An Adaptive Water Wave Optimization Algorithm with Enhanced Wave Interaction

Yangyan Xu
Department of Computer Science
Hangzhou Normal University
Hangzhou, China
xuyangyan@stu.hznu.edu.cn

Chenxin Wu
Department of Computer Science
Hangzhou Normal University
Hangzhou, China
wuchenxin@compintell.cn

Bingbing Jiang
Department of Computer Science
Hangzhou Normal University
Hangzhou, China
jiangbb@hznu.edu.cn

Weiguo Sheng*
Department of Computer Science
Hangzhou Normal University
Hangzhou, China
*Corresponding author: w.sheng@ieee.org

***Abstract*—Water Wave Optimization (WWO), as a nature-inspired optimization algorithm, has received much attention. In this paper, we tend to improve the algorithm by proposing an adaptive WWO with enhanced wave interaction (AWWO-EI). In the proposed method, two operators (namely, Gaussian-based propagation and refraction learning) with an adaptive mechanism are introduced to enhance the wave interaction in the algorithm. The first operator, Gaussian-based propagation operation, is designed to strengthen the exploration ability of the algorithm by encouraging each individual to learn from different exemplars. While, the second operator, refraction learning, aims to improve the exploitation capability of WWO. Further, rather than using a fixed breaking coefficient, an adaptive mechanism has been employed to dynamically adjust its values during evolution. Experiments have been carried out to evaluate the performance of the proposed method and compare it with related methods. The results have demonstrated the superiority of the proposed method.**

***Index Terms*—Water Wave Optimization, Gaussian-based propagation, refraction learning, parameter adaptation.**

I. Introduction

Water Wave Optimization (WWO) [1] is a stochastic optimization algorithm based on the shallow wave theory. The algorithm has received much attention owing to its easy implementation and simple algorithmic framework [2]–[16]. The main idea of WWO is to assign each solution (analogous to a wave) a wavelength that is inversely proportional to its fitness, and make each wave propagate in a range proportional to its wavelength.

Nevertheless, the basic WWO suffers from the issue of premature convergence. Many WWO variants have been developed [3], [4], [7], [12] to alleviate the issue. For example, Zheng et al. [3] proposed a modified version of WWO by removing the refraction operator, which decreases the possibility of premature convergence. Zhang et al. [4] tried to improve WWO by designing a comprehensive learning mechanism for the refraction process with the purpose of preserving solution diversity. Wu et al. [7] proposed an elite opposition-based WWO, in which an elite opposition-based learning scheme is devised to increase population diversity. Zhang et al. [12] proposed a sine cosine WWO algorithm, in which a sine cosine algorithm is combined with WWO to balance the exploration and exploitation. The above methods could be used to alleviate the premature convergence. However, they are typically time-consuming and could have a limited performance.

Another issue with WWO is that the critical parameters of WWO are kept fixed during the run of WWO. Since the run of WWO is a dynamic process, using a fixed parameter could also significantly restrict the performance of WWO [17], [18]. It is therefore desirable to set these parameters dynamically. Although parameter control has been extensively studied for other nature-inspired algorithms, such as genetic algorithm, no work has been carried out to control the parameters in WWO.

In this paper, we propose an adaptive WWO with enhanced wave interaction, denoted as AWWO-EI, to tackle the issues. In the proposed method, Gaussian-based propagation, refraction learning, and adaptive parameter control mechanisms have been devised. The idea of Gaussian-based propagation is to allow each individual to learn from different exemplars, thus strengthen the exploration ability of WWO. The refraction learning is designed to be based on the best current individual with the purpose of improving the exploitation capability. While the parameter adaptation scheme is devised to dynamically adjust the critical parameter of breaking coefficient in WWO. Numerical results on benchmark suites [19], [20] show that the proposed method has a good performance and outperforms related methods.

The remainder of the paper is structured as follows. Section II briefly reviews related works. Section III describes the proposed AWWO-EI algorithm in detail. Section IV presents the numerical experiments, and finally Section V concludes with a summary and future work.

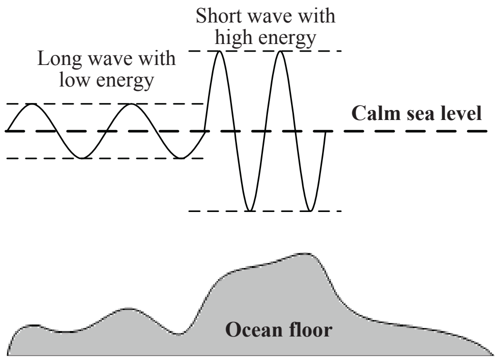

Fig. 1. Illustration of the relationship between wavelengths and wave energy (fitness) in WWO.

II. Related Work

A. The Basic WWO

WWO is an algorithm borrowing ideas from shallow water wave models for continuous optimization, in which each solution x is analogous to a wave. When a wave propagates from deep water (a location with low fitness) to shallow water (a location with high fitness), its wave height h increases and its wavelength λ_x decreases, as illustrated in Fig. 1.

At each generation, for each wave x, the propagation operator gets a new wave x' by creating a different offset at each dimension d and adds it to the original wave:

$$x'(d) = x(d) + \lambda_x \cdot \mathit{rand}(-1,1) \cdot L(d) \tag{1}$$

where *rand* is a function producing a random number uniformly distributed within a specified range, and $L(d)$ is the length of the d^{th} dimension in search space.

At each generation, the wavelength λ_x of each wave x is updated according to its fitness $f(x)$ as follows:

$$\lambda_x = \lambda_x \cdot \alpha^{-(f(x)-f_{\min}+\varepsilon)/(f_{\max}-f_{\min}+\varepsilon)} \tag{2}$$

where $f_{\max}$ and $f_{\min}$ indicate the maximum and minimum fitness values of the current population, α is the wavelength reduction coefficient, and ε is a small value to avoid division by zero.

