Computers & Operations Research 55 (2015) 1–11

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

# Water wave optimization: A new nature-inspired metaheuristic

Yu-Jun Zheng

*College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China*

ARTICLE INFO

Available online 23 October 2014

*Keywords:*
Optimization
Metaheuristic method
Water wave optimization (WWO)
Wave-current-bottom interactions
High-speed train scheduling

ABSTRACT

Nature-inspired computing has been a hot topic in scientific and engineering fields in recent years. Inspired by the shallow water wave theory, the paper presents a novel metaheuristic method, named water wave optimization (WWO), for global optimization problems. We show how the beautiful phenomena of water waves, such as propagation, refraction, and breaking, can be used to derive effective mechanisms for searching in a high-dimensional solution space. In general, the algorithmic framework of WWO is simple, and easy to implement with a small-size population and only a few control parameters. We have tested WWO on a diverse set of benchmark problems, and applied WWO to a real-world high-speed train scheduling problem in China. The computational results demonstrate that WWO is very competitive with state-of-the-art evolutionary algorithms including invasive weed optimization (IWO), biogeography-based optimization (BBO), bat algorithm (BA), etc. The new metaheuristic is expected to have wide applications in real-world engineering optimization problems.



## 1. Introduction

Nature-inspired computing has been fascinating computer scientists for a long time, giving rise to popular areas such as artificial neural networks [1], cellular automata [2], molecular computing [3], and evolutionary algorithms (EAs) [4]. Taking inspiration from natural evolution processes, EAs are a class of metaheuristic methods for solving complex optimization problems which typically have non-convex and highly nonlinear solution spaces, and which are otherwise computationally difficult to solve by conventional mathematical programming methods. Due to the increasing complexity of real-world optimization problems, on one hand, classical EAs including genetic algorithms (GAs) [5], evolutionary programming (EP) [6], evolution strategies (ES) [7], particle swarm optimization (PSO) [8], etc., have been extensively studied; on the other hand, a variety of novel EAs, such as invasive weed optimization (IWO) [9], water drops algorithm [10], biogeography-based optimization (BBO) [11], cuckoo search [12], fireworks algorithm [13], bat algorithm (BA) [14], etc., have been proposed and have aroused much interests in the last years [15–22].

More than 71% of the earth's surface is covered by water. Surface water waves can be generated by any sort of geophysical mass flow. When entering shoaling waters, surface waves are either refracted by varying depth or current, or diffracted around abrupt bathymetric features such as submarine ridges or valleys, losing part of their energy. They continuing their shoreward march give up some energy by dissipation near the bottom. Nevertheless each crest becomes steeper and mightier, and makes its final display of power by breaking and splashing on the shoreline [23].

Historically, the first to attempt a theory of water waves subjected to gravity force, surface tension, and other forces, traces back the work by Newton in 1687, against hydrostatics by Archimedes in the 3rd century BC [24]. The linear wave theory reached a real level of advances by the works of Laplace, Lagrange, Poisson, and Cauchy, accompanied by nonlinear waves considered by Gerstner, Stokes, and Kelland [25].

Modern water wave theory began with weak, nonlinear interactions among gravity waves on the surfaces of deep water [26], which were extended by Hasselmann [27], and subsequently culminated in the wave turbulence theory by Zakharov et al. [28]. In shallow coastal water, the nonlinear wave field is dominated by near-resonant quadratic interactions involving triplets of waves. It is the main wave–current–bottom interactions that have made rich and progressive coastal wave modeling since the late 1960s, albeit less mature relative to the well-established deep-water wave models [25].

In this paper we propose a new optimization method inspired by shallow water wave models. The metaheuristic, named water wave optimization (WWO), borrows ideas from wave motions controlled by the wave–current–bottom interactions to the design of search mechanisms for high-dimensional global optimization problems. Experiments on a diverse set of function optimization problems show that WWO is very competitive with some popular meta-heuristic algorithms proposed in recent years, such as IWO, BBO, BA, etc. We have also successfully applied WWO to a high-speed train scheduling problem in China, the results of which

*E-mail address:* yujun.zheng@computer.org

http://dx.doi.org/10.1016/j.cor.2014.10.008

demonstrate the applicability and effectiveness of WWO to real-world problems.

The rest of the paper is organized as follows: Section 2 gives a brief review of the water wave theory. Section 3 describes how the shallow water wave models can be used to derive an effective metaheuristic optimization method. Section 4 presents the comparative experiments on benchmark functions and Section 5 depicts the application to a train scheduling problem, and finally Section 6 concludes with discussion.

## 2. Water wave theory

In *Principia* (1687), Newton proposed an analogy with oscillations in a U-tube, deducing that the frequency of deep-water waves must be proportional to the inverse of the square root of the "breadth of the wave". But he was aware that the result was approximate, observing that "These things are true upon the supposition that the parts of water ascend or descend in a right line; but in truth, that ascent and descent is rather performed in a circle".

From first principles, Laplace (1776) showed that wave motion starting from rest is governed by (what we now call) Laplace's equation, and arrived at the periodic solutions for linear plane waves:

$$x = A\left(e^{z_0/c} + e^{-z_0/c}\right)\sin\frac{x_0}{c} \tag{1}$$

$$z = A\left(e^{z_0/c} - e^{-z_0/c}\right)\cos\frac{x_0}{c} \tag{2}$$

where $x$ and $z$ respectively denote the small horizontal and vertical displacements of individual fluid particles with initial positions $(x_0, z_0)$, $A$ is a function of time $t$, and $c$ is a constant. Laplace further observed that the products in the above equations can be decomposed into oppositely traveling waves with forms $\cos(x_0/c \pm ft)$, where $f$ is the wave frequency.

From the nonlinear free-surface boundary conditions, Kelland [29] tackled waves in fluid of arbitrary depth and obtained (implicitly) the surface displacement as

$$z = h + (e^{\alpha z} - e^{\alpha z})\alpha\ \sin(\alpha(ct - x)) \tag{3}$$

where $\alpha = 2\pi/\lambda$, $\lambda$ is wavelength, $c$ is wave speed, and $h$ is depth. It turns out that, when $z$ is calculated by successive approximations, the result contains higher harmonic terms that are correct up to third order in wave amplitude [30].

Mainly due to the complicated seabed topologies in coastal regions, coastal (shallow-water) waves have not been studied as thoroughly as deep-water waves. In conjunction with the effects of ambient currents, wave–current–bottom interactions make up the most fundamental dynamic mechanism in coastal waters manifesting itself as refraction, diffraction, scattering, and resonant wave–wave interactions involved in energy change.

The main wave–current–bottom interactions have made rich and progressive coastal wave modeling since the late 1960s [25]. The first generation wave models have been formulated in terms of the basic transport equation, consisting of a superposition of the energy input by the wind, the nonlinear transfer due to resonant wave–wave interactions, and the dissipation due to white capping and turbulence [27]. However, it has now also become clear that a universal high-frequency equilibrium spectrum of the form originally proposed by Phillips [26] does not exist, and these models overestimated the wind input and underestimated the strength of the nonlinear transfer by almost an order of magnitude [31].

In 1970s, extensive wave growth measurements and experiments have leaded to the development of the second generation wave models. These models fundamentally changed the view of the spectral energy balance of the first generation ones by introducing the specification of the spectral shape, either at the outset in the formulation of the transport equation itself or as a side condition in the computation of the spectrum [32]. However, they were unable to properly simulate complex windseas generated by rapidly changing wind fields, and also encountered basic difficulties in treating the transition between windsea and swell.

An exhaustive study of first and second generation wave models was carried out in [33], and it was proposed that in third generation models the wave spectrum should be computed alone by integration of the basic spectral transport equation, without any prior restriction of the spectral shape. The first attempt of such a model was the WAM model [31], which describes the evolution of a two-dimensional wave spectrum on a spherical latitude–longitude grid for an arbitrary region of the ocean. The evolution equation of the one-dimensional version of WAM is

$$\frac{\partial}{\partial t}F = S_{in} + S_{dis} + S_{nl} \tag{4}$$

where $F$ is the wave spectrum, $S_{in}$ is the input term, $S_{dis}$ is the dissipation term, and $S_{nl}$ is the nonlinear wave-wave interaction term.

Taking the formulations for the generation, the dissipation and the quadruplet wave-wave interactions from the WAM model, Booij et al. [34] developed the SWAN model, which is a fully discrete spectral model based on the action balance equation which implicitly takes into account the interaction between waves and currents through radiation stresses:

$$\frac{\partial}{\partial t}N(\sigma,\theta) + \nabla_{x,y}(c_{x,y}N(\sigma,\theta)) + \frac{\partial}{\partial\sigma}(c_\sigma N(\sigma,\theta)) + \frac{\partial}{\partial\sigma}(c_\theta N(\sigma,\theta)) = \frac{S(\sigma,\theta)}{\sigma} \tag{5}$$

where in the left-hand side, the first term is the rate of change of action density in time, the second is the rectilinear propagation of action in geographical $(x,y)$-space, the third is the shifting of the relative frequency due to currents and time-varying depths with propagation velocity $c_\sigma$ in $\sigma$-space, and the fourth term is the propagation in $\theta$-space (depth- and current-induced refraction) with propagation velocity $c_\theta$. The term $S(\sigma,\theta)$ at the right hand side is the source term representing the growth by wind, the wave-wave interactions and the decay by bottom friction, white-capping and depth-induced wave breaking.

In summary, the wave models describe the evolution of wave heights, periods, and propagation directions using numerical techniques considering wind forcing, nonlinear wave interactions, frictional dissipation, etc [35]. The models are undergoing constant improvement, in terms of speed, accuracy and generality. Nevertheless, simplified analytical models remain (and will remain) in common use, and they indeed provide insight and are also accurate enough for many practical applications [36].

