García Márquez, Optimal fuzzy based economic emission dispatch of combined heat and power units using dynamically controlled Whale Optimization Algorithm, *Appl. Energy*, **315** (2022), 119033. https://doi.org/10.1016/j.apenergy.2022.119033
14. M. Al-qaness, A. Ewees, H. Fan, L. Abualigah, M. Elaziz, Boosted ANFIS model using augmented marine predator algorithm with mutation operators for wind power forecasting, *Appl. Energy*, **314** (2022), 118851. https://doi.org/10.1016/j.apenergy.2022.118851
15. M. Al-qaness, A. Ewees, H. Fan, A. Airassas, M. Elaziz, Modified aquila optimizer for forecasting oil production, *Geo-spatial Inf. Sci.*, **2022** (2022), 1–17. https://doi.org/10.1080/10095020.2022.2068385
16. A. Dahou, M. Al-qaness, M. Elaziz, A. Helmi, Human activity recognition in IoHT applications using Arithmetic Optimization Algorithm and deep learning, *Measurement*, **199** (2022), 111445. https://doi.org/10.1016/j.measurement.2022.111445
17. M. Elaziz, A. Ewees, M. Al-qaness, L. Abualigah, R. Ibrahim, Sine–Cosine-Barnacles Algorithm Optimizer with disruption operator for global optimization and automatic data clustering, *Expert Syst. Appl.*, **207** (2022), 117993. https://doi.org/10.1016/j.eswa.2022.117993
18. X. Chen, X. Qi, Z. Wang, C. Cui, B. Wu, Y. Yang, Fault diagnosis of rolling bearing using marine predators algorithm-based support vector machine and topology learning an d out-of-sample embedding, *Measurement*, **176** (2021), 109116. https://doi.org/10.1016/j.m easurement.2021.109116
19. P. H. Dinh, A novel approach based on three-scale image decomposition and marine predators algorithm for multi-modal medical image fusion, *Biomed. Signal Process. Control*, **67** (2021), 102536. https://doi.org/10.1016/j.bspc.2021.102536
20. M. A. Sobhy, A. Y. Abdelaziz, H. M. Hasanien, M. Ezzat, Marine predators algorithm for load frequency control of modern interconnected power systems including renewable energy sources and energy storage units, *Ain Shams Eng. J.*, **12** (2021), 3843–3857. https://doi.org/10.1016/j.asej.2021.04.031

21. A. Faramarzi, M. Heidarinejad, S. Mirjalili, A. Gandomi, Marine predators algorithm: a nature-inspired metaheuristic, *Expert Syst. Appl.*, **152** (2020), 113377. https://doi.org/10.1016/j.eswa.2020.113377
22. M. A. Elaziz, D. Mohammadi, D. Oliva, K. Salimifard, Quantum marine predators algorithm for addressing multilevel image segmentation, *Appl. Soft Comput.*, **110** (2021), 107598. https://doi.org/10.1016/j.asoc.2021.107598
23. M. Ramezani, D. Bahmanyar, N. Razmjooy, A new improved model of marine predator algorithm for optimization problems, *Arabian J. Sci. Eng.*, **46** (2021), 8803–8826. https://doi.org/10.1007/s13369-021-05688-3
24. M. Abdel-Basset, D. El-Shahat, R. K. Chakrabortty, M. Ryan, Parameter estimation of photovoltaic models using an improved marine predators algorithm, *Energy Convers. Manage.*, **227** (2021), 113491. https://doi.org/10.1016/j.enconman.2020.113491
25. K. Zhong, G. Zhou, W. Deng, Y. Zhou, Q. Luo, MOMPA: multi-objective marine predator algorithm, *Comput. Methods Appl. Mech. Eng.*, **385** (2021), 114029. https://doi.org/10.1016/j.cma.2021.114029
26. R. Sowmya, V. Sankaranarayanan, Optimal vehicle-to-grid and grid-to-vehicle scheduling strategy with uncertainty management using improved marine predator algorithm, *Comput. Electr. Eng.*, **100** (2022), 107949. https://doi.org/10.1016/j.compeleceng.2022.107949
27. E. H. Houssein, I. E. Ibrahim, M. Kharrich, S. Kamel, An improved marine predators algorithm for the optimal design of hybrid renewable energy systems, *Eng. Appl. Artif. Intell.*, **110** (2022), 104722. https://doi.org/10.1016/j.engappai.2022.104722
28. D. Yousri, A. Ousama, Y. Shaker, A. Fathy, T. Babu, H. Rezk, et al., Managing the exchange of energy between microgrid elements based on multi-objective enhanced marine predators algorithm, *Alexandria Eng. J.*, **61** (2022), 8487–8505. https://doi.org/10.1016/j.aej.2022.02.008
29. Y. Ma, X. Zhang, J. Song, L. Chen, A modified teaching–learning-based optimization algorithm for solving optimization problem, *Knowledge-Based Syst.*, **212** (2020), 106599. https://doi.org/10.1016/ j.knosys.2020.106599
30. N. E. Humphries, N. Queiroz, J. Dyer, N. Pade, M. Musyl, K. Schaefer, et al., Environmental context explains Lévy and Brownian movement patterns of marine predators, *Nature*, **465** (2010), 1066–1069. https://doi.org/10.1038/nature09116
31. D. W. Sims, E. J. Southall, N. E. Humphries, G. Hays, C. Bradshaw, J. Pitchford, et al., Scaling laws of marine predator search behaviour, *Nature*, **451** (2008), 1098–1102. https://doi.org/10.1038/nature06518
32. G. M. Viswanathan, E. P. Raposo, M. Luz, Lévy flights and superdiffusion in the context of biological encounters and random searches, *Phys. Life Rev.*, **5** (2008), 133–150. https://doi.org/10.1016/j.plrev.2008.03.002
33. F. Bartumeus, J. Catalan, U. L. Fulco, M. Lyra, G. Viswanathan, Optimizing the encounter rate in biological interactions: Lévy versus Brownian strategies, *Phys. Rev. Lett.*, **88** (2002), 097901. https://doi.org/10.1103/PhysRevLett.88.097901
34. A. Einstein, Investigations on the theory of the brownian movement, *DOVER*, **35** (1956), 318–320. https://doi.org/10.2307/2298685
35. J. D. Filmalter, L. Dagorn, P. D. Cowley, M. Taquet, First descriptions of the behavior of silky sharks, Carcharhinus falciformis, around drifting fish aggregating devices in the Indian Ocean, *Bull. Mar. Sci.*, **87** (2011), 325–337. https://doi.org/10.5343/bms.2010.1057