The breaking operator breaks a newly discovered current best wave x^* into a battery of solitary waves. Each solitary wave is generated by randomly choosing k dimensions (where k is a random number between 1 and a predefined parameter $k_{\max}$) and getting a newly generated offspring at each dimension d as follows:

$$x'(d) = x^*(d) + Gaussian(0,1) \cdot \beta \cdot L(d) \tag{3}$$

where x^* is the current best wave in the population, β is the breaking coefficient, and $Gaussian(0,1)$ is a function producing a Gaussian random number with mean 0 and standard deviation 1. If the fittest one among the solitary waves is better than x^*, x' will replace x^* in the current population.

The refraction operator conducts on waves whose heights h reduces to zero. It allows the stagnant waves to learn from the current best wave x^* at each dimension d:

$$x'(d) = Gaussian(\frac{x^*(d)+x(d)}{2}, \frac{|x^*(d)-x(d)|}{2}) \tag{4}$$

where $Gaussian(\mu, \sigma)$ generates a Gaussian random number with mean μ and standard deviation σ. After refraction, the wavelength of wave x is generated as:

$$\lambda_{x'} = \lambda_x \frac{f(x)}{f(x')} \tag{5}$$

B. Recent Advances

In the past few years, a variety of improved WWO algorithms have been put forward [14], [15], [21]–[24]. In [21], a hybrid algorithm by combining the Firefly with WWO was proposed. In this work, WWO is used for adjusting the parameters of the firefly algorithm. In [22], Shao et al. proposed a discrete WWO to address the blocking flow-shop scheduling problem. In this method, a two-stage propagation is designed to direct the algorithm towards good solutions. In [23], an adaptive wavelength reduction coefficient method is developed to improve the exploration ability of WWO. In [14], Zhang et al. devised a wind-driven WWO to enhance its global search ability. This, however, could make the propagation process of WWO complicated. In [24], the authors integrated a min-max method with WWO to address a multi-objective optimal bidding strategy problem. In this method, a Bare-bone technique implemented in the refraction process is modified to increase the search space. Further, a chaotic map is employed to improve the convergence speed. In [15], WWO was integrated with sequential quadratic programming (SQP) for solving constrained high-dimensional problems. This method is time-consuming as it performs SQP on the individuals obtained by the WWO.

III. Proposed Algorithm

An overview of the proposed method is shown in Algorithm 1. The main idea of the proposed algorithm lies in the two newly designed operators and the introduction of adaptively controlling the parameter of breaking coefficient. In the following subsections, we should describe the proposed algorithm in detail.

A. Gaussian-based Propagation

WWO relies mainly on the propagation operator to explore the solution space. Here, we propose a Gaussian-based propagation operator to enhance the exploration capability of WWO. The proposed Gaussian-based propagation generates a new Gaussian wave by allowing the wave to learn from not only the current best but also other exemplars in the population. Specifically, when performing the Gaussian-based propagation on a wave x, at each dimension d, we first select the one with the best fitness from the current generation, denoted by x^*, as the exemplar. Then, two other individuals are randomly selected from the current population according to:

$$x'(d) = x(d) + (x^*(d) - x(d)) \cdot \gamma + (x_r(d) - x_s(d)) \cdot \gamma \tag{6}$$

Algorithm 1: The framework of AWWO-EI

Input: Define objective function f(x), $x = (x_1, x_2, ..., x_d)$
Output: The best wave x^*

Randomly initialize a population P of N solutions;
Let x^* be the best among the waves;
while *the stop criterion is not satisfied* **do**
  for *each wave* $x \in P$ **do**
    Generate γ according to Eq. (7);
    if $rand() < 0.9$ **then**
      Perform the Gaussian-based propagation operator according to Eq. (6);
    if $f(x') > f(x)$ **then**
      Replace x with x';
      if $f(x') > f(x^*)$ **then**
        Break x' into new waves according to Eq. (3);
        Update x^* with the current best one among the new waves and x';
    else
      Perform the refraction learning operator according to Eq. (8);
  Update memories T_β according to Algorithm 2;
  Update N according to the linear population size reduction strategy;
  Choose top N waves for the next generation of evolution

$$\gamma = Gaussian(\mu, \sigma) \tag{7}$$

Here, x^* denotes the best wave found so far in the current population. The indices r, s are randomly chosen from $[1, N]$, N is the number of waves. γ is a dynamically generated Gaussian random number with a mean μ and a standard deviation σ. By employing this operator, every Gaussian wave has a chance to learn from different exemplars at different dimensions, and thus could be used to greatly increase the solution diversity.

B. Refraction Learning

Our proposed refraction learning is based on the best wave of the current population to improve its exploitation capability. Specifically, when performing the refraction learning operator on a wave x, we update the wave at each dimension d as:

$$x'(d) = x(d) + (x^*(d) - x(d)) \cdot \eta \tag{8}$$

where x^* is the current best individual found so far in the population, η is a predefined parameter which is used to control the magnitude of learning. It should be noted that, in the above process, the limitation of wave height h has been removed. This enables refraction learning to support Gaussian-based propagation to further enhance its search capability.

C. Adaptive Mechanism

The standard WWO contains two main parameters, namely the wavelength reduction coefficient α and the breaking coefficient β. The performance of WWO depends critically on the settings of these parameters and the optimal values depend on the specific problems to be addressed and may change during the run of the algorithm. Inspired by the strategies [25]–[33] of adaptively adjusting the control parameters during the evolutionary process, we introduce the following adaptive mechanism to control the critical parameter of breaking coefficient β in WWO. The procedure of the proposed adaptive control strategy is shown in Algorithm 2.