## 3. Water wave optimization

The WWO takes inspiration from shallow water wave models for solving optimization problems. Without losing generality, suppose we have a maximization problem with objective function $f$. In WWO, the solution space $X$ is analogous to the seabed area, and the fitness of a point $\mathbf{x} \in X$ is measured inversely by its seabed depth: the shorter the distance to the still water level, the higher the fitness $f(\mathbf{x})$ is. It should be noted that by analogy the 3-D space of the seabed is generalized to an $n$-dimensional space.

As most other EAs, WWO maintains a population of solutions, each of which is analogous to a "wave" having a height (or amplitude) $h \in \mathbb{Z}^+$ and a wavelength $\lambda \in \mathbb{R}^+$. Upon initialization, for each wave, $h$ is set to a constant $h_{\max}$ and $\lambda$ is set to 0.5. During the problem-solving process, we consider three types of operations on the waves: *Propagation*, *Refraction*, and *Breaking*.

### 3.1. Propagation

At each generation, each wave needs to be propagated exactly once. The propagation operator creates a new wave $\mathbf{x}'$ by shifting each dimension $d$ of the original wave $\mathbf{x}$ as

$$x'(d) = x(d) + rand(-1,1) \cdot \lambda L(d) \tag{6}$$

where $rand(-1,1)$ is a uniformly distributed random number within the range $[-1,1]$, and $L(d)$ is the length of the $d$th dimension of the search space ($1 \le d \le n$). If the new position is outside the feasible range, it will be reset to a random position in the range.

When a wave travels from deep water (low fitness location) to shallow water (high fitness location), its wave height increases and its wavelength decreases, as illustrated in Fig. 1. Thus, after propagation we calculate the fitness of the offspring wave $\mathbf{x}'$. If $f(\mathbf{x}') > f(\mathbf{x})$, $\mathbf{x}$ is replaced by $\mathbf{x}'$ in the population, and the wave height of $\mathbf{x}'$ is reset to $h_{\max}$. Otherwise, $\mathbf{x}$ is remained, but its height $h$ is decreased by one, which mimics energy dissipation due to inertial resistance, vortex shedding, and bottom friction.

After each generation, the wavelength of each wave $\mathbf{x}$ is updated as follows:

$$\lambda = \lambda \cdot \alpha^{-(f(\mathbf{x})-f_{\min}+\epsilon)/(f_{\max}-f_{\min}+\epsilon)} \tag{7}$$

where $f_{\max}$ and $f_{\min}$ are respectively the maximum and minimum fitness values among the current population, $\alpha$ is the wavelength reduction coefficient, and $\epsilon$ is a very small positive number to avoid division-by-zero. Eq. (7) ensures that higher fitness waves have smaller wavelengths, and thus propagate within smaller ranges.

### 3.2. Refraction

In wave propagation, if the wave ray is not perpendicular to the isobath, its direction will be deflected. It is observed that the rays converge in shallow regions while diverge in deep regions, as illustrated in Fig. 2.

In WWO, we only perform refraction on waves whose heights decrease to zero, and use a simple way to calculate the position after refraction:

$$x'(d) = N\left(\frac{x^*(d)+x(d)}{2}, \frac{|x^*(d)-x(d)|}{2}\right) \tag{8}$$

where $\mathbf{x}^*$ is the best solution found so far, and $N(\mu,\sigma)$ is a Gaussian random number with mean $\mu$ and standard deviation $\sigma$. That is, the new position is a random number centered halfway between the original position and the known best position, and the standard deviation equal to the absolute value of their difference. Such a calculation has turned out to be competitive for many difficult numerical optimization problems [37].

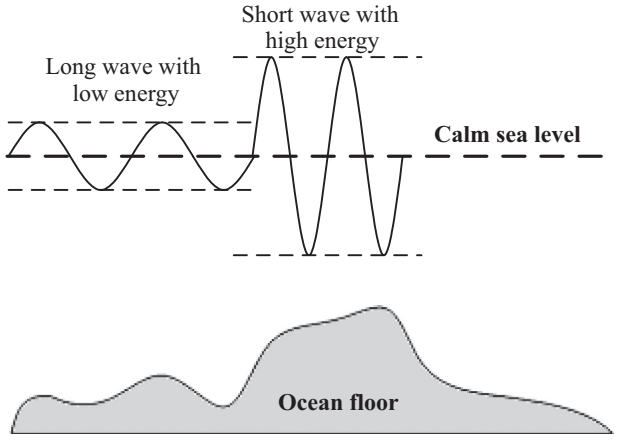


**Fig. 1.** Different wave shapes in deep and shallow water.

After refraction, the wave height of $\mathbf{x}'$ is also reset to $h_{\max}$, and its wavelength is set as

$$\lambda' = \lambda \frac{f(\mathbf{x})}{f(\mathbf{x}')} \tag{9}$$

### 3.3. Breaking

When a wave moves to a position where the water depth is below a threshold value, the wave crest velocity exceeds the wave celerity. Consequently, the crest becomes steeper and steeper, and finally the wave breaks into a train of solitary waves, as illustrated in Fig. 3.

In WWO we perform the breaking operation only on a wave $\mathbf{x}$ that finds a new best solution (i.e., $\mathbf{x}$ becomes the new $\mathbf{x}^*$), and conduct a local search around $\mathbf{x}^*$ to simulate wave breaking. In detail, we randomly choose $k$ dimensions (where $k$ is a random number between 1 and a predefined number $k_{\max}$), and at each dimension $d$ generate a solitary wave $\mathbf{x}'$ as

$$x'(d) = x(d) + N(0,1) \cdot \beta L(d) \tag{10}$$

where $\beta$ is the breaking coefficient. If none of the solitary waves are better than $\mathbf{x}^*$, $\mathbf{x}^*$ is remained; Otherwise $\mathbf{x}^*$ is replaced by the fittest one among the solitary waves.

### 3.4. The algorithmic framework of WWO

Now we can give the general framework of WWO, as shown in Algorithm 1. In general, the propagation operator makes high fitness waves search small areas and low fitness waves explore large areas; the refraction operator helps waves to escape search stagnation, and thus improves the diversity of the population and reduces premature convergence; the breaking operator enables an intensive search around a (potentially) promising area. The combination of the three operators provides the algorithm with a good balance between exploration and exploitation.

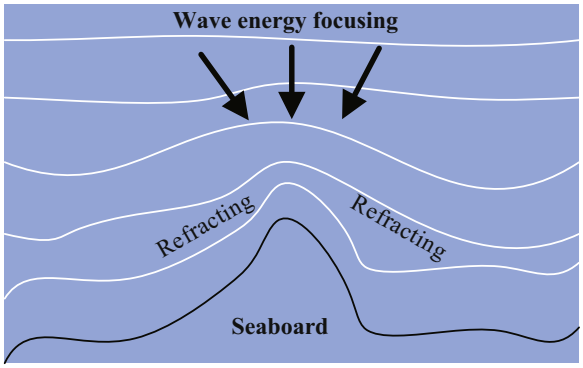


**Fig. 2.** Wave refraction.

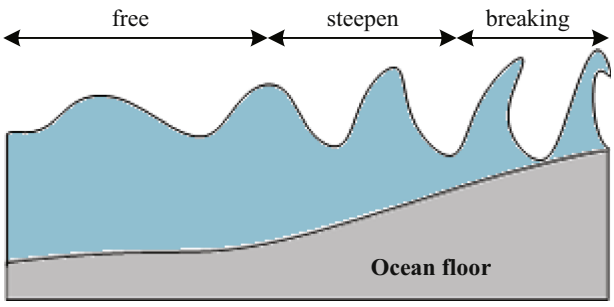


**Fig. 3.** Wave breaking.

**Algorithm 1.** The WWO algorithm.

```
Randomly initialize a population P of n waves (solutions);
while stop criterion is not satisfied do
  for each x ∈ P do
    Propagate x to a new x′ based on Eq. (6);
    if f(x′) > f(x) then
      if f(x′) > f(x*) then
        Break x′ based on Eq. (10);
        Update x* with x′;
      Replace x with x′;
    else
      Decrease x.h by one;
      if x.h = 0 then
        Refract x to a new x′ based on Eq. (8) and (9);
  Update the wavelengths based on Eq. (7);
return x*.
```

As we can see, the framework is simple and thus easy to implement. Also, empirical tests show that, WWO performs well with a small population size $n$ (about 5–10), which makes the algorithm more computationally efficient and has a high potential for parallelization. Another choice is to use a population size reduction strategy, which is effective in improving performance of many EAs [38].

Besides the population size, there are four control parameters of WWO: the maximum wave height $h_{\max}$, the wavelength reduction coefficient $\alpha$, the breaking coefficient $\beta$, and the maximum number $k_{\max}$ of breaking directions. In general, the larger $h_{\max}$ the longer the average life span of the waves is; if $h_{\max}$ is small (say 1 or 2), the waves will be frequently replaced by new waves, and therefore the solution diversity will be increased. A large $\alpha$ causes the algorithm to explore a large area, while a small $\alpha$ makes the algorithm perform a more intensive exploitation. Furthermore, a small $\beta$ with a large $k_{\max}$ help the improvement of best solutions and thus increase the convergence speed. Empirically, we recommend to set $h_{\max}$ to 5 or 6, $\alpha$ to 1.001–1.01, $\beta$ to 0.001–0.01, and $k_{\max}$ to $\min(12, D/2)$, where $D$ is the dimension of the problem.

