Contents lists available at ScienceDirect

Computers and Structures

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

Water Evaporation Optimization: A novel physically inspired optimization algorithm

A. Kaveh [*], T. Bakhshpoori

Centre of Excellence for Fundamental Studies in Structural Engineering, School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

ARTICLE INFO

Article history:
Received 3 August 2015
Accepted 12 January 2016
Available online 26 February 2016

Keywords:
Metaheuristics
Water Evaporation Optimization
Molecular dynamics simulations
Global optimization
Global search
Local search

ABSTRACT

In this paper a novel physically inspired non-gradient algorithm is developed for solution of global optimization problems. The algorithm being called Water Evaporation Optimization (WEO) mimics the evaporation of a tiny amount of water molecules on the solid surface with different wettability which can be studied by molecular dynamics simulations. WEO is tested and analyzed in comparison to other existing methods on three sets of continuous test problems, a set of 17 benchmark unconstrained functions (consisting of three types of functions: unimodal, multimodal, and shifted and rotated functions), a set of 13 classical benchmark constraint functions, and three benchmark constraint engineering problems, reported in the specialized literature. The results obtained indicate that the proposed technique is highly competitive with other efficient well-known metaheuristics. The features used in WEO are analyzed and its potential implications for real size constrained engineering optimization problems are discussed in details.

1. Introduction

Metaheuristic algorithms are a widely studied field in artificial intelligence in terms of: applying to a wide variety of combinatorial optimization problems, improving the current algorithms, and generating novel ones. The primary reason for this is their ability to reach near global optimums in a reasonable time for non-smooth problems in which gradient information are not available or not reliable.

In the recent decades, many metaheuristics with different philosophy and characteristics are developed and applied to a wide range of fields which can be categorized into three separate classes in terms of how they have been inspired [1]:

(i) Evolutionary algorithms. These methods use techniques that imitate natural evolution. Of these, John Holland's Genetic Algorithm (GA) [2], and Rechenberg and Schwefel's Evolutionary strategies (ES) [3] enjoyed more attention.

(ii) Physical algorithms. These methods inspired by the physical laws. Of these, Simulated Annealing (SA) algorithm of Kirkpatrick et al. [4] mimics the annealing process used in materials science, Tabu Search (TS) method of Glover [5] utilizes a short term memory of the specific changes of recent moves within the search space and preventing future moves from undoing those changes, and Harmony Search (HS) algorithm of Geem et al. [6] that mimics the evolution of a harmony relationship between several sound waves of differing frequencies when played simultaneously, have been accepted as most prevalent algorithms.

(iii) Swarm algorithms. These methods imitate the processes of decentralized, self-organized systems, which can be either natural or artificial in nature. Of these, Ant Colony Optimization (ACO) algorithm of Dorigo et al. [7] which follows the processes of an ant colony searching for food, Kennedy and Eberhart's Particle Swarm Optimization (PSO) algorithm [8] that mimics animal flocking behaviors, and Karaboga's Artificial Bee Colony (ABC) algorithm [9] that is inspired by the food foraging behavior of honey bee swarms, considered as prevalent swarm algorithms.

In recent years many novel algorithms are developed and applied to different multidisciplinary optimization problems which can be categorized in one of the three classes mentioned in the previous paragraph. Some of these methods are listed at the following: Glowworm Swarm Optimization (GSO) [10], Firefly Algorithm (FA) [11], Monkey Search (MS) [12], Cuckoo Search (CS) [11], Bat Algorithm (BA) [11], Krill Herd (KH) Algorithm [13], Grey Wolf Optimizer (GWO) [14], Bird Mating Optimizer (BMO) [15], and Social Spider Optimization (SSO-C) [16] which can be

[*] Corresponding author. Tel.: +98 21 77240104; fax: +98 21 77240398.
E-mail address: alikaveh@iust.ac.ir (A. Kaveh).

http://dx.doi.org/10.1016/j.compstruc.2016.01.008
0045-7949/© 2016 Elsevier Ltd. All rights reserved.

considered as swarm algorithms. Intelligent Water Drops (IWD) algorithm [17], Imperialist Competitive Algorithm (ICA) [18], League Championship Algorithm (LCA) [19], Gravitational Search Algorithm (GSA) [20], Galaxy-based Search Algorithm (GbSA) [21], Spiral Optimization (SO) [22], Teaching–Learning-Based Optimization (TLBO) [23], Big-Bang Big-Crunch (BB-BC) algorithm [24], Water Cycle Algorithm (WCA) [25], Golden Ball (GB) algorithm [26], and Group Counseling Optimizer (GCO) [27] which can be considered as physical algorithms.