Algorithm 2: The procedure of updating β.

if $A_\beta \neq \emptyset$ **then**
  $T_{\beta,t,G+1} = mean_L(A_\beta)$;
  if $T_{\beta,t,G+1} < lower$ ***or*** $T_{\beta,t,G+1} > upper$ **then**
    $T_{\beta,t,G+1} = rand(lower, upper)$;
  $t++$;
  if $t > MS$, $t = 1$;
else
  $T_{\beta,t,G+1} = T_{\beta,t,G}$;

At the initial stage, β is set to a random value within a specific range (between 0.001 and 0.01) recommended in the literature [1]. During the procedure, the β_i value that succeeds in producing a new wave $x'_{i,G}$, which is better than the parent wave $x_{i,G}$, is recorded as T_β. At the end of each generation G, the memory contents A_β are updated accordingly. The index t in the algorithm 2 is a number between 1 and MS, which determines the position in the memory to be updated. Here, MS is the memory size to be maintained for the control parameters β, T_β. The values of $lower$ and $upper$ are the recommended range of β. If $A_\beta = \emptyset$, the memory will not be updated. The $mean_L(A_\beta)$ is calculated as Lehmer mean:

$$mean_L(A_\beta) = \frac{\sum_{t=1}^{|A_\beta|} w_t \cdot A_{\beta,t}^2}{\sum_{t=1}^{|A_\beta|} w_t \cdot A_{\beta,t}} \tag{9}$$

$$w_t = \frac{\Delta F_t}{\sum_{l=1}^{|A_\beta|} \cdot \Delta F_l} \tag{10}$$

$$\Delta F_t = |F(x'_{t,G}) - F(x_{t,G})| \tag{11}$$

Here, ΔF_t is fitness improvement during the evolution.

IV. Experiments

A. Experimental Settings

To verify the performance of the proposed algorithm, we conduct numerical experiments on the CEC'14 and CEC'15 test suites [19], [20]. We employ 30 and 100 decision variables of the data sets, and set the maximum NFEs (number of fitness evaluations) of 300,000 and 1,000,000 on these two test suites, respectively. The basic WWO, two variants of WWO

TABLE I
RESULTS ON THE CEC'14 TEST SUITES WITH 30 VARIABLES.

Algorithms		CEC'14 (F1-F10)									
		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
WWO	med	1.8402E+06	6.6527E+03	3.6494E+03	4.9570E+02	**5.2000E+02**	6.2374E+02	7.0001E+02	8.8507E+02	1.0070E+03	3.8902E+03
	std	1.0189E+06	7.4759E+03	1.0669E+04	3.5796E+01	2.3973E-05	5.3100E+00	1.7248E-02	2.5220E+01	3.4476E+01	6.1805E+02
WWO_1	med	4.2332E+06	7.2523E+06	1.7698E+03	5.2363E+02	5.2126E+02	6.0439E+02	7.0105E+02	**8.2494E+02**	**9.2395E+02**	**1.3794E+03**
	std	2.3496E+06	1.6561E+07	1.3734E+03	4.1657E+01	9.3408E-02	1.3278E+00	2.4286E-01	6.4433E+00	4.8205E+00	1.6853E+02
WWO_2	med	**1.8935E+04**	**2.0000E+02**	**3.0000E+02**	4.0001E+02	5.2007E+02	6.0261E+02	7.0001E+02	8.4676E+02	9.4378E+02	2.2705E+03
	std	2.4509E+04	0.0000E+00	0.0000E+00	1.9949E+01	1.9575E-01	1.8357E+00	9.1132E-03	1.6049E+01	1.2057E+01	5.0361E+02
AWWO-EI	med	2.6327E+04	**2.0000E+02**	**3.0000E+02**	**4.0001E+02**	5.2001E+02	**6.0244E+02**	**7.0000E+02**	8.4328E+02	9.4378E+02	2.4782E+03
	std	1.9001E+04	0.0000E+00	0.0000E+00	9.5597E+00	1.1710E-01	1.5459E+00	6.8092E-03	1.2997E+01	1.2216E+01	4.1126E+02

Algorithms		CEC'14 (F11-F20)									
		F11	F12	F13	F14	F15	F16	F17	F18	F19	F20
WWO	med	4.8718E+03	1.2006E+03	1.3004E+03	1.4003E+03	1.5055E+03	1.6115E+03	7.3069E+04	4.1042E+03	1.9141E+03	1.3986E+04
	std	7.9669E+02	3.5535E-01	1.1937E-01	5.0530E-02	1.8299E+00	6.7754E-01	7.3674E+04	3.2979E+03	1.6095E+01	1.2405E+04
WWO_1	med	5.3769E+03	1.2046E+03	1.3003E+03	1.4003E+03	1.5059E+03	1.6124E+03	1.0408E+04	2.0533E+03	1.9057E+03	2.0253E+03
	std	1.2699E+03	1.2547E+00	8.6237E-02	4.8928E-02	2.3843E+00	5.9127E-01	1.3665E+04	4.7708E+02	1.1976E+00	9.2766E+00
WWO_2	med	3.3171E+03	1.2001E+03	1.3002E+03	1.4002E+03	1.5032E+03	**1.6104E+03**	**2.5073E+03**	1.8829E+03	1.9045E+03	**2.0191E+03**
	std	5.7823E+02	6.5738E-02	4.5245E-02	4.2838E-02	7.2044E-01	8.5252E-01	2.9286E+02	2.2051E+01	1.3889E+00	7.7265E+00
AWWO-EI	med	**3.2098E+03**	**1.2001E+03**	**1.3002E+03**	**1.4002E+03**	**1.5031E+03**	1.6105E+03	2.6100E+03	**1.8754E+03**	**1.9045E+03**	2.0196E+03
	std	5.9731E+02	1.0263E-01	4.9921E-02	3.5929E-02	5.9648E-01	7.8638E-01	3.2175E+02	2.7677E+01	1.3116E+00	8.0433E+00