## 4. Computational experiment

### 4.1. Experimental settings and comparative methods

First we test the performance of the proposed WWO on 30 benchmark functions of the CEC 2014 Competition on Single Objective Real-Parameter Numerical Optimization [39]. The benchmark suite covers various types of function optimization problems, as summarized in Table 1. Details definitions of the functions can be found in [39].

On the test functions, we compare WWO with five recent popular metaheuristic methods:

- The invasive weed optimization (IWO) algorithm, which simulates colonizing behavior of weeds by using seeds reproduction, spatial dispersal, and competitive exclusion operations [9].
- The biogeography-based optimization (BBO) algorithm, which borrows ideas from island biogeography to evolve a population of solutions to a given optimization problem by continuously migrating features from (probably) high-fitness individuals to low-fitness ones [11].
- The gravitational search algorithm (GSA), which assigns each solution to the problem with four specifications including position, inertia, active gravitational mass and passive gravitational mass, and performs the search using a collection of masses interacting with each other based on the laws of gravity and motion [40].

**Table 1**
Summary of the CEC 2014 benchmark functions.

| Type | ID | Function | $f^*$ |
|---|---|---|---|
| Unimodal | $f_1$ | Rotated high conditioned elliptic function | 100 |
| | $f_2$ | Rotated bent cigar function | 200 |
| | $f_3$ | Rotated discus function | 300 |
| Multimodal | $f_4$ | Shifted and rotated Rosenbrock function | 400 |
| | $f_5$ | Shifted and rotated Ackley's function | 500 |
| | $f_6$ | Shifted and rotated Weierstrass function | 600 |
| | $f_7$ | Shifted and rotated Griewank's function | 700 |
| | $f_8$ | Shifted Rastrigin function | 800 |
| | $f_9$ | Shifted and rotated Rastrigin's function | 900 |
| | $f_{10}$ | Shifted Schwefel function | 1000 |
| | $f_{11}$ | Shifted and rotated Schwefel's function | 1100 |
| | $f_{12}$ | Shifted and rotated Katsuura function | 1200 |
| | $f_{13}$ | Shifted and rotated HappyCat function | 1300 |
| | $f_{14}$ | Shifted and rotated HGBat function | 1400 |
| | $f_{15}$ | Shifted and rotated Expanded Griewank's plus Rosenbrock's function | 1500 |
| | $f_{16}$ | Shifted and rotated Expanded Scaffer's F6 function | 1600 |
| Hybrid | $f_{17}$ | Hybrid function 1 ($f_9$,$f_8$,$f_1$) | 1700 |
| | $f_{18}$ | Hybrid function 2 ($f_2$,$f_{12}$,$f_8$) | 1800 |
| | $f_{19}$ | Hybrid function 3 ($f_7$,$f_6$,$f_4$,$f_{14}$) | 1900 |
| | $f_{20}$ | Hybrid function 4 ($f_{12}$,$f_3$,$f_{13}$,$f_8$) | 2000 |
| | $f_{21}$ | Hybrid function 5 ($f_{14}$,$f_{12}$,$f_4$,$f_9$,$f_1$) | 2100 |
| | $f_{22}$ | Hybrid function 6 ($f_{10}$,$f_{11}$,$f_{13}$,$f_9$,$f_5$) | 2200 |
| Composition | $f_{23}$ | Composition function 1 ($f_4$,$f_1$,$f_2$,$f_3$,$f_1$) | 2300 |
| | $f_{24}$ | Composition function 2 ($f_{10}$,$f_9$,$f_{14}$) | 2400 |
| | $f_{25}$ | Composition function 3 ($f_{11}$,$f_9$,$f_1$) | 2500 |
| | $f_{26}$ | Composition function 4 ($f_{11}$,$f_{13}$,$f_1$,$f_6$,$f_7$) | 2600 |
| | $f_{27}$ | Composition function 5 ($f_{14}$,$f_9$,$f_{11}$,$f_6$,$f_1$) | 2700 |
| | $f_{28}$ | Composition function 6 ($f_{15}$,$f_{13}$,$f_{11}$,$f_{16}$,$f_1$) | 2800 |
| | $f_{29}$ | Composition function 7 ($f_{17}$,$f_{18}$,$f_{19}$) | 2900 |
| | $f_{30}$ | Composition function 8 ($f_{20}$,$f_{21}$,$f_{22}$) | 3000 |

**Table 2**
Comparative results on unimodal benchmark functions.

| | | IWO | BBO | GSA | HuS | BA | WWO |
|---|---|---|---|---|---|---|---|
| $f_1$ | max | 2.77E+06 | 8.09E+07 | 5.31E+07 | 1.26E+07 | 5.51E+08 | 1.17E+06 |
| | min | 3.44E+05 | 5.75E+06 | 4.56E+06 | 1.61E+06 | 1.18E+08 | **1.44E+05** |
| | median | [2]1.42E+06 | [5]2.14E+07 | [4]8.37E+06 | [3]5.10E+06 | [6]3.10E+08 | [1]**6.26E+05** |
| | std | 5.72E+05 | 1.67E+07 | 1.32E+07 | 2.62E+06 | 1.05E+08 | 2.45E+05 |
| $f_2$ | max | 4.06E+04 | 8.04E+06 | 1.61E+04 | 2.41E+04 | 6.35E+09 | 1.48E+03 |
| | min | 6.09E+03 | 1.15E+06 | 3.47E+03 | 3.09E+02 | 1.13E+09 | **2.00E+02** |
| | median | [4]1.52E+04 | [5]3.95E+06 | [2]8.38E+03 | [3]9.09E+03 | [6]2.49E+09 | [1]**2.68E+02** |
| | std | 8.67E+03 | 1.55E+06 | 2.90E+03 | 6.01E+03 | 7.55E+08 | 2.02E+02 |
| $f_3$ | max | 1.50E+04 | 5.07E+04 | 7.58E+04 | 3.36E+03 | 1.11E+05 | 1.32E+03 |
| | min | 3.50E+03 | 5.92E+02 | 2.04E+04 | **3.00E+02** | 3.44E+04 | 3.15E+02 |
| | median | [3]7.29E+03 | [4]7.65E+03 | [5]4.51E+04 | [1]**3.02E+02** | [6]7.19E+04 | [2]4.87E+02 |
| | std | 2.69E+03 | 1.28E+04 | 1.04E+04 | 5.41E+02 | 1.75E+04 | 1.85E+02 |

- The hunting search (HuS) algorithm, which is derived based on a model of group hunting of animals when searching for food, making artificial hunters (solutions) move towards the leader while avoiding hunters to be too close to each other by position correction [41].
- The bat algorithm (BA), where virtual bats (solutions) move in the search space, adjusting their velocities and positions by using echo to detect the distance and orientation of targets [14].

It should be noted that fine-tuning control parameters for each problem can effectively improve algorithm performance. However, searching for distinct parameter settings for each problem can be very time-consuming and costly, and such tuning processes may lead to unfair comparison. Thereby, here we adopt a fixed parameter setting for every algorithm in order to evaluate its overall performance on the whole benchmark suite. The recommended parameter setting of the algorithms are as follows.

- IWO: The maximum and minimum number of seeds $s_{\max}=3$, $s_{\min}=0$, the initial and maximum population sizes $p_{\text{init}}=10$, $p_{\max}=30$, the initial and final values of standard deviation of spatial dispersal $\sigma_{\text{init}}(d)=0.25(U_d-L_d)$, $\sigma_{\text{final}}(d)=0.0005(U_d-L_d)$, where $U_d$ and $L_d$ are respectively the upper and lower limits of the $d$th dimension.
- BBO: A simple linear migration model is employed, where the maximum immigration rate $I$ and maximum emigration rate $E$ are both set to 1, and the population size $p=50$.
- GSA: The initial gravitational constant $G_0=100$ and its decreasing coefficient $\alpha=20$, $p=50$, and the number of active agents used in gravitational computation linearly decreases from $p$ to 1.
- HuS: The maximum movement toward the leader $MML=0.3$, hunting group consideration rate $HGCR=0.3$, the maximum and minimum of relative search radius $Ra_{\max}=0.01$, $Ra_{\min}=5\text{E}-6$, the iterations per epoch $E=50$, reorganization parameters $\alpha=0.05$, $\beta=-0.5$, and $p=10$.
- BA: The maximum and minimum frequency rates $f_{\max}=2$, $f_{\min}=0$, the maximum and minimum echo loudness $A_{\max}=1$, $A_{\min}=0$, the maximum and minimum wavelength $r_{\max}=1$, $r_{\min}=0$, the decreasing coefficients $\alpha=\gamma=0.9$, and $p=25$.
- WWO: $h_{\max}=6$, $\alpha=1.026$, $\beta$ linearly decreases from 0.25 to 0.001, and $p$ linearly decreases from 50 to 3.

The experimental environment is a computer of Intel Core i7-4500 M processor and 8GB DDR3 memory. In the experiments, we use 30-D problems for testing, and set the maximum number of fitness evaluations (NFE) to 150,000 for every algorithm on the problems to ensure a fair comparison. Every algorithm has been run 60 times (with different random seeds) on each test problem, and the evaluation is based on the average performance over the 60 runs.

### 4.2. Experimental results

Tables 3–6 respectively present the experimental results on unimodal, multimodal, hybrid and composition functions, where "max" and "min" respectively denote the maximum and minimum fitness values of the algorithm among the 60 runs, "median" denotes the median of the result fitness values over the 60 runs, "std" is the corresponding standard deviation, and the superscript before the median denotes the rank of the algorithm in terms of median values among the six algorithms. Best min and medium results among the comparative algorithms on each problem are shown in bold.

Table 6 summarizes the rankings of the six algorithms over the test functions, from which we can see that the proposed WWO algorithm has the highest overall ranking on the whole benchmark suite as well as on every group of test functions.

