International Journal of Innovative
Computing, Information and Control
Volume **16**, Number **1**, February **2020**

ICIC International ©2020 ISSN 1349-4198
pp. **107–122**

AN IMPROVED WATER EVAPORATION OPTIMIZATION ALGORITHM

Yanjiao Wang and Xiangyang Che*

School of Electrical Engineering
Northeast Electric Power University
No. 169, Changchun Road, Chuanying District, Jilin 132012, P. R. China
wangyanjiao1028@126.com; *Corresponding author: y_9890@126.com

Received June 2019; revised October 2019

Abstract. *The existing water evaporation optimization algorithm has some shortcomings, such as slow convergence speed, and low convergence accuracy. Therefore, this paper proposes an improved water evaporation optimization (IWEO) algorithm. In IWEO, firstly, in monolayer evaporation phase, the construction method of monolayer evaporation probability matrix (MEP) is improved to make up for the defect of slow updating of individuals, and thus speed up the convergence; at the same time, several elite individuals are introduced into the calculation of step size S to enhance the population learning of excellent evolutionary information, which contributes to balancing the global search and local search ability. Secondly, in droplet evaporation phase, the optimal individual and a new step size factor are used to guide and disturb the population respectively, which improves the convergence accuracy of the algorithm while maintaining the diversity of the population. To verify the performance of IWEO, a series of experiments is carried out on 15 benchmark functions. The experimental results show that compared with water evaporation optimization algorithm and the other state-of-the-art algorithms, the proposed algorithm has significant advantages in convergence accuracy and speed.*
Keywords: Water evaporation optimization, Evaporation probability matrix, Step size S, Elite individuals, Optimal individual

1. **Introduction.** In the recent decades, many metaheuristics with different philosophy and characteristics have been developed and play an important role in productive practice. In terms of how they have been inspired, the metaheuristic algorithms can be divided into swarm algorithms [1,2], evolutionary algorithms [3,4] and physical algorithms. Among them, physical algorithms are inspired by a certain physical law or phenomenon. For instance, gravitational search algorithm (GSA) [5] is a physical algorithm that mimics gravitational phenomena; intelligent water drops (IWD) algorithm [6] is imitating the natural flow of water drops; water cycle algorithm (WCA) [7] is inspired from nature and based on the observation of water cycle process and how rivers and streams flow to the sea in the real world; golden ball (GB) algorithm [8] is based on soccer concepts; collision body optimization (CBO) [9] is inspired by the laws of one-dimensional collision. These physical heuristic algorithms have been accepted as most prevalent algorithms.

For developing a new physically based metaheuristic to solve global optimization problems, Kaveh and Bakhshpoori proposed a water evaporation optimization (WEO) algorithm [10] that mimics the evaporation of a tiny amount of water molecules adhered on a solid surface with different wettability. Owing to its simple concept, the WEO algorithm is relatively simple to implement. The experimental results on a series of benchmark

DOI: 10.24507/ijicic.16.01.107

functions show that WEO is able to maintain good diversity of population in the optimization process and is highly competitive with other efficient metaheuristics, such as particle swarm optimization with an aging leader and challengers (ALC-PSO) [11], and particle swarm inspired multi-elitist artificial bee colony algorithm (PS-MEABC) [12]. Moreover, Kaveh and Bakhshpoori employed WEO to resolve three engineering problems: the design of tension-compression spring, the design of welded beam and the design of pressure vessel [10], and the results indicate that the WEO obtains the best design and is competitive with others in the aspect of robustness. In [13], WEO is used to solve a set of six truss design problems from the small to normal scale, and the optimization results demonstrate the efficiency and robustness of the WEO and its competitive performance to other algorithms for continuous structural optimization problems. Saha et al. adopted WEO to handle the optimal power flow (OPF) problem [14], and comparative study with other heuristic algorithms demonstrates competitiveness of WEO in treating varied objectives.

Similar to other new metaheuristic algorithms, the water evaporation optimization algorithm also has some shortcomings. For example, in the monolayer evaporation phase, the constructed monolayer evaporation probability matrix makes the updating rate of individuals too low, which leads to slow convergence in the early stage of evolution; whether in the monolayer evaporation phase or in the droplet evaporation phase, the step size S consisting of two random individuals makes the algorithm attach importance to exploration but ignore exploitation, which reduces the overall convergence accuracy and speed.

In this paper, we present an improved water evaporation optimization (IWEO) algorithm to improve the performance of WEO. Firstly, in the monolayer evaporation phase, this algorithm designs a new construction method of the *MEP* to improve the convergence speed; meanwhile, it introduces the evolutionary information of multiple excellent individual into step size S to guide population evolution, so as to balance the exploration and exploitation ability. Secondly, in view of the need to focus on local search in late evolution, in the droplet evaporation phase, it makes optimal individual guide population evolution to improve the convergence accuracy of the algorithm.

The rest of this paper is organized as follows. Section 2 presents a brief description of WEO. In Section 3, we illustrate the proposed algorithm. Section 4 gives experimental results and analysis. Finally, Section 5 makes a conclusion.

2. Water Evaporation Optimization Algorithm Description.

As we all know, with the decrease of surface wettability, the water aggregation on the surface of solid materials will change from a flat single-layer molecule sheet to a sessile spherical cap, and the corresponding evaporation speed varies greatly under different water aggregation forms. In order to find out how the surface wettability affects the evaporation of the tiny water aggregation, Wang et al. [20] carried out molecular dynamics (MD) simulations on the evaporation of nanoscale water aggregation on a solid substrate with different surface wettability at room temperature. It was found that the evaporation speed of water layer is affected by the interaction energy from the substrate (E_{sub}) and the contact angle (θ) respectively under two different water aggregation forms (high surface wettability and low surface wettability), and the corresponding mathematical models of the evaporation flux which describes the evaporation speed were given.

Inspired by the phenomenon of water evaporation reflected in the simulation carried out by Wang et al. [20], the factors affecting evaporation flux are simulated as individual fitness, the evaporation flux is abstracted as the updating probability of individuals (called evaporation probability matrix) involved in individual evolution process, and in result

the water evaporation optimization (WEO) algorithm was proposed. In view of the fact that different aggregation forms of water molecule correspond to two different models of evaporation flux, WEO accordingly establishes "monolayer evaporation phase" in the early evolution and "droplet evaporation phase" in the late evolution. Next, we take a minimization problem as an example to describe the updating mechanism of individuals.

2.1. **Monolayer evaporation phase.** Firstly, the evaporation flux $J(i)$ of the ith individual can be calculated according to Formula (1).

$$J(i) = \exp\left(E_{sub}(i)\right) \tag{1}$$

where $E_{sub}(i)$ is the corresponding substrate interaction energy of the ith individual. In each iteration, $E_{sub}(i)$ is calculated as follows.

$$E_{sub}(i) = \frac{(E_{\max} - E_{\min}) \times (Fit_i - \mathrm{Min}(Fit))}{(\mathrm{Max}(Fit) - \mathrm{Min}(Fit))} + E_{\min} \tag{2}$$

where Fit_i is the fitness value of ith individual; $\mathrm{Max}(Fit)$ and $\mathrm{Min}(Fit)$ are the maximum and minimum fitness value in the current population; $E_{\max}$ and $E_{\min}$ are -0.5 and -3.5, respectively. The values of the two parameters and the values of $\theta_{\max}$ and $\theta_{\min}$ described below are based on the MD simulation results obtained by Wang et al. [20], and a more detailed introduction to these values can be found in [10] and [13].