36. D. Yousri, H. M. Hasanien, A. Fathy, Parameters identification of solid oxide fuel cell for static and dynamic simulation using comprehensive learning dynamic multi-swarm marine predators algorithm, *Energy Convers. Manage.*, **228** (2021), 113692. https://doi.org/10.1016/j.enconman.2020.113692
37. M. Abdel-Basset, R. Mohamed, S. Mirjalili, R. Chakrabortty, M. Ryan, An efficient marine predators algorithm for solving multi-objective optimization problems: analysis and validations, *IEEE Access*, **9** (2021), 42817–42844. https://doi.org/10.1109/ACCESS.2021.3066323
38. T. Niknam, R. Azizipanah-Abarghooee, M. R. Narimani, A new multi objective optimization approach based on TLBO for location of automatic voltage regulators in distribution systems, *Eng. Appl. Artif. Intell.*, **25** (2012), 1577–1588. https://doi.org/10.1016/j.engappai.2012.07.004
39. T. Niknam, F. Golestaneh, M. S. Sadeghi, θ-Multiobjective teaching–learning-based optimization for dynamic economic emission dispatch, *IEEE Syst. J.*, **6** (2012), 341–352. https://doi.org/10.1109/JSYST.2012.2183276
40. R. V. Rao, V. Patel, An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems, *Sci. Iran.*, **20** (2013), 710–720. https://doi.org/10.1016/j.scient.2012.12.005
41. P. K. Roy, S. Bhui, Multi-objective quasi-oppositional teaching learning based optimization for economic emission load dispatch problem, *Int. J. Electr.Power Energy Syst.*, **53** (2013), 937–948. https://doi.org/10.1016/j.ijepes.2013.06.015
42. H. Bouchekara, M. A. Abido, M. Boucherma, Optimal power flow using teaching-learning-based optimization technique, *Electr. Power Syst. Res.*, **114** (2014), 49–59. https://doi.org/10.1016/j.epsr.2014.03.032
43. M. Liu, X. Yao, Y. Li, Hybrid whale optimization algorithm enhanced with Lévy flight and differential evolution for job shop scheduling problems, *Appl. Soft Comput.*, **87** (2020), 105954. https://doi.org/10.1016/j.asoc.2019.105954
44. D. Tansui, A. Thammano, Hybrid nature-inspired optimization algorithm: hydrozoan and sea turtle foraging algorithms for solving continuous optimization problems, *IEEE Access*, **8** (2020), 65780–65800. https://doi.org/10.1109/ACCESS.2020.2984023
45. K. Zhong, Q. Luo, Y. Zhou, M. Jiang, TLMPA: teaching-learning-based marine predators algorithm, *AIMS Math.*, **6** (2021), 1395–1442. https://doi.org/10.3934/math.2021087

AIMS Press