Moreover, we conduct nonparametric Wilcoxon rank sum tests on the results of WWO and other comparative algorithms on the 30 benchmark functions, and present the test results in Table 7, where an $h$ value of 1 indicates that the performances of WWO and the comparative method are statistically different with 95% confidence, 0 implies there is no statistical difference, superscript $^+$ denotes WWO has significant performance improvement over the comparative method and $^-$ vice versa.

### 4.3. Result discussion

On the first unimodal group, WWO obtains the best median values on $f_1$ and $f_2$, and obtains the second best median value on $f_3$. According to the statistical tests, the results of WWO are significantly different from the other five algorithms on $f_1$ and $f_2$, and different from the other four except HuS on $f_3$. It should be noted that most comparative algorithms such as BA perform well on basic unimodal functions [14], but lose their performance on these three rotated unimodal functions. In contrast, WWO is capable of handling these rotated trap problems very effectively.

On the second multimodal group of 13 functions, WWO obtains the best median values on 7 functions, obtains the second best median values on 4 functions ($f_5$, $f_6$, $f_{10}$ and $f_{16}$), and ranks third on $f_9$ and fourth on $f_{11}$, and thus also exhibits the best overall performance. In some details:

- GSA ranks first on $f_5$, IWO ranks first on $f_6$, and BBO obtains the best median values on $f_{10}$ and $f_{16}$. WWO ranks second and outperforms four other algorithms on the 4 functions.
- IWO and BBO respectively rank first and second on $f_9$, followed by WWO which is significantly different from (better than) GSA, HuS and BA. $f_9$, the shifted and rotated Rastrigin's function, has a huge

**Table 3**
Comparative results on multimodal benchmark functions.

| | | IWO | BBO | GSA | HuS | BA | WWO |
|---|---|---|---|---|---|---|---|
| $f_4$ | max | 5.45E+02 | 6.54E+02 | 8.49E+02 | 5.64E+02 | 1.26E+04 | 5.42E+02 |
| | min | 4.02E+02 | 4.23E+02 | 5.73E+02 | 4.04E+02 | 2.01E+03 | **4.00E+02** |
| | median | [3]5.11E+02 | [4]5.42E+02 | [5]6.82E+02 | [2]5.03E+02 | [6]3.05E+03 | [1]**4.02E+02** |
| | std | 2.88E+01 | 3.84E+01 | 5.15E+01 | 3.66E+01 | 1.97E+03 | 3.64E+01 |
| $f_5$ | max | 5.20E+02 | 5.20E+02 | 5.20E+02 | 5.21E+02 | 5.21E+02 | 5.20E+02 |
| | min | 5.20E+02 | 5.20E+02 | **5.20E+02** | 5.21E+02 | 5.21E+02 | 5.20E+02 |
| | median | [3]5.20E+02 | [4]5.20E+02 | [1]**5.20E+02** | [5]5.21E+02 | [6]5.21E+02 | [2]5.20E+02 |
| | std | 3.77E−03 | 4.22E−02 | 6.47E−04 | 7.83E−02 | 4.81E−02 | 6.98E−04 |
| $f_6$ | max | 6.05E+02 | 6.18E+02 | 6.24E+02 | 6.29E+02 | 6.39E+02 | 6.13E+02 |
| | min | **6.00E+02** | 6.08E+02 | 6.17E+02 | 6.19E+02 | 6.32E+02 | 6.01E+02 |
| | median | [1]**6.02E+02** | [3]6.14E+02 | [4]6.20E+02 | [5]6.23E+02 | [6]6.37E+02 | [2]6.06E+02 |
| | std | 1.12E+00 | 2.35E+00 | 1.83E+00 | 2.18E+00 | 1.56E+00 | 2.62E+00 |
| $f_7$ | max | 7.00E+02 | 7.01E+02 | 7.00E+02 | 7.00E+02 | 9.63E+02 | 7.00E+02 |
| | min | 7.00E+02 | 7.01E+02 | **7.00E+02** | **7.00E+02** | 8.19E+02 | **7.00E+02** |
| | median | [4]7.00E+02 | [5]7.01E+02 | [1]**7.00E+02** | [3]7.00E+02 | [6]9.12E+02 | [1]**7.00E+02** |
| | std | 1.21E−02 | 2.64E−02 | 9.55E−04 | 5.56E−02 | 3.23E−01 | 6.26E−03 |
| $f_8$ | max | 8.75E−02 | 9.39E+02 | 8.01E+02 | 9.75E+02 | 1.12E+03 | 8.15E+02 |
| | min | 8.27E+02 | 8.39E+02 | 8.00E+02 | 9.10E+02 | 9.76E+02 | **8.00E+02** |
| | median | [3]8.43E+02 | [4]8.79E+02 | [2]8.00E+02 | [5]9.40E+02 | [6]1.07E+03 | [1]**8.00E+02** |
| | std | 1.01E+01 | 2.07E+01 | 2.06E−01 | 1.27E+01 | 2.56E+01 | 2.34E+00 |
| $f_9$ | max | 9.78E+02 | 9.84E+02 | 1.10E+03 | 1.09E+03 | 1.34E+03 | 9.84E+02 |
| | min | **9.30E+02** | 9.35E+02 | 1.02E+03 | 9.59E+02 | 1.15E+03 | 9.35E+02 |
| | median | [1]**9.46E+02** | [2]9.49E+02 | [5]1.06E+03 | [4]1.01E+03 | [6]1.25E+03 | [3]9.61E+02 |
| | std | 1.14E+01 | 1.14E+01 | 1.74E+01 | 2.60E+01 | 4.41E+01 | 1.11E+01 |
| $f_{10}$ | max | 3.57E+03 | 1.00E+03 | 5.25E+03 | 3.21E+03 | 7.45E+03 | 2.71E+03 |
| | min | 1.59E+03 | **1.00E+03** | 3.45E+03 | 1.36E+03 | 5.26E+03 | 1.02E+03 |
| | median | [4]2.58E+03 | [1]**1.00E+03** | [5]4.37E+03 | [3]2.17E+03 | [6]6.47E+03 | [2]1.49E+03 |
| | std | 3.80E+02 | 6.80E−01 | 3.61E+02 | 4.33E+02 | 5.19E+02 | 3.62E+02 |
| $f_{11}$ | max | 3.80E+03 | 4.51E+03 | 6.35E+03 | 4.23E+03 | 8.75E+03 | 3.89E+03 |
| | min | **1.48E+03** | 2.12E+03 | 3.70E+03 | 2.20E+03 | 7.20E+03 | 2.49E+03 |
| | median | [1]**2.92E+03** | [3]3.32E+03 | [5]4.99E+03 | [2]3.24E+03 | [6]8.24E+03 | [4]3.38E+03 |
| | std | 4.48E+02 | 5.12E+02 | 5.67E+02 | 4.66E+02 | 3.62E+02 | 2.89E+02 |
| $f_{12}$ | max | 1.20E+03 | 1.20E+03 | 1.20E+03 | 1.20E+03 | 1.20E+03 | 1.20E+03 |
| | min | **1.20E+03** | **1.20E+03** | **1.20E+03** | **1.20E+03** | **1.20E+03** | **1.20E+03** |
| | median | [1]**1.20E+03** | [1]**1.20E+03** | [1]**1.20E+03** | [1]**1.20E+03** | [1]**1.20E+03** | [1]**1.20E+03** |
| | std | 1.48E−02 | 5.62E−02 | 1.00E−03 | 7.77E−02 | 3.34E−01 | 5.61E−02 |
| $f_{13}$ | max | 1.30E+03 | 1.30E+03 | 1.30E+03 | 1.30E+03 | 1.30E+03 | 1.30E+03 |
| | min | 1.30E+03 | 1.30E+03 | 1.30E+03 | 1.30E+03 | 1.30E+03 | **1.30E+03** |
| | median | [2]1.30E+03 | [5]1.30E+03 | [3]1.30E+03 | [4]1.30E+03 | [6]1.30E+03 | [1]**1.30E+03** |
| | std | 6.50E−02 | 1.06E−01 | 6.65E−02 | 6.50E−02 | 5.48E−01 | 6.41E−02 |
| $f_{14}$ | max | 1.40E+03 | 1.40E+03 | 1.40E+03 | 1.40E+03 | 1.50E+03 | 1.40E+03 |
| | min | 1.40E+03 | 1.40E+03 | 1.40E+03 | 1.40E+03 | 1.44E+03 | **1.40E+03** |
| | median | [2]1.40E+03 | [5]1.40E+03 | [4]1.40E+03 | [3]1.40E+03 | [6]1.47E+03 | [1]**1.40E+03** |
| | std | 1.19E−01 | 1.99E−01 | 4.23E−02 | 4.74E−02 | 1.39E−01 | 4.41E−02 |
| $f_{15}$ | max | 1.51E+03 | 1.53E+03 | 1.51E+03 | 1.52E+03 | 5.92E+05 | 1.50E+03 |
| | min | **1.50E+03** | 1.51E+03 | 1.50E+03 | 1.51E+03 | 1.59E+04 | 1.50E+03 |
| | median | [3]1.50E+03 | [4]1.51E+03 | [2]1.50E+03 | [5]1.52E+03 | [6]1.55E+05 | [1]**1.50E+03** |
| | std | 8.48E−01 | 4.30E+00 | 7.30E−01 | 3.27E+00 | 1.40E+05 | 7.75E−01 |
| $f_{16}$ | max | 1.61E+03 | 1.61E+03 | 1.61E+03 | 1.61E+03 | 1.61E+03 | 1.61E+03 |
| | min | 1.61E+03 | **1.61E+03** | 1.61E+03 | 1.61E+03 | 1.61E+03 | 1.61E+03 |
| | median | [3]1.61E+03 | [1]**1.61E+03** | [6]1.61E+03 | [4]1.61E+03 | [5]1.61E+03 | [2]1.61E+03 |
| | std | 6.14E−01 | 5.92E−01 | 3.43E−01 | 7.25E−01 | 1.90E−01 | 4.67E−01 |

On $f_{13}$-$f_{16}$, the values in bold are better than those seemingly same values not in bold, because the digits after the second decimal place are omitted.

number of local optima, and thus is hard for the algorithms to obtain the global optimum in one or several runs.