Then, on the basis of the individual's evaporation flux, the monolayer evaporation probability matrix (MEP) of the current population is constructed using Formula (3).

$$MEP_{ij} = \begin{cases} 1 & \text{if } rand_{ij} < J(i) \\ 0 & \text{if } rand_{ij} \geq J(i) \end{cases} \tag{3}$$

where $rand_{ij}$ is a random number with uniform distribution in $[0, 1]$; MEP_{ij} is the updating probability of the jth variable of the ith individual.

Finally, the new population $newWM$ is generated according to Formula (4).

$$newWM = WM + S \times MEP \tag{4}$$

where S is a random permutation based step size, and it is calculated as follows.

$$S = rand(0, 1) \times (WM[permute1(i)(j)] - WM[permute2(i)(j)]) \tag{5}$$

where $permute1$ and $permute2$ represent two different random vectors with population size N as the dimensions.

Obviously, the better the original individual is, the smaller the E_{sub} value is, the smaller the evaporation flux $J(i)$ is, thus the more likely the corresponding updating probability of its genes is to be 0, therefore, the more genes it will retain to its offspring.

2.2. **Droplet evaporation phase.** In the droplet evaporation phase, the evaporation flux of an individual is calculated by Formula (6). Then, the evaporation probability matrix and the offspring population are constructed in the same way as the monolayer evaporation phase.

$$J(\theta(i)) = J_0 \times \left(\frac{2}{3} + \frac{\cos^3(\theta(i))}{3} - \cos(\theta(i))\right)^{-2/3} (1 - \cos(\theta(i))) \tag{6}$$

where J_0 is $1/2.6$; $\theta(i)$ is the corresponding contact angle of the ith individual, as shown in Formula (7).

$$\theta(i) = \frac{(\theta_{\max} - \theta_{\min}) \times (Fit_i - \mathrm{Min}(Fit))}{(\mathrm{Max}(Fit) - \mathrm{Min}(Fit))} + \theta_{\min} \tag{7}$$

where Fit_i is the fitness value of ith individual, Max(Fit) and Min(Fit) are the maximum and minimum fitness value in the current population, and corresponding to the MD simulation results, $\theta_{\max}$ and $\theta_{\min}$ are $-20°$ and $-50°$, respectively.

It can be seen from Formulas (6) and (7) that the better the ith individual is, the smaller the corresponding contact angle is, the smaller its evaporation flux is, thereby, the easier the updating probability of its genes is to be 0 in the evaporation probability matrix constructed by Formula (3), so the more genes it retains in its offspring.

To understand its operation process, the flowchart of WEO algorithm is illustrated in Figure 1 and the steps involved are as follows:

Step 1. Initialization. Set the population size to N, the dimension of the problem to d, the maximum number of algorithm iterations to T. Generate the initial population randomly and evaluate individuals;

Step 2. For $t \leq T$, the monolayer evaporation phase is performed. Calculate the evaporation flux of individuals based on Formula (1), construct the monolayer evaporation probability matrix (MEP) using Formula (3), and generate offspring population via Formula (4). If the newly generated individual is better than the current one, the latter should be replaced;

Step 3. Judge whether the number of iterations is greater than $T/2$. If so, proceed to *Step 4*, if not go to *Step 2*;

Step 4. For $t > T$, the droplet evaporation phase is performed. Calculate the evaporation flux based on Formula (6), and also construct the evaporation probability matrix and the offspring population by Formula (3) and Formula (4), respectively. If the newly generated individual is better than the current one, the latter should be replaced;

Step 5. Judge whether the number of iterations becomes larger than T. If so, return the best individual as the output and terminate the algorithm, otherwise go to *Step 4*.

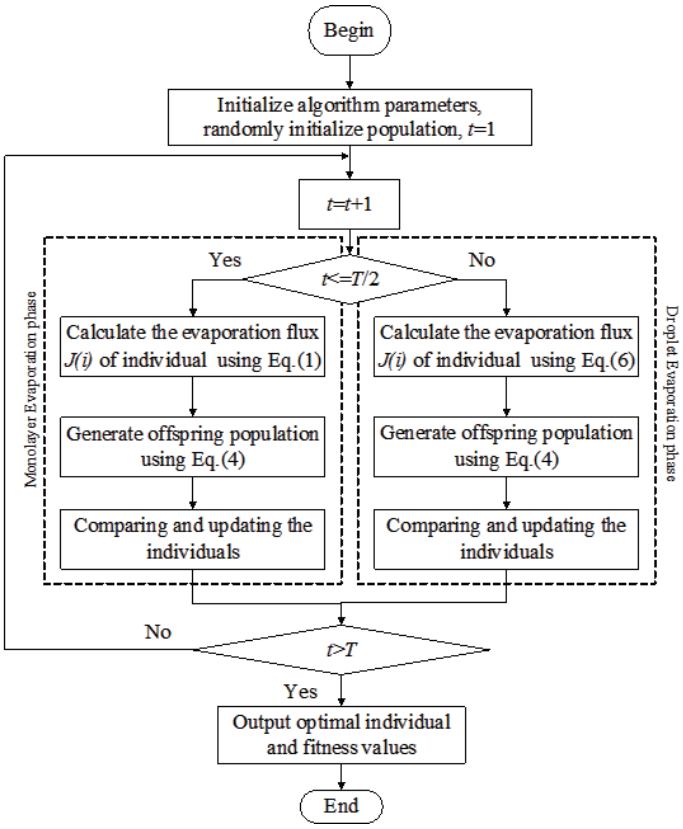


FIGURE 1. WEO algorithm optimization process

3. **Improved Water Evaporation Optimization Algorithm.** In order to improve the WEO algorithm's convergence speed and accuracy, in this paper, the monolayer evaporation phase and the droplet evaporation phase are improved respectively, and an improved water evaporation optimization algorithm is proposed.

3.1. **The improved monolayer evaporation phase.** As shown in Formula (4), the evaporation probability matrix *MEP* determines the retention ratio of the original individuals in the new individuals, that is, the updating rate of individuals. By calculating, the evaporation flux J in Formula (1) is about 0.030 to 0.607, and on the basis of these J values, the *MEP* constructed using Formula (3) results in 39.3-97.0% of the genes of the new individual being directly from the original individual. To study the number of individual's updated genes in each generation, the 30-dimensional Sphere function was tested, and the results are shown in Figure 2 and Figure 3. Among them, Figure 2 shows the average number of updated genes per individual in each iteration when the population size is 50 and the total number of iterations is 1600, and Figures 3(a)-3(c) show the distribution between the number of updated genes and the number of the corresponding individuals when the iterations are 1, 400 and 800, respectively.

It can be seen from Figure 2 that in the early iterations (the first 800 iterations), i.e., the monolayer evaporation phase, about 4 genes per individual are updated in each iteration, accounting for 13.3% of an individual's total genes. As is seen from Figure 3, individuals with less than 10 updated genes take up a large proportion of the population. These results clearly indicate that the updating rate of individuals at this stage, is too low and makes the algorithm inefficient to a large extent. In the second half of the whole iteration shown in Figure 2, by contrast, each individual updates about 20 genes in each iteration, which shows that on the basis of the J value of about 0.6-1 in Formula (6), the evaporation probability matrix constructed by Formula (3) can make the overall updating rate of individuals in the droplet evaporation phase reach a satisfactory level. To sum up, in the monolayer evaporation phase, the *MEP* constructed by Formula (3) causes slow individual change and insufficient convergence speed.