Algorithms		CEC'14 (F21-F30)									
		F21	F22	F23	F24	F25	F26	F27	F28	F29	F30
WWO	med	4.4134E+04	2.6129E+03	2.6164E+03	2.6480E+03	2.7198E+03	2.7004E+03	3.1062E+03	4.4734E+03	4.4269E+03	1.0595E+04
	std	2.7667E+04	1.6559E+02	1.0753E+00	7.0432E+00	6.1093E+00	1.0402E-01	2.5101E+02	6.2018E+02	4.0710E+02	2.5011E+03
WWO_1	med	4.4572E+04	**2.3519E+03**	2.6153E+03	2.6269E+03	2.7088E+03	2.7002E+03	3.1119E+03	**3.5893E+03**	4.2230E+03	4.2984E+03
	std	3.0532E+04	2.9069E+01	1.4518E-01	5.9863E+00	2.4042E+00	2.7369E+01	3.8964E+01	1.6984E+02	2.6708E+02	4.1180E+02
WWO_2	med	**2.3794E+03**	2.3873E+03	**2.6152E+03**	2.6247E+03	**2.7035E+03**	**2.7001E+03**	**3.1006E+03**	3.6684E+03	3.6586E+03	4.0460E+03
	std	1.1534E+02	1.0258E+02	3.2155E-12	3.9204E+00	8.3933E-01	4.8179E-02	4.1430E+01	1.4667E+02	1.8099E+02	3.3820E+02
AWWO-EI	med	2.3887E+03	2.3729E+03	**2.6152E+03**	**2.6244E+03**	2.7036E+03	2.7002E+03	3.1007E+03	3.6596E+03	**3.6122E+03**	**3.8869E+03**
	std	1.1494E+02	8.5388E+01	3.2155E-12	2.9748E+00	6.7721E-01	3.6630E-02	2.4394E+01	1.6403E+02	1.6731E+02	3.6284E+02
Total number of (+/=/-): WWO (29/0/1), WWO_1 (25/0/5), WWO_2 (3/24/3)											

(SimWWO and VarWWO [3]), as well as the following four popular evolutionary algorithms are used for comparison:

- A basic DE algorithm [34].
- A basic PSO algorithm [35].
- A BBO algorithm [36].
- A TLBO algorithm [37].

For AWWO-EI, we set the memory size MS = 5, k_{max} = 6, γ is dynamically generated from a Gaussian random number with a mean of 0.5 and a standard deviation of 0.3, η = 0.005, β is adaptively adjusted according to Algorithm 2, and N linearly reduces from N_{max} to N_{min} (N_{max} = $20 \times d$, N_{min} = 3, d is the dimension). The control parameters of the other seven algorithms are set as suggested in the corresponding papers. All compared algorithms employ the same number of NFEs as the termination condition.

All algorithms are run with a computer of Intel Core i7-8700 3.20 GHz CPU, 16 GB RAM. We run each algorithm 50 times on each test problem, and record the medians (med) and standard deviations (std) of function values among these runs. On each problem, the best median fitness values among the algorithms to be compared are marked with boldface in the results.

Additionally, nonparametric Wilcoxon rank sum test at a 0.05 significance level has been performed between AWWO-EI and each algorithm to be compared on each benchmark function. The sign "+" in the results indicates the performance of AWWO-EI is significantly better than the corresponding algorithm, "-" vice versa and "=" denotes there is no significant difference between the performance.

B. Exploring the Proposed Strategies

In this section, we first explore the proposed method. For this purpose, we compare AWWO-EI with its two variants including WWO_1, WWO_2 as well as the basic WWO. WWO_1 is equipped with Gaussian-based propagation only. WWO_2 is equipped with both Gaussian-based propagation and refraction learning. AWWO-EI is equipped with all three proposed strategies.

The results are shown in Table I. From the results, we can see that among the 30 benchmark functions, AWWO-EI achieves the best median values on sixteen functions, which is significantly better than all the three algorithms to be compared. Compared with WWO, the results also show that the differences are statistically significant as it achieves better performance on all functions except one. From the above results, it is clear that the proposed three strategies could significantly improve the performance of WWO.

C. Comparisons with Related Algorithms

In this section, we compare the performance of the proposed method with related seven methods. Table II presents the results of the eight algorithms to be compared.

We first examine the results on two unimodal functions. The comparison results show that AWWO-EI achieves the best performance on both functions in terms of the median values. Compared with the canonical WWO, AWWO-EI can learn from more exemplars, which enhances the wave interaction. This could strengthen the exploration capability of the algorithm. Further, the refraction learning in the proposed method could appropriately readjust its search direction, thus promoting information sharing among the waves in the population. Fi-

TABLE II
RESULTS ON THE CEC'15 WITH 30 AND 100 VARIABLES.