In the past decade, the first author and his students developed three novel physical algorithms and one novel swarm based algorithm besides many successful modified and hybridized algorithms [28–30]. These novel studies contain Charged System Search algorithm (CSS) algorithm [31], Ray Optimization algorithm (RO) [31], Dolphin echolocation optimization (DEO) [32] and Colliding Bodies Optimization (CBO) algorithm [33]. A comprehensive review on these methods and their applications can be found in [34,35].

As it is clear generating novel metaheuristic algorithms is an interesting subject in the field of optimization. The main objective of this paper is to develop a new physically based metaheuristic to solve global optimization problems. This new metaheuristic is a multiple population based algorithm, and it is called as Water Evaporation Optimization (WEO).

WEO mimics the evaporation of a tiny amount of water molecules adhered on a solid surface with different wettability which can be studied by molecular dynamics simulations. The evaporation of water is very important in biological and environmental science [36,37]. Based on the molecular dynamics simulations [38] it is well-known that, as the surface changed from hydrophobicity to hydrophility, the evaporation speed does not show a monotonically decrease from intuition, but increases first, and then decreases after reaching a maximum value. When the surface wettability of the substrate is not high enough, the water molecules accumulate into the form of a sessile spherical cap. The predominant factor that affects the evaporation speed is the geometry shape of the water congregation. Meanwhile when the surface wettability of the substrate is high enough, the water molecules spread to a monolayer and the geometric factor no longer affects much and the energy barrier provided by the substrate instead geometry shape, affects the evaporation speed.

Based on the previous paragraph the reader can see a fine analogy between this type of water evaporation phenomena and a population based metaheuristic algorithm. Water molecules are considered as algorithm individuals. Solid surface or substrate with variable wettability is reflected as the search space. Decreasing the surface wettability (substrate changed from hydrophility to hydrophobicity) reforms the water aggregation from a monolayer to a sessile droplet. Such a behavior is consistent with how the layout of individuals changes to each other as the algorithm progresses. And the decreasing wettability of the surface can represent the decrease of objective function for a minimizing optimization problem. Evaporation flux rate of the water molecules is considered as the most appropriate measure for updating the individuals which its pattern of change is in good agreement with the local and global search ability of the algorithm and can help us to develop the WEO with significantly well converged behavior and simple algorithmic structure.

WEO is tested and analyzed on three sets of test problems, a set of 17 benchmark unconstrained functions consisting of three types of functions: unimodal, multimodal, and shifted and rotated functions, a 13-fold classical benchmark constraint problems, and three benchmark constraint engineering problems reported in the specialized literature. The most effective available state-of-the-art metaheuristic optimization methods based on the author's knowledge are used here as the basis of the comparisons. The optimization results demonstrate the efficiency and competitive performance of the WEO.

The rest of this paper is organized as follows. Section 2 deals with analogy between WEO and the evaporation of a tiny amount of water on the solid surface with varying wettability which can be inferred by molecular dynamics simulation. Section 3 develops the novel proposed WEO algorithm in detail. Section 4 investigates the parameter settings and the search behavior of the WEO in depth and experimentally validates the WEO and compared to the most effective available state-of-the-art metaheuristic optimization methods on three test cases. Conclusions are derived in Section 5.

2. Analogy between water evaporation and population-based optimization

2.1. Molecular dynamics simulations for water evaporation

The evaporation of water is very important in biological and environmental science. The water evaporation from bulk surface such as a lake or a river is different from evaporation of water restricted on the surface of solid materials. Water evaporation of bulk surface is a classical topic and has been studied for long time [37,39] and is not the topic of this study. Recent works show that the nanoscale confined water and water restricted on the surface of solid materials is ubiquitous in nature [40], and this type of water evaporation is essential in the macroscopic world such as the water loss through the surface of soil [41]. Lee et al. [42] have used scanning electric microscope to study the evaporation efficiency of micro-scaled water droplet from nanoporous micro cantilevers of various hydrophobicity and stated that the dynamics of water evaporation between hydrophobic and hydrophilic conditions are very different. Wang et al. [38] presented Molecular Dynamics (MD) simulations on the evaporation of nanoscale water aggregation on a solid substrate with different surface wettability at room temperature to find out how the surface wettability affects the evaporation of the tiny water aggregation. At the following their simulation result is outlined which is considered as an effective essence to develop the present WEO algorithm.