To promote the individual updates and eventually speed up the convergence, a new *MEP* construction method that can lift the overall updating rate of individuals is needed.

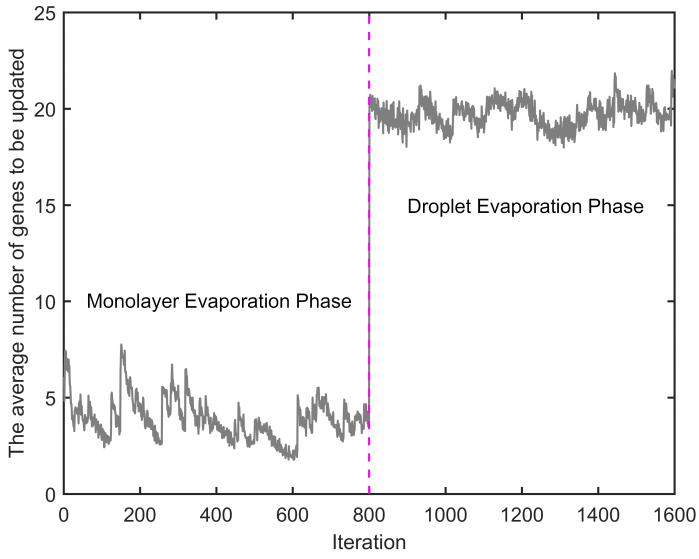


FIGURE 2. The average number of updated genes in each iteration

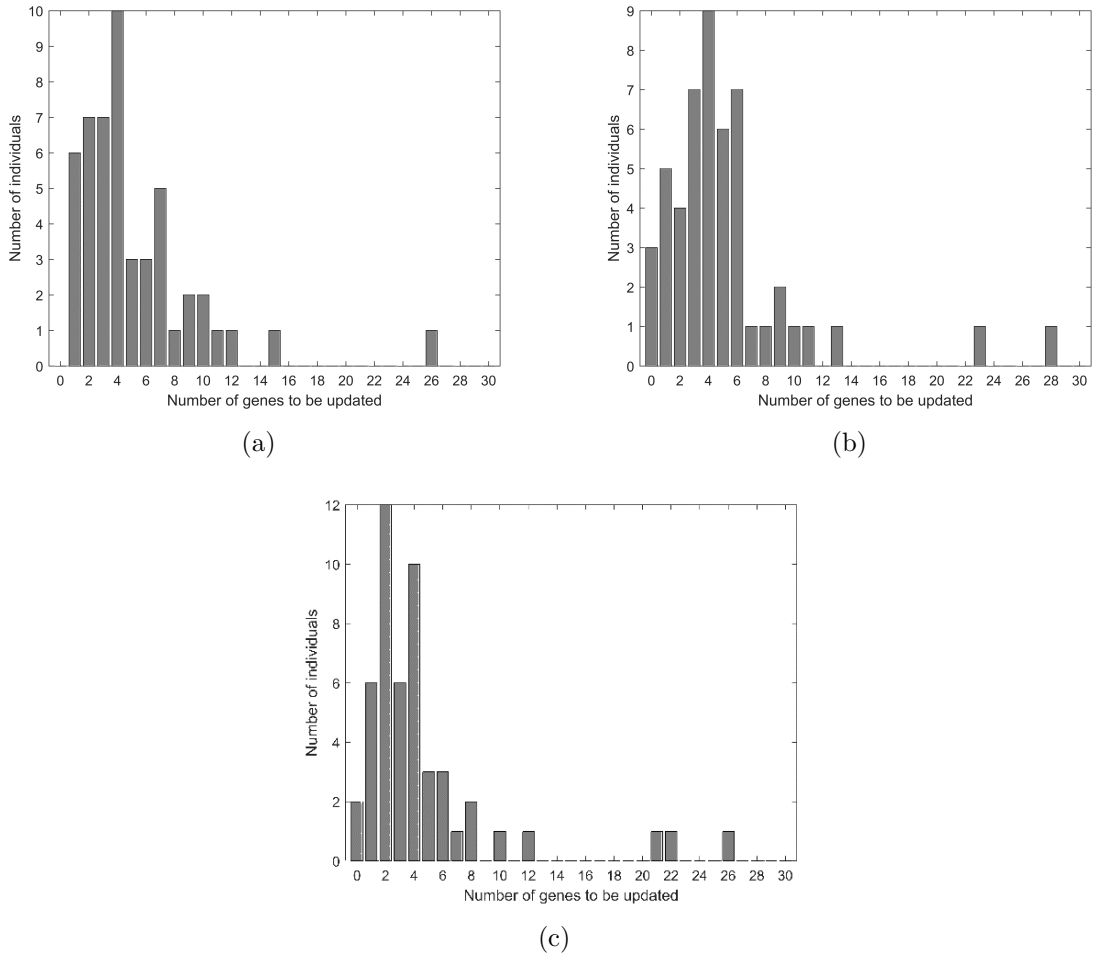


FIGURE 3. (a), (b) and (c) show the distribution between the number of updated genes and the number of the corresponding individuals when the iterations are 1, 400 and 800, respectively.

To this end, and based on the characteristics of early evolution of the algorithm, this paper improves the construction of *MEP* in the monolayer evaporation phase, as shown in Formula (8). To compare the convergence speed of the algorithm before and after the improvement, Figure 4 shows a comparison of the convergence curves.

$$MEP_{ij} = \begin{cases} 1 - \cos(0.5 \times rand) & \text{if } rand_{ij} < J(i) \\ \cos(0.5 \times rand) & \text{if } rand_{ij} \geq J(i) \end{cases} \tag{8}$$

Compared with Formula (3), the *MEP* constructed by Formula (8) converts approximately the ratio of the S in every new individual, so that the better an individual is, the higher its updating rate is. Therefore, in the early stage of evolution, better individuals strengthen the communication with other individuals, while poor individuals try to retain their own information. At the same time, the new *MEP* adjusts the gene updating probability from 0 or 1 to a random number in the interval 0-0.1224 or 0.8876-1, so that every gene of an individual can participate in the updating process of the individual. From Figure 4, it can be seen that the convergence speed has been significantly improved after the improvement.

In addition, as is seen from Formula (5), the step size S consists of two random individuals, from which the original individuals learn to generate new individuals. Evidently, the

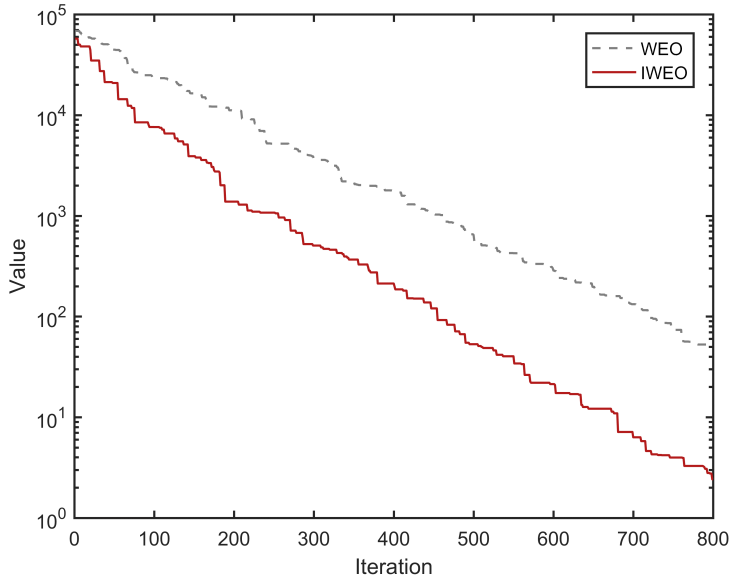


FIGURE 4. The convergence curves of WEO and IWEO

step size S can make population keep good diversity, but the search for new individuals is bound to be blind and this is at the cost of reducing the overall convergence speed and accuracy. To further improve the performance of the monolayer evaporation phase, in view of the fact that excellent individuals often carry representative evolutionary information, by this, a new calculation method of step size S is proposed as follows.