- $f_{11}$ is the shifted and rotated Schwefel's function which also has a huge number of local optima, and the second best local optimum is far from the global optimum. The median value of WWO on $f_{11}$ only ranks fourth (following IWO, HuS and BBO), which is its worst performance in the multimodal group. However, there is no statistically significant difference between WWO and BBO/HuS, i.e., only IWO has significant performance improvement over WWO.
- There is no statistically significant difference between the results of the six algorithms on $f_{12}$.
- On the remaining 6 functions, WWO always obtains the best median values, and its performance is significantly different from most other algorithms.

In particular, the results of WWO often reach or become very close to real optima on functions such as $f_2$ and $f_4$ which are narrow ridge (or have a very narrow valley from local optimum to global optimum).

**Table 4**
Comparative results on hybrid benchmark functions.

| | | IWO | BBO | GSA | HuS | BA | WWO |
|---|---|---|---|---|---|---|---|
| $f_{17}$ | max | 3.50E+05 | 2.31E+07 | 1.14E+06 | 1.10E+06 | 9.90E+06 | 6.16E+04 |
| | min | **5.37E+03** | 1.26E+06 | 1.85E+05 | 1.43E+04 | 1.45E+06 | 6.71E+03 |
| | median | [2]6.75E+04 | [5]3.13E+06 | [4]5.63E+05 | [3]1.51E+05 | [6]4.24E+06 | [1]**2.61E+04** |
| | std | 6.85E+04 | 4.19E+06 | 2.20E+05 | 1.61E+05 | 1.79E+06 | 1.24E+04 |
| $f_{18}$ | max | 1.80E+04 | 1.03E+05 | 4.20E+03 | 1.09E+04 | 3.64E+08 | 2.73E+03 |
| | min | 2.26E+03 | 6.74E+03 | 2.02E+03 | 2.02E+03 | 1.33E+07 | **1.85E+03** |
| | median | [4]4.35E+03 | [5]2.28E+04 | [2]2.13E+03 | [3]2.73E+03 | [6]8.54E+07 | [1]**2.01E+03** |
| | std | 3.69E+03 | 1.97E+04 | 3.78E+02 | 2.25E+03 | 1.00E+08 | 1.25E+02 |
| $f_{19}$ | max | 1.91E+03 | 1.98E+03 | 2.00E+03 | 2.04E+03 | 2.06E+06 | 1.91E+03 |
| | min | 1.90E+03 | 1.91E+03 | 1.91E+03 | 1.91E+03 | 1.95E+03 | **1.90E+03** |
| | median | [2]1.91E+03 | [3]1.91E+03 | [5]2.00E+03 | [4]1.92E+03 | [6]2.01E+03 | [1]**1.91E+03** |
| | std | 1.65E+00 | 2.77E+01 | 3.43E+01 | 3.31E+01 | 2.03E+01 | 1.38E+00 |
| $f_{20}$ | max | 5.34E+03 | 8.62E+04 | 6.82E+04 | 6.03E+04 | 4.44E+04 | 1.58E+04 |
| | min | 2.30E+03 | 8.64E+03 | 2.32E+03 | 2.22E+04 | 5.40E+03 | **2.14E+03** |
| | median | [1]**2.74E+03** | [5]2.72E+04 | [4]1.77E+04 | [6]3.68E+04 | [3]1.63E+04 | [2]4.25E+03 |
| | std | 7.00E+02 | 1.76E+04 | 1.39E+04 | 8.49E+03 | 1.03E+04 | 3.18E+03 |
| $f_{21}$ | max | 9.03E+04 | 1.67E+06 | 3.09E+05 | 1.66E+05 | 3.34E+06 | 1.76E+05 |
| | min | 6.74E+03 | 6.70E+04 | 5.87E+04 | 1.07E+04 | 1.43E+05 | **3.70E+03** |
| | median | [2]3.35E+04 | [5]4.22E+05 | [4]1.71E+05 | [3]4.70E+04 | [6]9.17E+05 | [1]**2.92E+04** |
| | std | 2.30E+04 | 3.35E+05 | 6.53E+04 | 4.24E+04 | 7.51E+05 | 3.50E+04 |
| $f_{22}$ | max | 2.52E+03 | 3.28E+03 | 3.63E+03 | 3.67E+03 | 3.56E+03 | 2.85E+03 |
| | min | 2.23E+03 | 2.25E+03 | 2.63E+03 | 2.37E+03 | 2.72E+03 | **2.22E+03** |
| | median | [1]**2.36E+03** | [3]2.71E+03 | [6]3.15E+03 | [4]3.08E+03 | [5]3.14E+03 | [2]2.48E+03 |
| | std | 7.34E+01 | 2.34E+02 | 2.50E+02 | 2.67E+02 | 2.05E+02 | 1.43E+02 |

On $f_{19}$, the values in bold are better than those seemingly same values not in bold, because the digits after the second decimal place are omitted.

**Table 5**
Comparative results on composition benchmark functions.

| | | IWO | BBO | GSA | HuS | BA | WWO |
|---|---|---|---|---|---|---|---|
| $f_{23}$ | max | 2.62E+03 | 2.62E+03 | 2.65E+03 | 2.62E+03 | 2.88E+03 | 2.62E+03 |
| | min | 2.62E+03 | 2.62E+03 | 2.50E+03 | 2.62E+03 | **2.51E+03** | 2.62E+03 |
| | median | [4]2.62E+03 | [6]2.62E+03 | [2]2.56E+03 | [5]2.62E+03 | [1]**2.51E+03** | [3]2.62E+03 |
| | std | 7.95E−02 | 1.32E+00 | 6.45E+01 | 8.45E−01 | 1.28E+02 | 1.45E−01 |
| $f_{24}$ | max | 2.63E+03 | 2.65E+03 | 2.60E+03 | 2.71E+03 | 2.60E+03 | 2.63E+03 |
| | min | 2.60E+03 | 2.63E+03 | **2.60E+03** | 2.63E+03 | 2.60E+03 | 2.62E+03 |
| | median | [3]2.62E+03 | [5]2.63E+03 | [1]**2.60E+03** | [6]2.66E+03 | [2]2.60E+03 | [4]2.63E+03 |
| | std | 1.08E+01 | 5.97E+00 | 1.71E−02 | 1.25E+01 | 1.20E+00 | 6.89E+00 |
| $f_{25}$ | max | 2.71E+03 | 2.72E+03 | 2.71E+03 | 2.75E+03 | 2.76E+03 | 2.72E+03 |
| | min | 2.70E+03 | 2.71E+03 | **2.70E+03** | 2.71E+03 | 2.70E+03 | 2.70E+03 |
| | median | [3]2.70E+03 | [5]2.71E+03 | [1]**2.70E+03** | [6]2.72E+03 | [2]2.70E+03 | [4]2.71E+03 |
| | std | 8.08E−01 | 3.01E+00 | 1.32E+00 | 6.27E+00 | 1.50E+01 | 2.00E+00 |
| $f_{26}$ | max | 2.70E+03 | 2.80E+03 | 2.80E+03 | 2.80E+03 | 2.70E+03 | 2.70E+03 |
| | min | 2.70E+03 | 2.70E+03 | 2.80E+03 | 2.70E+03 | 2.70E+03 | **2.70E+03** |
| | median | [2]2.70E+03 | [3]2.70E+03 | [5]2.80E+03 | [6]2.80E+03 | [4]2.70E+03 | [1]**2.70E+03** |
| | std | 5.43E−02 | 2.20E+01 | 5.43E−03 | 3.53E+01 | 5.37E−01 | 6.50E−02 |
| $f_{27}$ | max | 3.10E+03 | 3.51E+03 | 4.43E+03 | 6.47E+03 | 3.53E+03 | 3.50E+03 |
| | min | **3.01E+03** | 3.24E+03 | 3.10E+03 | 3.57E+03 | 3.21E+03 | 3.10E+03 |
| | median | [2]3.10E+03 | [4]3.40E+03 | [5]3.82E+03 | [6]4.84E+03 | [3]3.31E+03 | [1]**3.10E+03** |
| | std | 3.38E+01 | 6.35E+01 | 3.51E+02 | 6.83E+02 | 6.46E+01 | 5.90E+01 |
| $f_{28}$ | max | 3.85E+03 | 4.27E+03 | 6.92E+03 | 6.65E+03 | 6.10E+03 | 5.39E+03 |
| | min | 3.56E+03 | 3.61E+03 | 3.76E+03 | 4.70E+03 | **3.01E+03** | 3.10E+03 |
| | median | [1]**3.69E+03** | [3]3.79E+03 | [6]5.43E+03 | [5]5.36E+03 | [4]4.52E+03 | [2]3.78E+03 |
| | std | 4.12E+01 | 9.33E+01 | 7.15E+02 | 4.61E+02 | 5.93E+02 | 3.61E+02 |
| $f_{29}$ | max | 2.79E+04 | 8.64E+06 | 2.93E+06 | 4.11E+07 | 1.36E+07 | 5.06E+03 |
| | min | 5.37E+03 | 4.26E+03 | **3.10E+03** | 4.81E+03 | 6.16E+05 | 3.56E+03 |
| | median | [5]1.58E+04 | [3]5.26E+03 | [1]**3.10E+03** | [4]1.54E+04 | [6]4.21E+06 | [2]4.02E+03 |
| | std | 5.14E+03 | 1.11E+06 | 3.78E+05 | 7.70E+06 | 2.83E+06 | 3.60E+02 |
| $f_{30}$ | max | 1.69E+04 | 3.75E+04 | 1.14E+05 | 3.74E+04 | 5.08E+05 | 7.66E+03 |
| | min | 6.05E+03 | 7.78E+03 | 1.22E+04 | 8.27E+03 | 6.26E+04 | **4.25E+03** |
| | median | [2]8.85E+03 | [5]1.56E+04 | [3]1.46E+04 | [4]1.51E+04 | [6]1.77E+05 | [1]**5.63E+03** |
| | std | 2.08E+03 | 6.08E+03 | 1.84E+04 | 6.58E+03 | 9.11E+04 | 7.38E+02 |

On $f_{24}$–$f_{27}$, the values in bold are better than those seemingly same values not in bold, because the digits after the second decimal place are omitted.