MD simulations were carried out in a box with predefined dimensions. The substrate contained equal positive and negative charges, and is neutral. By changing the value of charge (q), a substrate with tunable surface wettability can be obtained. Initially, the fixed number of water molecules was piled upon the substrate in a water cube form as shown in Fig. 1(a). In the simulation, q is sampled from 0 to 0.7e with an increment of 0.1e. The simulations show that the water spreads smoothly on the substrate when $q \geqslant 0.4e$ (Fig. 1(c)). When $q < 0.4e$, the water shrinks gradually into a sessile droplet like a spherical cap as q decreases (Fig. 1(b)). In this phase, the contact angle can be affected by the amount of

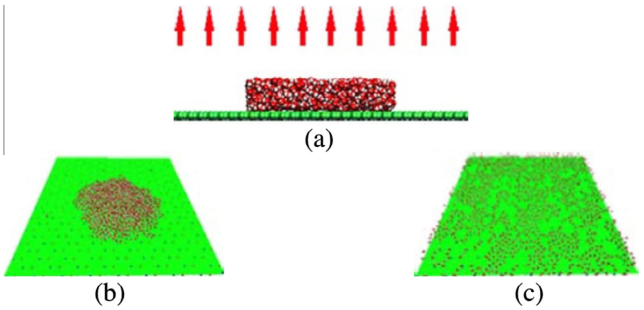

Fig. 1. (a) Side view of the initial system; (b) snapshot of water on the substrate with low wettability ($q = 0e$); (c) snapshot of water on the substrate with high wettability ($q = 0.7e$) [38].

aqueous [43], nearby liquid molecules [44], and the surface wettability. However, the contact angle (θ) can be used only as a phenomenology criterion of surface wettability and it can be consistent with the experimental results of relatively small amount of water.

The evaporation speed of the water layer can be described by the evaporation flux which is defined as the average number of the water molecules entering the accelerating region (the upward arrow denoted in Fig. 1(c)) from the substrate per nanosecond. Counter to intuition, the evaporation flux does not decrease monotonically as q increases. Actually the evaporation flux first increases as q increases when $q < 0.4$e; then the evaporation flux reaches its maximum around $q = 0.4$e; when $q \geqslant 0.4$e, the evaporation flux decreases as q increases.

To analyze this unusual variation of the evaporation flux, Wang et al. [38] assumed that the evaporation flux $J(q)$ can be denoted as a product of the aggregation probability of a water molecule in the interfacial liquid–gas surface and the escape probability of such surficial water molecule:

$$J(q) \propto P_{geo}(\theta(q))P_{ener}(E) \tag{1}$$

Here, $P_{geo}(\theta)$ is the probability for a water molecule on the liquid–gas surface, which is a geometry related factor. With respect to molecular dynamics simulation, $P_{geo}(\theta)$ is defined as the ratio of the number of surficial water molecules, to the total number of all condensed water molecules and is calculated as:

$$P_{geo}(\theta) = P_0\left(\frac{2}{3} + \frac{\cos^3\theta}{3} - \cos\theta\right)^{-2/3}(1 - \cos\theta) \tag{2}$$

where P_0 is a constant function of water molecule diameter and total volume of molecules. It should be noted that this probability is obtained for $q < 0.4$e in which the water molecules shrinks gradually into a sessile droplet like a spherical cap with contact angle θ to the substrate. For more detail, the reader can refer to MD simulations conducted by Wang et al. [38]. $P_{ener}(E)$ is the escape probability of a surficial water molecule. $E = E_{WW} + E_{sub}(q)$ is the average interaction energy exerted on the surficial water molecules, E_{ww} is the energy provided by the neighboring water molecules; $E_{sub}(q)$ represents the interaction energy from the substrate, mainly provided by the electrical charge q assigned on the substrate.

The relationship between the assigned charge q and the contact angle of the water droplet is shown in Fig. 2(a). The contact angle of the water droplet θ decreases as q increases and reaches 0° when $q = 0.4$e. When $q < 0.4$e, most of the surficial water molecules are relatively far from the substrate. According to Fig. 2(b), the energy E_{sub} provided by the substrate does not change much, and its variation is negligible if compared to the E_{sub} of $q \geqslant 0.4$e. At the same time, the E_{ww} provided by the neighboring water molecules almost keeps constant in simulation. Hence, for $q < 0.4$e, the escape probability of a surficial water molecule ($P_{ener}(E)$) is nearly a constant. Therefore the evaporation flux (Eq. (1)) will be updated as follows in which J_0 is constant equal to 1.24 ns^{-1}.