Algorithms		CEC'15 (F1-F10) with 30 variables									
		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
DE	med	4.9168E+06	1.2174E+03	3.2037E+02	4.5661E+02	3.4777E+03	1.4738E+06	7.1109E+02	3.1085E+05	1.0077E+03	2.9380E+05
	std	1.5636E+06	1.7510E+03	3.9542E-02	7.2926E+00	2.8354E+02	7.7573E+05	1.5470E+00	2.6394E+05	1.2302E+00	2.2914E+05
PSO	med	5.6270E+04	3.6136E+03	3.2000E+02	4.8956E+02	3.7291E+03	2.7272E+04	7.0659E+02	1.8072E+04	1.0103E+03	8.2091E+03
	std	4.1595E+04	4.0543E+03	4.6515E-02	3.5399E+01	6.3922E+02	2.3431E+04	5.3335E+00	1.9190E+04	3.8218E+01	4.8544E+03
BBO	med	1.1164E+07	2.9991E+03	**3.2000E+02**	4.8235E+02	3.6469E+03	2.3475E+06	7.1426E+02	8.5526E+05	1.0094E+03	9.5850E+05
	std	5.9239E+06	3.7883E+03	1.6543E-02	1.7634E+01	5.9823E+02	3.1388E+06	1.1652E+01	1.2472E+06	3.6748E+01	1.4688E+06
TLBO	med	9.2532E+04	1.9184E+03	3.2095E+02	5.1094E+02	5.3023E+03	4.8865E+04	7.1122E+02	5.9950E+04	1.0108E+03	1.5164E+04
	std	5.1530E+05	4.5738E+03	5.0489E-02	2.2405E+01	1.2856E+03	5.5395E+04	1.4490E+01	4.6774E+04	3.2062E+01	1.7203E+04
WWO	med	9.7146E+05	1.2748E+03	**3.2000E+02**	5.1641E+02	3.9663E+03	5.0976E+04	7.1730E+02	3.5054E+04	1.0080E+03	4.0195E+04
	std	6.5755E+05	3.0516E+03	5.9296E-06	2.9656E+01	6.9309E+02	7.2573E+04	2.0083E+00	1.6501E+04	1.0824E+00	3.4939E+04
SimWWO	med	3.1090E+06	2.0712E+02	**3.2000E+02**	4.7412E+02	2.8221E+03	9.0358E+04	7.1220E+02	4.0109E+04	1.0098E+03	9.8389E+04
	std	9.0251E+05	3.5852E+01	1.1285E-05	1.2462E+01	3.4523E+02	4.3652E+04	1.2483E+00	1.5844E+04	6.3132E-01	5.5414E+04
VarWWO	med	1.3288E+06	9.3770E+02	**3.2000E+02**	5.0700E+02	3.9701E+03	3.6613E+04	7.1642E+02	3.0775E+04	1.0085E+03	4.1187E+04
	std	6.6044E+05	2.1507E+03	6.1224E-06	2.6529E+01	7.0652E+02	2.7347E+04	2.8214E+00	1.9878E+04	1.2468E+00	2.6955E+04
AWWO-EI	med	**9.1011E+03**	**2.0000E+02**	3.2004E+02	**4.4577E+02**	**2.6309E+03**	**1.2890E+03**	**7.0402E+02**	**9.6210E+02**	**1.0064E+03**	**1.6587E+03**
	std	1.5594E+04	0.0000E+00	1.9254E-01	1.2393E+01	6.2762E+02	3.3383E+02	1.1364E+00	1.1734E+02	1.0821E+00	1.3961E+02

Algorithms		CEC'15 (F11-F15) with 30 variables and CEC'15 (F1-F5) with 100 variables									
		F11	F12	F13	F14	F15	F1	F2	F3	F4	F5
DE	med	1.6891E+03	1.3115E+03	**1.3000E+03**	**3.7866E+04**	**1.6000E+03**	6.3120E+08	1.5414E+03	3.2113E+02	1.1930E+03	2.6303E+04
	std	1.2330E+02	3.3124E+00	3.1706E-04	2.8155E+03	0.0000E+00	8.6432E+07	6.2264E+07	2.9100E-02	2.6086E+01	4.7895E+02
PSO	med	1.9398E+03	1.3157E+03	1.3000E+03	4.5440E+04	**1.6000E+03**	4.6675E+06	3.7254E+03	3.2003E+02	9.9903E+02	1.4530E+04
	std	1.1160E+02	3.9200E+01	1.2426E-03	4.5713E+03	5.4771E-01	1.3047E+06	6.9768E+03	2.7683E-02	1.0028E+02	1.4400E+03
BBO	med	1.8304E+03	1.3121E+03	1.3000E+03	4.6435E+04	**1.6000E+03**	2.1304E+08	1.4188E+07	3.2000E+02	8.5237E+02	1.3876E+04
	std	8.7380E+01	2.9391E+01	1.5055E-03	2.2385E+03	1.0160E+00	6.2818E+07	6.0476E+07	1.8520E-02	7.7491E+01	1.1635E+03
TLBO	med	2.0452E+03	1.4009E+03	1.3000E+03	4.6385E+04	1.6070E+03	1.2914E+07	6.0286E+07	3.2135E+02	1.1435E+03	2.8680E+04
	std	1.1615E+02	3.0802E+01	2.1699E-03	4.7294E+03	6.3572E+00	6.1138E+06	1.2450E+09	3.0774E-02	8.2899E+01	2.8534E+03
WWO	med	2.0461E+03	1.3146E+03	1.3000E+03	4.8628E+04	**1.6000E+03**	3.5245E+07	2.6073E+03	**3.2000E+02**	1.1049E+03	1.4646E+04
	std	2.1847E+02	2.2856E+00	3.2699E-02	1.6746E+03	1.4684E-08	1.0141E+07	6.1566E+03	2.0789E-04	1.0923E+02	2.4827E+03
SimWWO	med	**1.4220E+03**	1.3124E+03	1.3000E+03	4.7714E+04	**1.6000E+03**	1.2082E+08	8.5734E+05	3.2000E+02	1.0019E+03	1.2292E+04
	std	1.1316E+02	7.0967E-01	7.3617E-03	6.8540E+02	0.0000E+00	1.4212E+07	2.7532E+06	2.2712E-04	3.5757E+01	5.6406E+02
VarWWO	med	2.1018E+03	1.3137E+03	1.3000E+03	4.8055E+04	**1.6000E+03**	3.3280E+07	2.4477E+03	**3.2000E+02**	1.0562E+03	1.5079E+04
	std	2.4747E+02	2.6209E+00	2.1936E-02	1.8866E+03	2.2717E-10	1.1045E+07	7.4628E+03	2.0654E-06	8.8519E+01	1.3180E+03
AWWO-EI	med	1.5415E+03	**1.3084E+03**	1.3000E+03	4.5692E+04	**1.6000E+03**	**2.0228E+06**	**2.0000E+02**	3.2087E+02	**7.1739E+02**	**1.1951E+04**
	std	9.2891E+01	7.6936E-01	1.8086E-03	4.8006E+03	0.0000E+00	8.0216E+05	3.6439E-05	3.4005E-01	4.7753E+01	1.3393E+03