On the third hybrid group of 6 functions, WWO obtains the best median values on 4 functions ($f_{17}$, $f_{18}$, $f_{19}$ and $f_{21}$) and obtains the second best median values on 2 functions. Statistical tests show that the performance WWO is significantly different from other five algorithms on $f_{17}$ and $f_{18}$, and different from other four except IWO on $f_{19}$–$f_{22}$. Note in this group of hybrid functions, the variables are randomly divided into some subcomponents and then different basic functions are used for different subcomponents, which causes significant performance reduction of algorithms such as BBO and GSA, but WWO still remains as competitive performance as on the basic functions.

On the fourth composition group of 8 functions, WWO ranks first on 3 functions ($f_{26}$, $f_{27}$ and $f_{30}$), second on 2 functions ($f_{28}$ and $f_{29}$), third on 1 function ($f_{23}$) and fourth on 2 functions ($f_{24}$ and $f_{25}$). The relatively low performance of WWO on $f_{24}$ and $f_{25}$ is partially consistent with that on the subfunctions including $f_9$ and $f_{10}/f_{11}$, which cause that the composition functions also have a huge number of local optima.

In summary, the overall performance of WWO is the best among the six algorithms on the benchmark suite including unimodal, multimodal, hybrid, and composition functions. On some test functions (and their compositions) with too many local optima, the performance of WWO is not very satisfactory. This is mainly because we use a linearly decreasing population size for WWO in the experiments, and in the later iterations the number of solutions (waves) reduces to single digits and thus is difficult to jump out of local optima. We have also test the use of a fixed, relatively large population size (50 or 100) in WWO, which can effectively improve the performance of WWO on such test functions, but lose performance on many other test functions. In general, the population size reduction strategy is effective in improving the overall performance of WWO, but makes the algorithm ineffective in some special problems. However, we should compare the two strategies and select the better one for most real-world optimization problems.

Among the other five comparative algorithms, IWO exhibits the best performance on the whole suite. However, none of the algorithms can always be superior to others on all test functions. In fact, each algorithm has obtained the best median values on some functions: IWO ranks first on 7 functions, and BBO, GSA, HuS and BA respectively do so on 3, 6, 2 and 2 functions. That is, every algorithm has shown its strength and weakness on this benchmark suite, and we believe that this is the same case for various real-world problems. Thus, when selecting EAs for a new optimization problem, it is important to describe and quantify the boundary of effective algorithm performance in terms of the properties of problem instances by using objective measures and tools [42].

It should be noted that, the performance of WWO is not very competitive with those algorithms highly ranked in the CEC 2014 competition — most of such algorithms have used complicated search mechanisms such as hybrid operators, historical memory, replacement strategies, hyper-heuristic controllers, etc., and have their parameters fine-tuned for the benchmark suite. However, here our aim is just to test the performance of the basic WWO — with a quite simple framework and parameter setting — on the benchmark suite. It can be expected that, by introducing more sophisticated

**Table 6**
Sum of ranks of the six algorithms on CEC 2014 benchmark functions.

| | IWO | BBO | GSA | HuS | BA | WWO |
|---|---|---|---|---|---|---|
| Unimodal | 9 | 14 | 11 | 7 | 18 | 4 |
| Multimodal | 31 | 42 | 44 | 46 | 90 | 22 |
| Hybrid | 12 | 26 | 25 | 23 | 32 | 8 |
| Composition | 22 | 34 | 24 | 42 | 28 | 18 |
| Total | 74 | 116 | 104 | 118 | 168 | 52 |

**Table 7**
Statistical comparison between WWO and the other five algorithms.

| | IWO | | BBO | | GSA | | HuS | | BA | |
|---|---|---|---|---|---|---|---|---|---|---|
| | *p*-value | *h* | *p*-value | *h* | *p*-value | *h* | *p*-value | *h* | *p*-value | *h* |
| $f_1$ | 3.54E−15 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_2$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ | 1.17E−20 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_3$ | 3.56E−21 | $1^+$ | 2.34E−20 | $1^+$ | 3.56E−21 | $1^+$ | 9.50E−09 | $1^-$ | 3.56E−21 | $1^+$ |
| $f_4$ | 1.46E−15 | $1^+$ | 3.51E−19 | $1^+$ | 3.56E−21 | $1^+$ | 6.95E−17 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_5$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ | 2.13E−15 | $1^-$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_6$ | 1.04E−15 | $1^-$ | 1.01E−19 | $1^+$ | 3.55E−21 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_7$ | 1.12E−20 | $1^+$ | 2.75E−21 | $1^+$ | 1.00E+00 | 0 | 3.82E−08 | $1^+$ | 2.75E−21 | $1^+$ |
| $f_8$ | 3.04E−21 | $1^+$ | 3.04E−21 | $1^+$ | 5.30E−01 | 0 | 3.04E−21 | $1^+$ | 3.04E−21 | $1^+$ |
| $f_9$ | 2.02E−09 | $1^-$ | 8.66E−06 | $1^-$ | 3.56E−21 | $1^+$ | 1.06E−18 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{10}$ | 8.74E−18 | $1^+$ | 3.56E−21 | $1^-$ | 3.56E−21 | $1^+$ | 8.60E−14 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{11}$ | 4.21E−19 | $1^-$ | 1.77E−01 | 0 | 4.57E−21 | $1^+$ | 4.39E−01 | 0 | 3.56E−21 | $1^+$ |
| $f_{12}$ | 1.00E+00 | 0 | 1.00E+00 | 0 | 1.00E+00 | 0 | 1.00E+00 | 0 | 1.00E+00 | 0 |
| $f_{13}$ | 1.30E−01 | 0 | 3.64E−20 | $1^+$ | 2.40E−03 | $1^+$ | 1.40E−15 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{14}$ | 7.95E−01 | 0 | 2.52E−18 | $1^+$ | 3.66E−05 | $1^+$ | 1.37E−02 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{15}$ | 2.20E−03 | $1^+$ | 3.56E−21 | $1^+$ | 6.84E−01 | 0 | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{16}$ | 5.20E−01 | 0 | 2.96E−06 | $1^-$ | 3.56E−21 | $1^+$ | 3.54E−15 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{17}$ | 9.75E−12 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ | 1.65E−17 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{18}$ | 6.81E−21 | $1^+$ | 3.56E−21 | $1^+$ | 1.41E−12 | $1^+$ | 8.74E−18 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{19}$ | 4.06E−01 | 0 | 1.68E−16 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{20}$ | 7.14E−08 | $1^-$ | 3.14E−20 | $1^+$ | 3.14E−20 | $1^+$ | 3.56E−21 | $1^+$ | 1.89E−17 | $1^+$ |
| $f_{21}$ | 1.78E−01 | 0 | 1.17E−20 | $1^+$ | 8.61E−19 | $1^+$ | 9.35E−04 | $1^+$ | 3.74E−21 | $1^+$ |
| $f_{22}$ | 1.65E−07 | $1^-$ | 8.66E−09 | $1^+$ | 1.36E−20 | $1^+$ | 5.12E−19 | $1^+$ | 7.52E−21 | $1^+$ |
| $f_{23}$ | 8.66E−09 | $1^+$ | 7.90E−21 | $1^+$ | 1.00E+00 | 0 | 3.82E−20 | $1^+$ | 4.19E−05 | $1^-$ |
| $f_{24}$ | 1.40E−15 | $1^-$ | 1.16E−05 | $1^+$ | 3.56E−21 | $1^-$ | 2.12E−20 | $1^+$ | 3.56E−21 | $1^-$ |
| $f_{25}$ | 1.25E−18 | $1^-$ | 6.52E−12 | $1^+$ | 5.14E−23 | $1^-$ | 3.74E−21 | $1^+$ | 4.58E−10 | $1^-$ |
| $f_{26}$ | 1.86E−02 | $1^+$ | 1.66E−20 | $1^+$ | 3.56E−21 | $1^+$ | 2.52E−18 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{27}$ | 3.43E−01 | 0 | 5.92E−20 | $1^+$ | 1.66E−20 | $1^+$ | 3.56E−21 | $1^+$ | 6.53E−20 | $1^+$ |
| $f_{28}$ | 3.70E−14 | $1^-$ | 7.51E−01 | 0 | 3.83E−18 | $1^+$ | 1.01E−19 | $1^+$ | 1.17E−12 | $1^+$ |
| $f_{29}$ | 3.56E−21 | $1^+$ | 3.85E−19 | $1^+$ | 3.54E−15 | $1^-$ | 4.13E−21 | $1^+$ | 3.56E−21 | $1^+$ |
| $f_{30}$ | 1.01E−19 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ | 3.56E−21 | $1^+$ |

mechanisms, or integrating with powerful operators from other heuristic methods, WWO will also achieve considerable performance improvement.