$$S_{i,j} = r_{i,j} \times (WM_{e,j} - WM_{i,j}) + \phi_{i,j} \times (WM_{i,j} - WM_{k1,j}) \tag{9}$$

where, WM_e is a randomly selected elite individual from the first $\lceil p \times N \rceil$ individuals in the population sorted in ascending order of fitness values, $p \in (0, 1)$, $r_{i,j}$ is the random variable in the range of $[0, 1]$, $\phi_{i,j}$ is the random variable in the range of $[-1, 1]$, $k1 \in \{1, 2, \ldots, N\}$, and $e \neq k1 \neq i$.

In Formula (9), the new method enhances the population to learn elite individuals that carry fine evolutionary information, which will inevitably promote the population to approach the optimal solution rapidly. It is worth noting that the elite individual is not just locally optimal individual, but selected randomly from several excellent individuals, which can still maintain the diversity of the population and reduce the risk of falling into the local optimum.

To sum up, through the combination of Equation (8) and Equation (9), the updating rate of individuals can be lifted, and the updating rate of excellent individuals is higher than that of poor individuals. The excellent individuals gradually evolve toward the global optimal solution under the guidance of several elite individuals, thereby improving the convergence speed. Meanwhile, the poor individuals retain more of their own information to explore potential excellent areas, which helps the algorithm to maintain population diversity and avoid falling into local optimum. That is to say, the method proposed in this paper can balance the contradiction between exploration and exploitation to a certain extent.

3.2. **The improved droplet evaporation phase.** As can be seen from Section 2.2, the individual updating method adopted in this phase is the same as that in the monolayer evaporation phase, so the random individuals in the step size S will also inevitably affect the convergence speed and accuracy of this phase.

Given that droplet evaporation phase is at the later stage of the overall evolution, individuals within the population have tended to be excellent. Therefore, on the premise of avoiding falling into local optimum, fine search should be carried out in the region where the optimal solution is located, so as to improve the convergence accuracy. Based on the above considerations, this paper designs a new calculation method of step size S as follows.

$$S_{i,j} = r_{i,j} \times (WM_{best,j} - WM_{i,j}) + \phi_{i,j} \times (WM_{k2,j} - WM_{k3,j}) \tag{10}$$

where, WM_{best} is the optimal individual in the current population, $r_{i,j}$ and $\phi_{i,j}$ are the random variables in the range of $[0, 1]$ and $[-1, 1]$ respectively, $k2$, $k3 \in \{1, 2, \ldots, N\}$, and $k2 \neq k3 \neq i$.

Compared with Formula (5), the proposed calculation method of step size S as shown in Formula (10) has the following advantages. Firstly, the introduction of the optimal solution can strengthen the fine search around it and avoid missing the global optimal solution due to too random search, which helps the algorithm to improve convergence accuracy. Secondly, the step size factor in front of the difference vector consisting of two random individuals changes from *rand* to a random variable in the range of $[-1, 1]$, which increases the search direction to avoid the difficulty of convergence caused by one-way search, and also increases the random disturbance to a certain extent to maintain the population diversity.

3.3. **IWEO algorithm flow.** The specific implementation steps of the improved water evaporation optimization algorithm are as follows:

Step 1. Set the population size to N, the dimension to d, the proportion of elite individuals to p, the maximum number of iterations to T;

Step 2. Generate initial population using random method and evaluate these individuals based on the objective function of the problem at hand;

Step 3. Perform the improved monolayer evaporation phase in Section 3.1 to generate new population. Compare the newly generated individual and the current one, and retain the better of the two;

Step 4. Judge whether the number of iterations is greater than $T/2$, if so, proceed to *Step 5*, if not go to *Step 3*;

Step 5. Perform the improved droplet evaporation phase in Section 3.2 to generate new population. Compare the new individual and the current one, and retain the better of the two;

Step 6. Judge whether the number of iterations is larger than T, if so, the best individual is output and the algorithm terminates, if not go to *Step 5*.

3.4. **Discussions.** In this section, the differences of the IWEO algorithm from the classic WEO algorithm are discussed as follows.

As can be found in [10], the WEO uses the method shown as Formula (3) to construct the evaporation probability matrix *MEP* in both the monolayer evaporation phase and the droplet evaporation phase. However, by analyzing the influence of the evaporation probability matrix on the updating rate of individuals, this paper improves the construction method of the *MEP* in the monolayer evaporation phase, so the IWEO employs Formulas (8) and (3) to construct the *MEP* in these two phases, respectively.

Moreover, in [10], the WEO uses Formula (5) to calculate the step size S in the two evaporation phases. However, by analyzing the defects of Formula (5) and based on the characteristics of the early and late evolution of the algorithm, in this paper, the IWEO employs Formulas (9) and (10) to calculate step size S so as to improve the convergence speed and accuracy in both phases.

4. **Experiment.** In order to verify the performance of the proposed algorithm, 5 low-dimensional benchmark functions and 10 high-dimensional functions are used. Among them, f_4-f_{11} are unimodal functions, and Functions f_1-f_3 and f_{12}-f_{15} are multimodal functions where the number of their local minimum increases exponentially with the problem dimension.

To make experiments more reasonable and convincing, we compare IWEO with WEO and other three representative state-of-the-art algorithms including all-dimension neighborhood based particle swarm optimization with randomly selected neighbors (ADN-RSN-PSO) [16], modified differential evolution with self-adaptive parameters method (MDE) [17] and an enhanced artificial bee colony algorithm with adaptive differential operators (ABCADE) [18] in terms of convergence accuracy and convergence speed. For the sake of fairness, the population size of each algorithm is 50. The detailed parameter settings of all algorithms are shown in Table 1. Among them, the setting of the parameter p, added in IWEO relative to WEO, is based on the results of a large number of experiments, and the parameters of other algorithms are set according to the corresponding original works.

Table 1. Parameter setting of various algorithms

Algorithm	Parameter setting
ADN-RSN-PSO	$w = 0.7298$, $c_1 = c_2 = 2.05$
MDE	$CR = 0.4$, F is a random number in the range of $[0, 1]$
ABCADE	limit $= 200$, $m = 5$, $n = 10$, $c_1 = 0.9$, $c_2 = 0.999$
WEO	$E_{\max} = -0.5$, $E_{\min} = -3.5$, $J_0 = 1/2.6$, $\theta_{\min} = -50°$, $\theta_{\max} = -20°$
IWEO	$p = 0.3$, the other parameters are same as those in WEO

4.1. **Comparison of convergence accuracy.** The convergence accuracy of IWEO is examined in this section. The compared algorithms are tested on the low-dimensional functions f_1-f_5 and the high-dimensional functions f_6-f_{15}, and the max numbers of function evaluation on low-dimensional and high-dimensional functions are 8000 and 80000 respectively. To avoid the adverse effect of randomness in a single run, in this paper, each algorithm runs 30 times independently, and the minimum, mean value, maximum and standard deviation denoted as 'Best', 'Mean', 'Worst' and 'SD' of the objective function values gained are used to evaluate the convergence accuracy of the algorithm. In the experiment, the dimension of f_1-f_5 is $d = 2$, and the dimensions of f_6-f_{15} is $d = 30$. The specific results are shown in Table 2 and Table 3 respectively. For the sake of clarity, the best mean values are highlighted in boldface.