Algorithms		CEC'15 (F6-F15) with 100 variables									
		F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
DE	med	5.3803E+07	8.5495E+02	2.7055E+07	1.0087E+03	3.2946E+06	4.2601E+03	1.4004E+03	**1.3001E+03**	**1.1028E+05**	**1.6000E+03**
	std	1.0671E+07	9.7041E+00	6.7919E+06	5.2011E-01	1.1922E+06	9.3620E+01	2.8944E+01	8.2427E-04	1.3155E+03	0.0000E+00
PSO	med	1.3030E+06	8.1253E+02	6.2019E+05	1.0178E+03	1.5231E+04	4.1567E+03	1.4004E+03	1.3002E+03	1.1034E+05	1.6321E+03
	std	4.6248E+05	2.4809E+01	2.6081E+05	2.1553E+02	1.9373E+04	1.8830E+02	2.5223E+01	9.1408E-02	1.0811E+04	1.6438E+01
BBO	med	3.1251E+07	8.8847E+02	1.8060E+07	1.0156E+03	2.2426E+07	3.3984E+03	1.3593E+03	1.3003E+03	1.7868E+05	1.7515E+03
	std	9.5986E+06	3.3392E+01	8.8297E+06	1.8807E+02	1.1070E+07	1.6212E+02	3.4968E+01	1.0166E-01	9.0899E+03	8.3671E+01
TLBO	med	2.0492E+06	8.7738E+02	8.5573E+05	1.0157E+03	3.7404E+05	4.8575E+03	1.4005E+03	1.3002E+03	1.4625E+05	1.6883E+03
	std	7.9705E+05	3.8545E+01	4.6104E+05	1.9786E+02	7.7238E+05	1.7288E+02	4.6359E-02	1.4001E-01	1.8270E+04	4.2427E+01
WWO	med	2.8405E+06	8.7519E+02	5.3835E+05	1.0101E+03	6.3639E+05	4.5492E+03	1.3179E+03	1.3020E+03	1.7047E+05	1.6157E+03
	std	9.5102E+05	3.6550E+01	4.0062E+05	9.3761E-01	2.9757E+05	5.6281E+02	3.5228E+01	2.0863E+00	5.9238E+03	4.1657E+00
SimWWO	med	7.5499E+06	8.5447E+02	3.3131E+06	1.0162E+03	3.3583E+06	4.0183E+03	1.3685E+03	1.3139E+03	1.9696E+05	1.6467E+03
	std	1.6970E+06	1.9712E+01	1.0318E+06	7.0360E-01	7.7710E+05	1.2379E+03	8.7380E+00	4.0936E+00	6.7558E+03	8.4457E+00
VarWWO	med	2.4490E+06	8.8048E+02	6.2655E+05	1.0092E+03	7.0624E+05	4.6422E+03	1.3179E+03	1.3016E+03	1.7045E+05	1.6140E+03
	std	9.6500E+05	3.8280E+01	3.8107E+05	7.2597E-01	4.4094E+05	2.7053E+02	3.4909E+01	6.5483E-01	5.9505E+03	3.7530E+00
AWWO-EI	med	**1.9414E+05**	**7.7601E+02**	**4.7394E+04**	**1.0071E+03**	**5.6856E+03**	**2.9576E+03**	**1.3167E+03**	1.3006E+03	1.1172E+05	1.6036E+03
	std	8.0196E+04	4.4941E+01	1.9731E+04	5.5369E-01	6.3752E+02	4.1404E+02	3.1223E+01	1.7077E-01	1.7241E+04	6.9043E-01

nally, the adaptive strategy introduced in the proposed method can automatically adjust the breaking coefficient, thus improve the performance further. We then access the results on three multimodal functions. From the comparison results, we can see that AWWO-EI yields the best performance on two out of three multimodal problems. Subsequently, for the third set of experiments including three hybrid functions. Still, the results show that AWWO-EI achieves the best performance on all the three functions. For the fourth set of experiments including seven composition functions, AWWO-EI could obtain the best performance in terms of median values on F9, F10, F12, and F15 functions. From the statistical test results in Table III, we can see that the performance differences are statistically significant.

Experiments with the CEC'15 benchmark problems of 100 variables have also been carried out. The results are reported in Table II. From the results, we can find that AWWO-EI can achieve better results than the algorithms to be compared on eleven (F1, F2, and F4-F12) out of fifteen functions. It may indicate that AWWO-EI is well suited to address optimization problems with large search space. The convergence curves can be found in Fig. 2.

V. CONCLUSIONS

In this paper, an adaptive WWO with enhanced wave interaction (AWWO-EI) is proposed. In the proposed method, a Gaussian-based propagation is designed to learn different exemplars, which enhances the exploration ability of the algorithm. Further, refraction learning is developed to learn from the best current wave to enhance the exploitation capability