## 5. Application in high-speed train scheduling

In this section we consider a high-speed train scheduling problem. There are total $I$ trains to pass through a high-speed passenger line, which is divided into $K$ sections by $K+1$ stations denoted by $S=\{0,1,\ldots,K\}$. According to the design speed of the line, the minimum travel time from station $k-1$ to $k$ is $\tau_k$. Each train $i$ has a sequence of stop stations, denoted by $S_i$, which is a subsequence of $S$; the minimum stop time of the train at each station $k\in S_i$ is $\tau_{i,k}$, and the expected total travel time is $\widehat{T}_i$. The problem is to find an optimal train timetable, i.e., to determine for each train $i$ the departure time $x_{i,0}$ from station 0, the travel time $x_{i,k}$ from station $k-1$ to $k$ $(1\le k\le K)$, and the stop time $y_{i,k}$ on station $k$ $(k\in S_i)$, such that the following objective function is minimized:

$$f=\sum_{i=1}^{I} T_i+\frac{1}{I}\sum_{i=1}^{I} D_i^2 \tag{11}$$

where $T_i$ is the real total travel time of train $i$, and $D_i$ is calculated as follows:

$$D_i=\begin{cases} T_i-\widehat{T}_i, & \text{if } T_i>\widehat{T}_i \\ 0, & \text{else} \end{cases} \tag{12}$$

For each train $i$, its arrival time $t^a_{i,k}$ and departure time $t^b_{i,k}$ on station $k$ can be respectively calculated as

$$t^a_{i,k}=x_{i,0}+\sum_{\kappa=1}^{k} x_{i,\kappa}+\sum_{\kappa\in S_i\wedge\kappa<k} y_{i,\kappa} \tag{13}$$

$$t^b_{i,k}=t^a_{i,k}+y_{i,k} \tag{14}$$

**Table 8**
The summary of the train scheduling problem instances.

| Instance | $I$ | $K$ | $n$ |
|---|---|---|---|
| #1 | 18 | 23 | 620 |
| #2 | 32 | 23 | 1056 |
| #3 | 36 | 33 | 1630 |
| #4 | 56 | 33 | 2505 |

Thus we have $T_i=(t^a_{i,K}-x_{i,0})$. The problem is also subject to the following constraints:

$$\tau_k\le x_{i,k}\le\widehat{x}_k,\ \forall i\in\{1\ldots I\},k\in S \tag{15}$$

$$\tau_{i,k}\le y_{i,k}\le\widehat{y}_k,\ \forall i\in\{1\ldots I\},k\in S_i \tag{16}$$

$$|t^a_{i,k}-t^a_{j,k}|\ge h_a,\ \forall i\ne j,i,j\in\{1\ldots I\},k\in S \tag{17}$$

$$|t^b_{i,k}-t^b_{j,k}|\ge h_b,\ \forall i\ne j,i,j\in\{1\ldots I\},k\in S \tag{18}$$

where $\widehat{x}_k$ is a predefined upper limit of travel time from station $k-1$ to $k$, $\widehat{y}_k$ is the upper limit of stop time at station $k$, $h_a$ is the minimum (safety) time interval between two trains entering the same station, and $h_b$ is that between two trains leaving the same station.

The constraints (17) and (18) are transformed into a penalty function as

$$g=\sum_{i\ne j,i,j\in\{1\ldots I\},k\in S} g^a_{i,j,k}+g^b_{i,j,k} \tag{19}$$

where

$$g^a_{i,j,k}=\begin{cases} h_a-|t^a_{i,k}-t^a_{j,k}| & \text{if } |t^a_{i,k}-t^a_{j,k}|<h_a \\ 0, & \text{else} \end{cases} \tag{20}$$

$$g^b_{i,j,k}=\begin{cases} h_b-|t^b_{i,k}-t^b_{j,k}| & \text{if } |t^b_{i,k}-t^b_{j,k}|<h_b \\ 0, & \text{else} \end{cases} \tag{21}$$

Therefore, the problem is an $n$-dimensional optimization problem where $n=I(K+1)+\sum_{i=1}^{I}|S_i|$, the components of each solution to the problem are bounded by Eq. (15) and (16), and the solution is evaluated by the penalized objective function:

$$f'=f+Mg \tag{22}$$

where $M$ is a large positive constant set to 10,000 in this paper.

We apply WWO to the train scheduling problem on two high-speed passenger lines in Southeast China: the Xiangtang(Nanchang)-Putian line and the Ningbo–Xiamen line, each of which with two instances: one for regular timetable scheduling and the other for spring festival timetable scheduling. Table 8 presents the basic information of the four problem instances. All times are measured in minutes, and for every new solution found by WWO, all of the components are rounded to the nearest integers (minutes).

WWO uses the same parameter settings described in the above section, and it is compared with the other five algorithms used in the above section and a chaotic PSO (CPSO) algorithm [43] that is previously employed by the company. The maximum NFE is set to

**Table 9**
The comparative results on the train scheduling instances.

| | | IWO | BBO | GSA | HuS | BA | CPSO | WWO |
|---|---|---|---|---|---|---|---|---|
| #1 | max | 4.155E+03 | 4.102E+03 | 4.168E+03 | 4.460E+03 | 4.275E+03 | 4.189E+03 | 4.099E+03 |
| | min | **4.088E+03** | **4.088E+03** | **4.088E+03** | 4.275E+03 | 4.092E+03 | 4.106E+03 | **4.088E+03** |
| | mean | [2]4.125E+03 | [2]4.095E+03 | [4]4.136E+03 | [6]4.320E+03 | [5]4.185E+03 | [3]4.129E+03 | [1]**4.093E+03** |
| | std | 1.540E+01 | 1.032E+01 | 1.767E+01 | 1.250E+02 | 6.933E+01 | 1.751E+01 | 6.424E+00 |
| #2 | max | 9.375E+03 | 9.393E+03 | 9.468E+03 | 9.375E+03 | 9.384E+03 | 9.303E+03 | 9.211E+03 |
| | min | 9.211E+03 | 9.189E+03 | 9.402E+03 | 9.189E+03 | 9.230E+03 | 9.205E+03 | **9.120E+03** |
| | mean | [5]9.306E+03 | [2]9.215E+03 | [7]9.452E+03 | [4]9.298E+03 | [6]9.323E+03 | [3]9.269E+03 | [1]**9.169E+03** |
| | std | 7.288E+01 | 1.333E+02 | 7.880E+01 | 1.006E+02 | 8.501E+01 | 4.671E+01 | 3.463E+01 |
| #3 | max | 1.226E+04 | 1.230E+04 | 1.368E+04 | 1.258E+04 | 1.215E+04 | 1.237E+04 | 1.196E+04 |
| | min | 1.196E+04 | **1.179E+04** | 1.326E+04 | 1.200E+04 | **1.179E+04** | 1.196E+04 | **1.179E+04** |
| | mean | [4]1.205E+04 | [3]1.197E+04 | [7]1.352E+04 | [6]1.217E+04 | [2]1.192E+04 | [5]1.210E+04 | [1]**1.186E+04** |
| | std | 1.551E+02 | 5.606E+02 | 7.370E+02 | 7.579E+02 | 3.183E+02 | 1.035E+02 | 4.631E+01 |
| #4 | max | 5.172E+04 | 2.496E+04 | 5.009E+06 | 8.495E+06 | 2.480E+04 | 6.160E+06 | 2.136E+04 |
| | min | 2.297E+04 | 2.180E+04 | 3.983E+04 | 3.026E+04 | 2.180E+04 | 2.220E+04 | **2.120E+04** |
| | mean | [4]3.154E+04 | [2]2.316E+04 | [6]3.809E+06 | [7]4.200E+06 | [3]2.335E+04 | [5]3.372E+06 | [1]**2.132E+04** |
| | std | 1.503E+04 | 5.302E+03 | 1.338E+06 | 3.263E+06 | 5.506E+03 | 4.254E+06 | 4.329E+01 |

100$n$ where $n$ is the instance size. Each algorithm has been run 60 times on each instance, and the computational results over the 60 runs are shown in Table 9.

As wee can see from the results, both the median, maximum and minimum objective values of WWO are always the best among the seven algorithms on all the four instances. On the simplest instance #1, only BBO obtains a very similar result to WWO; but with the increase of instance size, the performance advantage of WWO over the other algorithms becomes more and more obvious. Particularly, on the largest instance #4, the median and std values of CPSO, GSA and HuS are unreasonably large because the algorithms occasionally fail to obtain a feasible solution. Even if only considering the best objective values among the 60 runs, the efficiency of the schedules obtained by WWO is significantly better than the old CPSO on all the instances, which can lead to considerable economic benefits. Thus, we can conclude that among the seven methods, WWO is the most suitable for solving the considered train scheduling problem, or at least for the four significant real-world instances.

## 6. Conclusion

Just as biological genetics inspired the development of GAs and island biogeography inspired the development of BBO [11], in this paper we consider shallow water wave theory as the basis for the development of a new nature-inspired optimization method – WWO. The metaheuristic uses three operators: *Propagation* making high fitness waves search small areas and low fitness waves search large areas, *Refraction* for improving the diversity and thus reducing premature convergence, and *Breaking* for intensively exploiting the local area around a promising point.