As shown in Table 2, for the low-dimensional benchmark functions with $d = 2$, IWEO obtains the best results on all functions except f_5, on which ABCADE performs the best. IWEO is able to find the global optimal solutions in all 30 runs on f_2, f_3, and on the remainder functions (i.e., f_1, f_4, f_5), the solutions gained by IWEO are very close to their global optima. ABCADE has the same performance as IWEO on f_2 and is slightly better on f_5, but it is inferior to IWEO on other functions. The other three algorithms fail to converge to their global optimum solutions on these low-dimensional functions, and among them, WEO and ADN-RSN-PSO have the same and worst performance.

As shown in Table 3, the results of IWEO on all high-dimensional functions at $d = 30$ are obviously better than those of other algorithms. IWEO can get the corresponding global optimal solutions in all 30 runs on f_9, f_{11} and f_{12}, and obtain the results very close to the theoretical optimal solutions on remaining functions. ABCADE has the same performance as IWEO on f_{12}; however, it is distinctly second best on other functions.

TABLE 2. The results on low-dimensional functions at $d = 2$

Fun	Algorithm	Best	Mean	Worst	SD
f_1 Booth	ADN-RSN-PSO	1.1924e-11	5.7754e-07	7.2548e-06	1.7350e-06
	MDE	3.6544e-16	1.6210e-12	1.6322e-11	3.1536e-12
	ABCADE	1.2469e-18	2.8147e-13	4.3089e-12	1.0415e-12
	WEO	1.1148e-08	5.8862e-07	2.7863e-06	7.3003e-07
	IWEO	0	**1.6224e-29**	8.5986e-29	2.1446e-29
f_2 Boachevsky1	ADN-RSN-PSO	3.8969e-13	0.0017	0.0257	0.0060
	MDE	5.5511e-16	1.3887e-13	1.3506e-12	2.6519e-13
	ABCADE	0	**0**	0	0
	WEO	1.5387e-06	1.0250e-04	6.3772e-04	1.4400e-04
	IWEO	0	**0**	0	0
f_3 Boachevsky3	ADN-RSN-PSO	5.5511e-17	1.4117e-04	0.0012	3.7794e-04
	MDE	4.1419e-12	3.5038e-10	1.3639e-09	4.4528e-10
	ABCADE	0	5.3522e-14	1.5505e-12	2.8281e-13
	WEO	1.8522e-06	1.6298e-04	6.3267e-04	1.8372e-04
	IWEO	0	**0**	0	0
f_4 Matyas	ADN-RSN-PSO	8.4527e-11	8.3507e-06	1.1634e-04	2.3599e-05
	MDE	8.3692e-15	2.7931e-12	4.7937e-11	9.1621e-12
	ABCADE	7.2256e-17	1.0487e-13	1.1123e-12	2.8395e-13
	WEO	1.1831e-09	5.2270e-08	3.1339e-07	6.4877e-08
	IWEO	5.0688e-25	**2.3577e-22**	2.6311e-20	4.7873e-21
f_5 Easom	ADN-RSN-PSO	-0.7323	-0.0919	-0.0100	0.2229
	MDE	-1.0000	-1.0000	-1.0000	4.9241e-06
	ABCADE	-1	**-1**	-1	0
	WEO	-1.0000	-0.9332	0	0.2537
	IWEO	-1	-1.0000	-1.0000	1.8968e-15

In addition, from Table 4, MDE ranks third, while ADN-RSN-PSO and WEO have poor results.

In summary, when the dimension is same, IWEO can obtain better objective function values than other algorithms, and at the same time, it can be seen that the standard deviation of IWEO is the smallest on all functions except f_5. So, the IWEO proposed in this paper is superior to the other four algorithms in terms of accuracy and stability.

To rank and compare AND-RSN-PSO, MDE, ABCADE, WEO and IWEO rationally, the nonparametric tests of Friedman and Wilcoxon are performed on SPSS19.0. The average rankings of all algorithms over all problems based on Friedman test and the P value of the other four algorithm versus IWEO based Wilcoxon test are provided in Table 4.

As shown in Table 4, the average rankings demonstrate that the results of IWEO are better than those of other algorithms whether on low-dimensional functions at $d = 2$ or on high-dimensional functions at $d = 30$. The P value of AND-RSN-PSO and WEO vs. IWEO based Wilcoxon test is less than 0.05 on functions at $d = 2$ and that of AND-RSN-PSO, MDE and WEO vs. IWEO is less than 0.05 on functions at $d = 30$, which indicate there are significant differences in the performance between these algorithms and IWEO, and the advantage of IWEO on functions at $d = 30$ is more obvious.

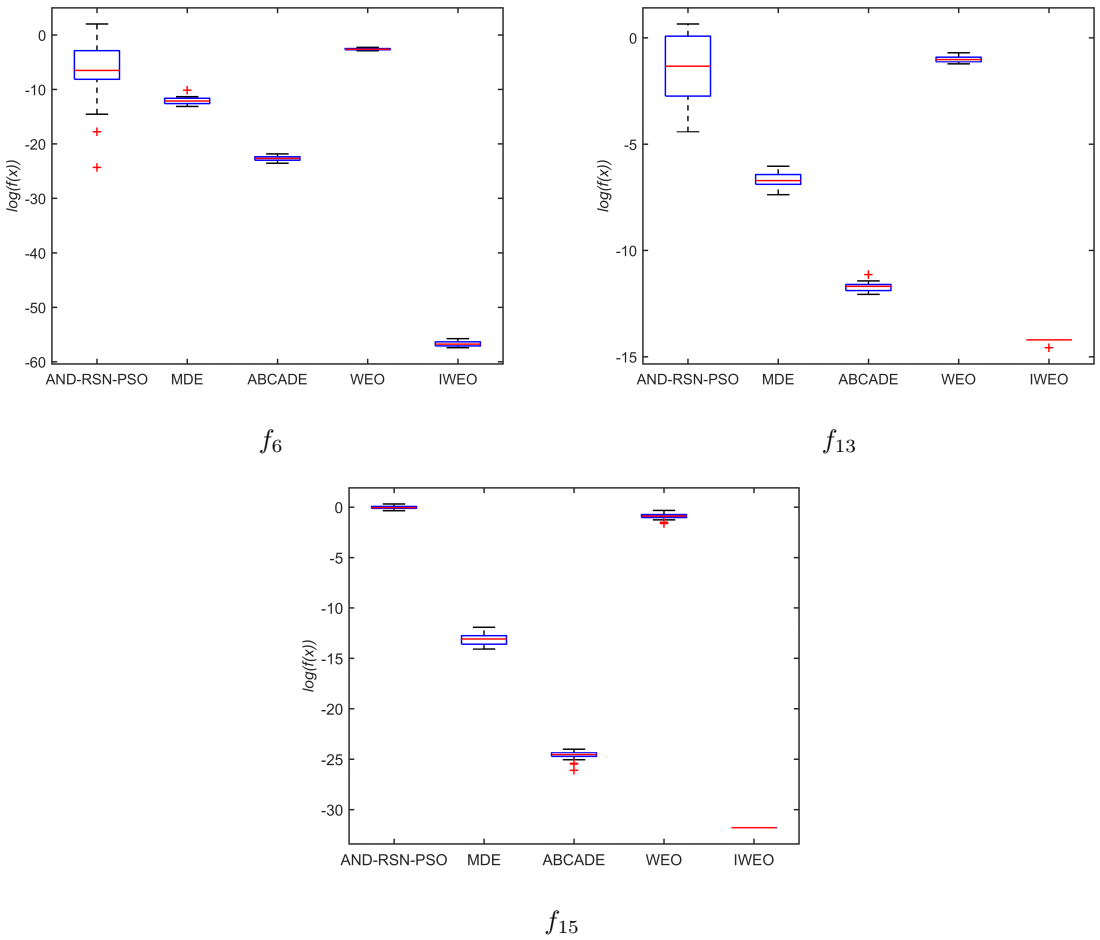
In order to comprehensively examine the effects of 30 experiments, the box plots of the experimental results on 3 representative functions at $d = 30$ (unimodal functions f_6 and multimodal functions f_{13} and f_{15}) are shown in Figure 5. As shown in Figure 5, IWEO can