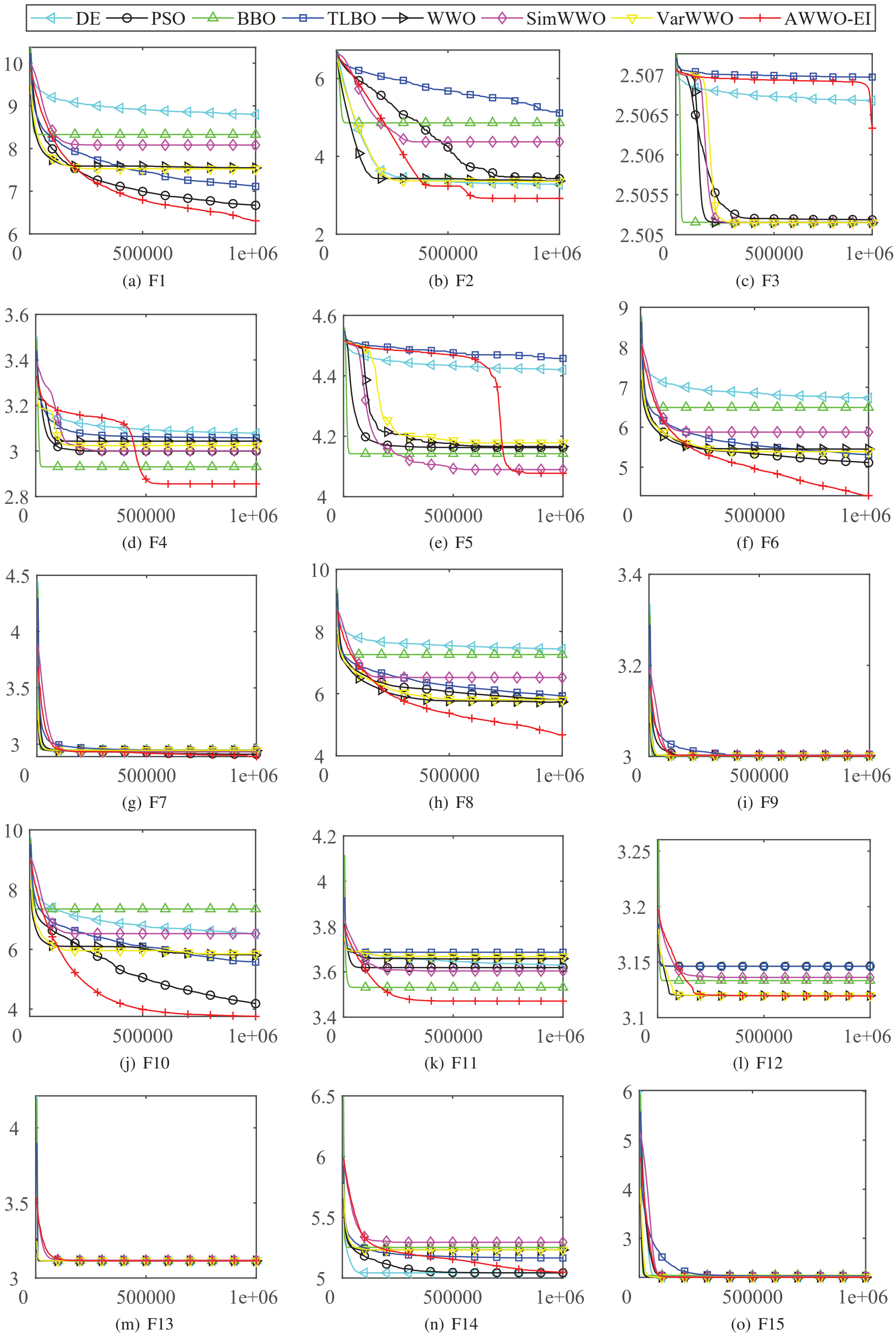

Fig. 2. Convergence curves of the eight algorithms on the CEC'15 test problems with 100 variables. The x-axis denotes NFEs, and the y-axis denotes the common logarithm of the median values.

TABLE III
THE RESULTS OF WILCOXON RANK SUM TESTS.

Results for the CEC'15 with 30 variables
Total number of (+/=/-): DE (12/1/2), PSO (11/2/2), BBO (12/1/2), TLBO (14/0/1), WWO (14/0/1), SimWWO (11/2/2), VarWWO (14/0/1)
Results for the CEC'15 with 100 variables
Total number of (+/=/-): DE (12/0/3), PSO (12/0/3), BBO (13/0/2), TLBO (14/0/1), WWO (14/0/1), SimWWO (12/2/1), VarWWO (14/0/1)

of the algorithm. Additionally, we strengthen the performance of the proposed method by adaptively adjusting the breaking coefficient. The experiments have demonstrated the goodness of our proposed algorithm and it could achieve significantly better results than related works.

In future work, we will tend to introduce niching techniques [38]–[41] into WWO for multimodal optimization problems.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61873082.

REFERENCES

[1] Y. J. Zheng, "Water wave optimization: a new nature-inspired metaheuristic," *Computers & Operations Research*, vol. 55, pp. 1–11, 2015.

[2] X. B. Wu, J. Liao, and Z. C. Wang, "Water wave optimization for the traveling salesman problem," in *International Conference on Intelligent Computing*. Springer, 2015, pp. 137–146.

[3] Y. J. Zheng and B. Zhang, "A simplified water wave optimization algorithm," in *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 807–813.

[4] B. Zhang, M. X. Zhang, J. F. Zhang, and Y. J. Zheng, "A water wave optimization algorithm with variable population size and comprehensive learning," in *International Conference on Intelligent Computing*. Springer, 2015, pp. 124–136.

[5] M. Kilany, A. E. Hassanien, and A. Badr, "Accelerometer-based human activity classification using water wave optimization approach," in *2015 11th International Computer Engineering Conference (ICENCO)*. IEEE, 2015, pp. 175–180.

[6] W. Xu, Z. Ye, and Y. Hou, "A fast image match method based on water wave optimization and gray relational analysis," in *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 2. IEEE, 2017, pp. 771–776.

[7] X. Wu, Y. Zhou, and Y. Lu, "Elite opposition-based water wave optimization algorithm for global optimization," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[8] F. Zhao, L. Zhang, H. Liu, Y. Zhang, W. Ma, C. Zhang, and H. Song, "An improved water wave optimization algorithm with the single wave mechanism for the no-wait flow-shop scheduling problem," *Engineering Optimization*, pp. 1–16, 2018.