We have compared WWO with five other popular EAs on the CEC 2014 test functions, the results of which show that the overall performance of WWO ranks first not only on the whole benchmark set, but also on each of the four subsets including unimodal, multimodal, hybrid, and composition functions. We have also used WWO to solve a high-speed train scheduling problem, which shows the feasibility and effectiveness of the metaheuristic in real-world applications.

The paper is a preliminary study opening up a wide range of possibilities for further improvement and extension. For example, currently we adopt a simple energy dissipation mechanism that linearly decreases the height of an unimproved wave from $h_{\max}$ to 0, and there is room for more complex, nonlinear dissipation models which may bring more performance improvement. Similar adaptive mechanisms may also apply to the wavelength reduction coefficient $\alpha$ and the breaking coefficient $\beta$.

In another aspect, the proposed WWO uses a default global topology where each wave can be attracted by the global best by refraction. An interesting direction is to apply various local topologies to diversify the modes of interactions amongst individuals, as that has been done in PSO [44] and EBO [45].

More abstractly, WWO can be regarded as a framework, where the wave-inspired operators can be implemented with different ways. It would be fruitful to design specific implementations of propagation, refraction, and breaking for a variety of problems, especially those combinatorial optimization problems. There is a wide range of avenues for future research.

## Acknowledgments

This work was supported by grants from National Natural Science Foundation of China under Grant nos. 61020106009, 61105073 and 61473263.

## References

[1] McCulloch WS, Pitts WH. A logical calculus for the ideas immanent in nervous activity. Bull Math Biophys 1943;5:115–33.
[2] Von Neumann J, Burks AW, et al. Theory of self-reproducing automata. Urbana: University of Illinois press; 1966.
[3] Adleman L. Molecular computation of solutions to combinatorial problems. Science 1994;266(5187):1021–4. http://dx.doi.org/10.1126/science.7973651.
[4] De Jong KA. Evolutionary computation: a unified approach. MIT Press Cambridge; 2006.
[5] Holland JH. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence. Cambridge, MA, USA: MIT Press; 1975.
[6] Fogel LJ, Owens AJ, Walsh MJ. Artificial intelligence through simulated evolution. John Wiley & Sons Chichester; 1967.
[7] Beyer H-G, Schwefel H-P. Evolution strategies – a comprehensive introduction. Nat Comput 2002;1(1):3–52. http://dx.doi.org/10.1023/A:1015059928466.
[8] Kennedy J, Eberhart R. Particle swarm optimization. In: IEEE international conference on neural networks, vol. 4; 1995. p. 1942–8. http://dx.doi.org/10.1109/ICNN.1995.488968.
[9] Mehrabian A, Lucas C. A novel numerical optimization algorithm inspired from weed colonization. Ecol Inf 2006;1(4):355–66. http://dx.doi.org/10.1016/j.ecoinf.2006.07.003.
[10] Shah-Hosseini H, Problem solving by intelligent water drops. In: IEEE congress on evolutionary computation; 2007. p. 3226–31. http://dx.doi.org/10.1109/CEC.2007.4424885.
[11] Simon D. Biogeography-based optimization. IEEE Trans Evol Comput 2008; 12(6):702–13. http://dx.doi.org/10.1109/TEVC.2008.919004.
[12] Yang X-S, Deb S. Cuckoo search via lévy flights. In: World congress on nature biologically inspired computing; 2009. p. 210–4. http://dx.doi.org/10.1109/NABIC.2009.5393690.
[13] Tan Y, Zhu Y. Fireworks algorithm for optimization. In: Tan Y, Shi Y, Tan K, editors. Advances in swarm intelligence, lecture notes in computer science, vol. 6145. Springer; 2010. p. 355–64. http://dx.doi.org/10.1007/978-3-642-13495-1_44.
[14] Yang X-S, Gandomi AH. Bat algorithm: a novel approach for global engineering optimization. Eng Comput 2012;29(5):464–83. http://dx.doi.org/10.1108/02644401211235834.
[15] Banks A, Vincent J, Phalp K. Natural strategies for search. Nat Comput 2009; 8(3):547–70. http://dx.doi.org/10.1007/s11047-008-9087-7.
[16] Cagnina LC, Esquivel SC, Coello CAC. A fast particle swarm algorithm for solving smooth and non-smooth economic dispatch problems. Eng Optim 2011;43(5):485–505. http://dx.doi.org/10.1080/0305215X.2010.497186.
[17] Boussaïd I, Chatterjee A, Siarry P, Ahmed-Nacer M. Biogeography-based optimization for constrained optimization problems. Comput Oper Res 2012;39(12):3293–304. http://dx.doi.org/10.1016/j.cor.2012.04.012.
[18] Boussaïd I, Lepagnot J, Siarry P. A survey on optimization metaheuristics. Inf Sci 2013;237(1):82–117. http://dx.doi.org/10.1016/j.ins.2013.02.041.
[19] Zheng Y-J, Ling H-F. Emergency transportation planning in disaster relief supply chain management: a cooperative fuzzy optimization approach. Soft Comput 2013;17(7):1301–14. http://dx.doi.org/10.1007/s00500-012-0968-4.
[20] Zheng Y-J, Chen S-Y, Ling H-F. Evolutionary optimization for disaster relief operations: a survey. Appl Soft Comput 2014. http://dx.doi.org/10.1016/j.asoc.2014.09.041.
[21] Zheng Y-J, Ling H-F, Wu X-B, Xue J-Y. Localized biogeography-based optimization. Soft Comput 2014. http://dx.doi.org/10.1007/s00500-013-1209-1.
[22] Zheng Y-J, Ling H-F, Shi H-H, Chen H-S, Chen S-Y. Emergency railway wagon scheduling by hybrid biogeography-based optimization. Comput Oper Res 2014;43(3):1–8. http://dx.doi.org/10.1016/j.cor.2013.09.002.
[23] Mei CC, Liu P. Surface waves and coastal dynamics. Annu Rev Fluid Mech 1993;25(1):215–40.
[24] Craik AD. The origins of water wave theory. Annu Rev Fluid Mech 2004; 36(1):1–28. http://dx.doi.org/10.1146/annurev.fluid.36.050802.122118.
[25] Huang H. Dynamics of surface waves in coastal waters: wave-current-bottom interactions. Springer; 2009.
[26] Phillips O. On the dynamics of unsteady gravity waves of finite amplitude. J Fluid Mech 1960;9(2):193–217.
[27] Hasselmann K. On the non-linear energy transfer in a gravity-wave spectrum. J Fluid Mech 1962;12(15):481–500.
[28] Zakharov VE, L'vov VS, Falkovich G. Kolmogorov spectra of turbulence I, wave turbulence. Berlin: Springer-Verlag; 1992.
[29] Kelland P. On the theory of waves. Part II. Trans R Soc Edinb 1844; 15(01):101–44.
[30] Rayleigh L. On waves. Philos Mag 1876;5(1):257–79.
[31] The WAMDI Group. The WAM model — a third generation ocean wave prediction model. J Phys Oceanogr 1988;18(12):1775–810 doi:10.1175/1520-0485(1988)018 < 1775:TWMTGO > 2.0.CO;2.
[32] Hasselmann W, K, Sell DB, Ross PM. A parametric wave prediction model. J Phys Oceanogr 1976;6(2):200–28. doi:10.1175/1520-0485(1976)006 < 0200: APWPM > 2.0.CO;2.
[33] The SWAMP Group, Sea wave modelling project. An intercomparison study of wind wave prediction models, part 1: principal results and conclusions. In: Ocean wave modelling, New York: Plenum Press; 1985. p. 3–153.

[34] Booij N, Ris R, Holthuijsen LH. A third-generation wave model for coastal regions: 1. Model description and validation. J Geophys Res 1999;104 (C4):7649–66.
[35] Durran D. Numerical methods for wave equations in geophysical fluid dynamics. New York: Springer-Verlag; 1999.
[36] Clamond D. On the lagrangian description of steady surface gravity waves. J Fluid Mech 2007;589:433–54. http://dx.doi.org/10.1017/S0022112007007811.
[37] Kennedy J. Bare bones particle swarms. In: IEEE swarm intelligence symposium; 2003. p. 80–7. http://dx.doi.org/10.1109/SIS.2003.1202251.
[38] Brest J, Sepesy Maucec M. Population size reduction for the differential evolution algorithm. Appl Intell 2008;29(3):228–47. http://dx.doi.org/10.1007/s10489-007-0091-x.
[39] Liang JJ, Qu BY, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization, Tech. Rep. 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China; 2014.
[40] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179(13):2232–48. http://dx.doi.org/10.1016/j.ins.2009.03.004.
[41] Oftadeh R, Mahjoob M, Shariatpanahi M. A novel meta-heuristic optimization algorithm inspired by group hunting of animals: hunting search. Comput Math Appl 2010;60(7):2087–98. http://dx.doi.org/10.1016/j.camwa.2010.07.049.
[42] Smith-Miles K, Baatar D, Wreford B, Lewis R. Towards objective measures of algorithm performance across instance space. Comput Oper Res 2014;45: 12–24. http://dx.doi.org/10.1016/j.cor.2013.11.015.
[43] Gao L, Ren P, Li N. Optimal scheduling based on CPSO for high-speed passenger trains. J Northeast Univ: Nat Sci 2007;28(2):176–9 (192).
[44] Kennedy J, Mendes R. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. IEEE Trans Syst Man Cybern, Part C 2006;36 (4):515–9. http://dx.doi.org/10.1109/TSMCC.2006.875410.
[45] Zheng Y-J, Ling H-F, Xue J-Y. Ecogeography-based optimization: enhancing biogeography-based optimization with ecogeographic barriers and differentiations. Comput Oper Res 2014;50:115–27. http://dx.doi.org/10.1016/j.cor.2014.04.013.