TABLE 3. The results on high-dimensional functions at $d = 30$

Fun	Algorithm	Best	Mean	Worst	SD
f_6 Sphere	AND-RSN-PSO	4.6793e-25	3.3823	101.1303	18.4617
	MDE	7.2891e-14	3.7216e-12	7.1280e-11	1.2840e-11
	ABCADE	5.8382e-24	1.9269e-24	5.7204e-23	1.2758e-23
	WEO	0.0021	0.0058	0.0160	0.0030
	IWEO	3.7490e-58	**3.3990e-57**	1.7277e-56	3.9317e-57
f_7 Schwefel 2.22	AND-RSN-PSO	8.7478e-06	0.4108	3.1626	0.9135
	MDE	2.8967e-08	1.3162e-07	3.2238e-07	7.0975e-08
	ABCADE	3.0547e-14	4.9706e-14	1.0875e-13	1.8525e-14
	WEO	0.1226	0.1885	0.2988	0.0411
	IWEO	6.5136e-31	**2.5766e-30**	8.3971e-30	1.74333e-30
f_8 SumSquares	AND-RSN-PSO	4.1468e-22	0.0208	0.4985	0.00907
	MDE	5.2102e-15	2.3058e-13	9.0565e-13	2.3339e-13
	ABCADE	3.5349e-25	3.0806e-24	1.0385e-23	2.6982e-24
	WEO	3.4881e-04	8.0976e-04	0.0022	3.6903e-04
	IWEO	2.4546e-59	**3.8908e-58**	1.6867e-57	4.1048e-58
f_9 Dixon-Price	AND-RSN-PSO	0.9721	9.9948	1.0973	0.0213
	MDE	0.6667	0.6667	0.6679	2.2355e-04
	ABCADE	1.8206e-21	1.0812e-16	1.6100e-15	4.1549e-16
	WEO	6.4153	10.8001	13.8718	2.0209
	IWEO	0	**0**	0	0
f_{10} Elliptic	AND-RSN-PSO	1.3462e-20	1.1200e+03	2.3478e+04	4.4054e+03
	MDE	3.3143e-10	9.7263e-09	5.9207e-08	1.1422e-08
	ABCADE	3.8165e-20	2.4725e-19	1.0076e-18	2.7333e-19
	WEO	2.8246	4.5078	7.5078	1.4468
	IWEO	2.2644e-55	**6.7861e-54**	4.2111e-53	1.0009e-53
f_{11} Step	AND-RSN-PSO	4.6739	6.0181	7.4985	0.7745
	MDE	3.4733e-16	6.4200e-15	6.7099e-14	1.5459e-14
	ABCADE	2.2676e-27	1.0702e-25	5.2982e-25	1.4926e-25
	WEO	8.1592e-06	1.3718e-05	2.3441e-05	4.7091e-06
	IWEO	0	**0**	0	0
f_{12} Griwank	AND-RSN-PSO	37.7860	59.2189	72.1914	9.4495
	MDE	1.5099e-13	1.1451e-12	3.1450e-12	8.5340e-13
	ABCADE	0	**0**	0	0
	WEO	0.0137	0.0637	0.1616	0.0344
	IWEO	0	**0**	0	0
f_{13} Ackley	AND-RSN-PSO	3.7345e-05	0.8576	4.4305	1.3895
	MDE	4.1776e-08	2.8015e-07	9.0653e-07	2.2637e-07
	ABCADE	5.6044e-13	2.0362e-12	3.8716e-12	9.2668e-13
	WEO	0.0907	0.1595	0.3272	0.0492
	IWEO	2.6645e-15	**6.0988e-15**	6.2172e-15	6.4863e-16
f_{14} Generalized Penalized1	AND-RSN-PSO	2.6390	3.2565	5.7843	0.7224
	MDE	1.4003e-13	8.4250e-13	4.0953e-12	1.1335e-12
	ABCADE	7.8323e-22	4.9456e-20	2.8290e-19	9.1765e-20
	WEO	704.2388	2.5554e+05	2.9694e+06	5.8978e+05
	IWEO	1.3498e-32	**1.6203e-24**	2.4305e-23	6.2756e-24
f_{15} Generalized Penalized2	AND-RSN-PSO	0.3568	0.9965	1.6534	0.3092
	MDE	8.2995e-15	1.4551e-13	1.1940e-12	2.2185e-13
	ABCADE	1.9987e-26	2.2192e-25	8.5732e-25	2.3177e-25
	WEO	0.0248	0.1464	0.3281	0.0843
	IWEO	1.5705e-32	**1.5705e-32**	1.5705e-32	5.5674e-48

TABLE 4. The results of nonparametric test for all functions

Algorithms	$d = 2$		$d = 30$	
	Ave-Ranking	P value	Ave-Ranking	P value
AND-RSN-PSO	4.60	0.043	4.70	0.005
MDE	2.80	0.068	3.00	0.005
ABCADE	1.90	0.109	1.60	0.180
WEO	4.40	0.043	4.30	0.005
IWEO	1.30		1.40	



f_6 f_{13} f_{15}

FIGURE 5. The box plot of results of all algorithms

obtain better and more stable solutions than other compared algorithms, which further illustrates the conclusions reflected in Table 3 and Table 4.

4.2. **Comparison of convergence speed.** This section examines and compares the convergence speed of IWEO. In all experiments, each algorithm is independently run on f_1-f_5 at $d = 2$ and f_6-f_{15} at $d = 30$ for 30 times, and the mean values obtained under the same evaluation times are used to evaluate the convergence speed of algorithm. The max numbers of function evaluation on low-dimensional functions and high-dimensional functions are 4000, 5000, 6000 and 40000, 50000, 60000 respectively. The specific test results are shown in Table 5.