[9] Y. Zhou, J. Zhang, X. Yang, and Y. Ling, "Optimal reactive power dispatch using water wave optimization algorithm," *Operational Research*, pp. 1–17, 2018.

[10] X. H. Zhou, Z. G. Xu, M. X. Zhang, and Y. J. Zheng, "Water wave optimization for artificial neural network parameter and structure optimization," in *International Conference on Bio-Inspired Computing: Theories and Applications*. Springer, 2018, pp. 343–354.

[11] H. Chen, Y. Hou, Q. Luo, Z. Hu, and L. Yan, "Text feature selection based on water wave optimization algorithm," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*. IEEE, 2018, pp. 546–551.

[12] J. Zhang, Y. Zhou, and Q. Luo, "An improved sine cosine water wave optimization algorithm for global optimization," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 4, pp. 2129–2141, 2018.

[13] F. Zhao, H. Liu, Y. Zhang, W. Ma, and C. Zhang, "A discrete water wave optimization algorithm for no-wait flow shop scheduling problem," *Expert Systems with Applications*, vol. 91, pp. 347–363, 2018.

[14] J. Zhang, Y. Zhou, and Q. Luo, "Nature-inspired approach: a wind-driven water wave optimization algorithm," *Applied Intelligence*, vol. 49, no. 1, pp. 233–252, 2019.

[15] G. Singh, M. Rattan, S. S. Gill, and N. Mittal, "Hybridization of water wave optimization and sequential quadratic programming for cognitive radio system," *Soft Computing*, vol. 23, no. 17, pp. 7991–8011, 2019.

[16] A. Liu, P. Li, W. Sun, X. Deng, W. Li, Y. Zhao, and B. Liu, "Prediction of mechanical properties of micro-alloyed steels via neural networks learned by water wave optimization," *Neural Computing and Applications*, pp. 1–16, 2019.

[17] H. H. Hoos, "Automated algorithm configuration and parameter tuning," in *Autonomous Search*. Springer, 2011, pp. 37–71.

[18] C. Huang, Y. Li, and X. Yao, "A survey of automatic parameter tuning methods for metaheuristics," *IEEE Transactions on Evolutionary Computation*, 2019.

[19] J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, vol. 635, 2013.

[20] J. Liang, B. Qu, P. Suganthan, and Q. Chen, "Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization," *Technical Report201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, 2014.

[21] K. Lenin, B. Ravindhranath Reddy, and M. Suryakalavathi, "Hybridization of firefly and water wave algorithm for solving reactive power problem," in *International Journal of Engineering Research in Africa*, vol. 21. Trans Tech Publ, 2016, pp. 165–171.

[22] Z. Shao, D. Pi, and W. Shao, "A novel discrete water wave optimization algorithm for blocking flow-shop scheduling problem with sequence-dependent setup times," *Swarm and Evolutionary Computation*, vol. 40, pp. 53–75, 2018.

[23] A. Soltanian, F. Derakhshan, and M. Soleimanpour-Moghadam, "Mwwo: Modified water wave optimization," in *2018 3rd Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*. IEEE, 2018, pp. 1–5.

[24] A. A. Hematabadi and A. A. Foroud, "Optimizing the multi-objective bidding strategy using min-max technique and modified water wave optimization method," *Neural Computing and Applications*, vol. 31, no. 9, pp. 5207–5225, 2019.

[25] Á. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[26] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[27] S. Meyer Nieberg and H. G. Beyer, "Self-adaptation in evolutionary algorithms," in *Parameter Setting in Evolutionary Algorithms*. Springer, 2007, pp. 47–75.

[28] F. Peng, K. Tang, G. Chen, and X. Yao, "Multi-start jade with knowledge transfer for numerical optimization," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 1889–1895.

[29] W. Sheng, S. Chen, M. Fairhurst, G. Xiao, and J. Mao, "Multilocal search and adaptive niching based memetic algorithm with a consensus criterion for data clustering," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 5, pp. 721–741, 2013.

[30] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 71–78.

[31] W. Sheng, S. Chen, M. Sheng, G. Xiao, J. Mao, and Y. Zheng, "Adaptive multi-subpopulation competition and multi-niche crowding based memetic algorithm for automatic data clustering," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 838–858, 2016.

[32] R. Tanabe and A. S. Fukunaga, "Improving the search performance of shade using linear population size reduction," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 1658–1665.

[33] W. Sheng, P. Shan, J. Mao, Y. Zheng, S. Chen, and Z. Wang, "An adaptive memetic algorithm with rank-based mutation for artificial neural network architecture optimization," *IEEE Access*, vol. 5, pp. 18 895–18 908, 2017.

[34] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[35] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4. Citeseer, 1995, pp. 1942–1948.

[36] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[37] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[38] W. Sheng, S. Swift, L. Zhang, and X. Liu, "A weighted sum validity function for clustering with a hybrid niching genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 6, pp. 1156–1167, 2005.

[39] L. Wei and M. Zhao, "A niche hybrid genetic algorithm for global optimization of continuous multimodal functions," *Applied Mathematics and Computation*, vol. 160, no. 3, pp. 649–661, 2005.

[40] W. Sheng, P. Shan, S. Chen, Y. Liu, and F. E. Alsaadi, "A niching evolutionary algorithm with adaptive negative correlation learning for neural network ensemble," *Neurocomputing*, vol. 247, pp. 173–182, 2017.

[41] Z. J. Wang, Z. H. Zhan, Y. Lin, W. J. Yu, H. Wang, S. Kwong, and J. Zhang, "Automatic niching differential evolution with contour prediction approach for multimodal optimization problems," *IEEE Transactions on Evolutionary Computation*, 2019.