As shown in Table 5, under the same number of function evaluation, the convergence accuracy of IWEO is significantly better than the other four algorithms on all functions

TABLE 5. The results of convergence speed

Fun	Number of FES	Algorithm				
		AND-RSN-PSO	MDE	ABCADE	WEO	IWEO
f_1 Booth	4000	3.6516e-04	4.1615e-08	6.3143e-09	4.8915e-04	**3.2624e-15**
	5000	2.0739e-05	1.2651e-09	1.1780e-09	1.0030e-04	**8.6721e-19**
	6000	3.0229e-06	1.8179e-11	1.3450e-12	1.4076e-05	**3.1448e-22**
f_2 Boachevsky1	4000	1.7910e-04	2.5375e-07	**0**	0.0690	2.3315e-16
	5000	5.7926e-05	5.1894e-09	**0**	0.0153	**0**
	6000	5.1984e-05	2.2602e-11	**0**	0.0039	**0**
f_3 Boachevsky3	4000	0.0052	1.9593e-06	3.1245e-08	0.0296	**5.4870e-10**
	5000	1.7245e-04	2.5842e-08	3.2638e-10	0.0088	**2.2946e-12**
	6000	3.6464e-06	1.0075e-09	9.9269e-12	0.0027	**6.0248e-15**
f_4 Matyas	4000	4.7634e-04	6.5393e-07	3.3328e-08	3.3470e-04	**5.0550e-12**
	5000	2.6537e-04	1.9918e-08	1.3969e-09	1.4294e-05	**1.1273e-14**
	6000	7.9422e-05	2.2803e-09	2.1318e-10	3.4473e-06	**5.7044e-17**
f_5 Easom	4000	−0.7201	−0.8075	**−1.0000**	−0.7447	**−1.0000**
	5000	−0.9789	−0.9862	**−1.0000**	−0.7888	**−1.0000**
	6000	−0.9953	−0.9985	**−1.0000**	−0.8300	**−1.0000**
f_6 Sphere	40000	8.7910e-15	6.3039e-05	2.7411e-06	6.7014	**1.1741e-26**
	50000	2.2162e-20	1.1934e-07	7.0838e-13	0.9278	**2.3528e-34**
	60000	1.2687e-20	9.5258e-10	2.3411e-16	0.1348	**6.4301e-42**
f_7 Schwefel 2.22	40000	7.4694e-06	0.0011	3.7236e-05	2.5643	**1.5191e-14**
	50000	4.0388e-06	7.3906e-05	2.7235e-08	1.1349	**2.0386e-18**
	60000	1.8820e-08	7.4927e-06	3.2807e-10	0.4735	**2.0435e-22**
f_8 SumSquares	40000	2.8169e-22	4.5110e-06	2.2493e-10	0.7650	**1.1658e-27**
	50000	7.3131e-25	3.8994e-08	1.0281e-13	0.1175	**3.3824e-35**
	60000	2.8179e-32	2.8285e-10	3.4481e-17	0.0156	**1.0829e-42**
f_9 Dixon-Price	40000	33.9773	0.7363	**3.3049e-04**	70.9354	0.5781
	50000	1.3764	0.6757	**5.3402e-07**	46.2834	1.2915e-04
	60000	1.0354	0.6686	9.5710e-10	27.5826	**1.8549e-12**
f_{10} Elliptic	40000	3.7334e+05	0.2954	7.9326e-06	9.6789e+03	**1.2509e-23**
	50000	5.7240e+05	8.9740e-04	2.6456e-09	1.4583e+03	**3.7458e-31**
	60000	2.8372e+04	6.5050e-06	1.3904e-12	240.9365	**7.0831e-39**
f_{11} Step	40000	4.7607	4.0734e-08	5.5505e-12	0.0165	**3.8094e-29**
	50000	4.1544	3.9701e-10	1.8890e-15	0.0024	**0**
	60000	4.5447	5.7431e-12	5.8795e-19	3.2177e-04	**0**
f_{12} Griwank	40000	47.2934	3.4796e-04	4.1072e-04	1.0704	**0**
	50000	40.1082	1.6072e-07	5.2821e-11	0.8149	**0**
	60000	32.4511	5.6295e-10	9.0377e-04	0.3479	**0**
f_{13} Ackley	40000	2.4609e-05	0.0014	1.6134e-05	2.4991	**3.0376e-14**
	50000	6.9409e-06	1.6258e-04	3.1370e-07	1.3978	**6.9278e-15**
	60000	3.0808e-08	1.4933e-05	7.1968e-09	0.5593	**6.5725e-15**
f_{14} Generalized Penalized1	40000	3.0214	1.0359e-04	0.0015	8.3234e+06	**8.5896e-08**
	50000	3.2638	1.8084e-06	1.5091e-09	1.0149e+06	**2.6364e-12**
	60000	3.0027	7.9157e-09	1.4353e-13	5.1761e+05	**1.0826e-15**
f_{15} Generalized Penalized2	40000	0.5030	1.1632e-06	1.0660e-11	1.6962	**2.4081e-27**
	50000	0.4902	9.2401e-09	4.9678e-15	0.9411	**1.5705e-32**
	60000	0.3199	1.4151e-11	1.5121e-18	0.5631	**1.5705e-32**

except f_2, f_5 and f_9. Although the convergence speed of ABCADE on f_2, f_5 is similar to that of IWEO, and is not much different on f_9, the convergence speed of ABCADE on other functions is obviously slower than that of IWEO. As a result, compared with other algorithms, the algorithm proposed in this paper has obvious advantages in terms of convergence speed.

To compare the convergence process of five algorithms intuitively, the convergence curves of their single run on unimodal functions f_6-f_8 and multimodal functions f_{12}-f_{14} at $d = 30$ are given in Figure 6. As can be seen from Figure 6, IWEO has the fastest

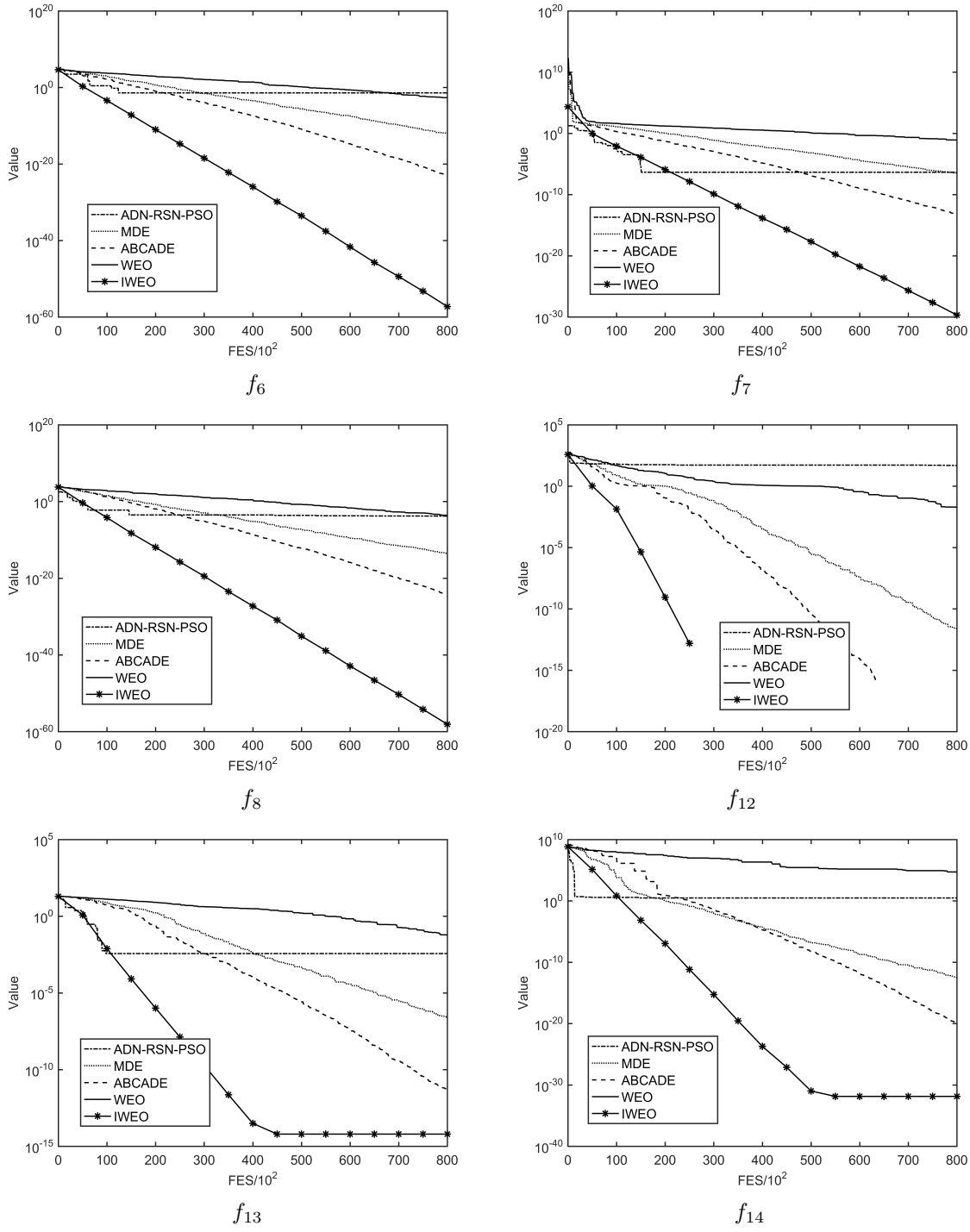


FIGURE 6. The convergence curve on functions at $d = 30$

convergence speed on all functions, especially on unimodal functions. And compared with the algorithms which reach the stagnation state on multimodal functions, IWEO can continue to search until it gains the global optimal solutions.

5. **Conclusions.** In this paper, we proposed an improved water evaporation optimization (IWEO) algorithm. It has the following characteristics. 1) In monolayer evaporation phase, through the theoretical analysis of the relationship between the monolayer evaporation probability matrix (*MEP*) and the updating rate of individuals, the construction method of the *MEP* is improved, which promotes more individual genes to be involved in evolution and speeds up the convergence; meanwhile, multiple excellent evolutionary information is introduced into step size S to balance the exploration and exploitation ability of the algorithm. 2) In droplet evaporation phase, the optimal solution and a new step size factor are used to guide the evolution of individuals and carry out multi-direction disturbance, respectively, which can improve the convergence accuracy as much as possible while maintaining the diversity of the population. In Section 4, a series of experiments on 15 benchmark functions is executed to verify the effectiveness of IWEO. The results show that IWEO has higher convergence accuracy and faster convergence speed than WEO algorithm and other three state-of-the-art metaheuristic algorithms.

In our work, although the IWEO algorithm greatly improves the result quality, it still cannot converge to global optimum stably on some benchmark functions (such as multimodal function f_{14}), and has not been used to solve practical engineering problems. So in the future, we will study the stability problem of IWEO on these functions. At the same time, we will also apply the improved algorithm to solving phased array radar resources management problem.

Acknowledgments. The authors disclosed receipt of the following financial support for the research, authorship of this article: This work was supported in part by the National Natural Science Foundation of China (No. 61501107 and No. 61603073), and the Project of Scientific and Technological Innovation Development of Jilin (No. 201750227 and No. 201750219).

REFERENCES

[1] T. Q. Wu, M. Yao and J. H. Yang, Dolphin swarm algorithm, *Frontiers of Information Technology & Electronic Engineering*, vol.17, no.8, pp.717-729, 2016.

[2] C. Li, H. Zhang, H. Zhang and Y. Liu, Short-term traffic flow prediction algorithm by support vector regression based on artificial bee colony optimization, *ICIC Express Letters*, vol.13, no.6, pp.475-482, 2019.

[3] M. Y. Cheng and D. Prayogo, Symbiotic organisms search: A new metaheuristic optimization algorithm, *Computers & Structures*, vol.139, pp.98-112, 2014.

[4] X. Zhao, X. Wu and X. Yu, An improved genetic algorithm based on a Levy distribution selection operator, *ICIC Express Letters*, vol.11, no.1, pp.87-93, 2017.

[5] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, GSA: A gravitational search algorithm, *Information Sciences*, vol.179, no.13, pp.2232-2248, 2009.

[6] I. Kamkar, M.-R. Akbarzadeh-T and M. Yaghoobi, Intelligent water drops a new optimization algorithm for solving the vehicle routing problem, *IEEE International Conference on Systems Man & Cybernetics*, Istanbul, Turkey, 2010.

[7] H. Eskandar, A. Sadollah, A. Bahreininejad and M. Hamdi, Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems, *Computers & Structures*, vols.110-111, no.10, pp.151-166, 2012.

[8] E. Osaba, F. Diaz and E. Onieva, Golden ball: A novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts, *Applied Intelligence*, vol.41, no.1, pp.145-166, 2014.

[9] A. Kaveh and V. R. Mahdavi, Colliding bodies optimization method for optimum discrete design of truss structures, *Advances in Engineering Software*, vol.139, pp.43-53, 2014.

[10] A. Kaveh and T. Bakhshpoori, Water evaporation optimization: A novel physically inspired optimization algorithm, *Computers & Structures*, vol.167, pp.69-85, 2016.
[11] W. N. Chen, J. Zhang, Y. Lin and N. Chen, Particle swarm optimization with an aging leader and challengers, *IEEE Trans. Evolutionary Computation*, vol.17, no.2, pp.241-258, 2013.
[12] Y. Xiang, Y. Peng, Y. Zhong, Z. Chen, X. Lu and X. Zhong, A particle swarm inspired multi-elitist artificial bee colony algorithm for real-parameter optimization, *Computational Optimization and Applications*, vol.57, no.2, pp.493-516, 2014.
[13] A. Kaveh and T. Bakhshpoori, A new metaheuristic for continuous structural optimization: Water evaporation optimization, *Structural & Multidisciplinary Optimization*, vol.54, no.1, pp.23-43, 2016.
[14] A. Saha, P. Das and A. K. Chakraborty, Water evaporation algorithm: A new metaheuristic algorithm towards the solution of optimal power flow, *Engineering Science & Technology, an International Journal*, vol.20, no.6, pp.1540-1552, 2017.
[15] A. Kaveh and T. Bakhshpoori, An accelerated water evaporation optimization formulation for discrete optimization of skeletal structures, *Computers & Structures*, vol.177, pp.218-228, 2016.
[16] W. Sun, A. Lin, H. Yu, Q. Liang and G. Wu, All-dimension neighborhood based particle swarm optimization with randomly selected neighbors, *Information Sciences*, vol.405, pp.141-156, 2017.
[17] X. Li and M. Yin, Modified differential evolution with self-adaptive parameters method, *Journal of Combinatorial Optimization*, vol.31, no.2, pp.546-576, 2016.
[18] Z. Liang, K. Hu, Q. Zhu and Z. Zhu, An enhanced artificial bee colony algorithm with adaptive differential operators, *Applied Soft Computing Journal*, vol.58, pp.480-494, 2017.
[19] J. Zhan and T. Li, Comparative research on intelligent optimization algorithm of urban underground logistics system, *ICIC Express Letters*, vol.12, no.10, pp.1041-1046, 2018.
[20] S. Wang, Y. Tu, R. Wan and H. Fang, Evaporation of tiny water aggregation on solid surfaces with different wetting properties, *The Journal of Physical Chemistry B*, vol.116, no.47, pp.13863-13867, 2012.