$$J(\theta) = J_0 P_{geo}(\theta), \quad q < 0.4 \tag{3}$$

For $q \geqslant 0.4$e, the adhered water forms a flat single-layer molecule sheet with only a few water molecules overlapping upon it, and the shape of the tiny water aggregation does not change much with different q. According to the definition of $P_{geo}(\theta)$, all the water molecules are on the surface layer now, therefore $P_{geo}(\theta) = 1$. According to the thermal dynamics, for the system under the NVT ensemble (NVT ensemble indicates a canonical ensemble representing possible states of a mechanical system in thermal equilibrium), the probability for a free molecule to gain kinetic energy more than E_0 is proportional to $\exp\left(-\frac{E_0}{K_B T}\right)$ [45]. Based on the MD simulations the evaporation flux decreases almost exponentially with respect to E_{sub}. Therefore the evaporation flux (Eq. (1)) will be updated as follows in which T is the room temperature and K_B is the Boltzman constant [45].

$$J(q) = \exp\left(-\frac{E_{sub}}{K_B T}\right), \quad q \geqslant 0.4e \tag{4}$$

2.2. Inspiration of the WEO algorithm

Based on the previous subsection the reader can see a fine analogy between this type of water evaporation phenomena and a population based metaheuristic algorithm, if he or she notes their MD simulations results from end to beginning. It should be noted that this analogy is stated for a minimization problem. Water molecules can be considered as algorithm individuals. Solid surface or substrate with variable wettability is reflected as the search space. Decreasing the surface wettability (substrate changed from hydrophility to hydrophobicity) reforms the water aggregation from a monolayer to a sessile droplet. Such a behavior is in coincidence with how the layout of the algorithm individuals changes to each other as the algorithm progresses. Furthermore, decreasing wettability of the surface (decreasing q from 0.7e to 0.0e) can represent the reduction of the objective function for a minimization optimization problem as the algorithm progresses. Evaporation flux variation of the water molecules can be considered as the most appropriate measure for updating the individuals which is in a good agreement with the local and global search ability of the

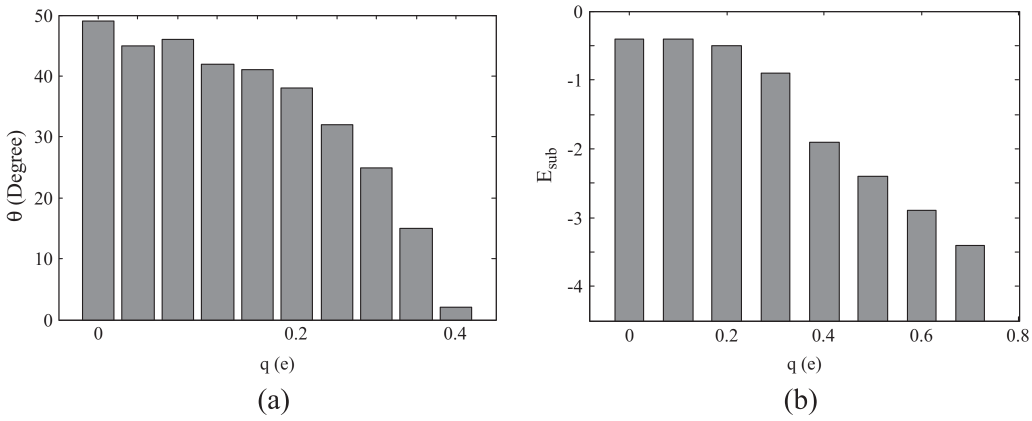

Fig. 2. (a) The contact angle θ of the water droplet with different assigned charge q; (b) the interaction energy exerted on the outermost-layer water by the substrate (E_{sub}) with different assigned charge q [38].

algorithm, and can help us to develop the WEO with significantly well converged behavior and simple algorithmic structure.

The evaporation flux is considered as a measure for determining the probability of updating the individuals of the algorithm that reaches its maximum around $q = 0.4e$. This situation is considered until the algorithm reaches the middle of the optimization process. In other words, WEO updates the individuals with the probability based on Eq. (4) (this probability is named Monolayer Evaporation Probability or MEP) until reaches to half the number of function evaluations. This first phase can be considered as the global search ability of the algorithm. After this phase, individuals will be updated with the probability based on Eq. (3) (which is named as Droplet Evaporation Probability or DEP). This phase can be considered as the local search ability of the algorithm. These two phases are introduced extensively in the following, and then the updating mechanism of individuals are introduced.

2.2.1. Monolayer evaporation phase

In the monolayer evaporation phase we can estimate the Eq. (4) with a simple exponential function of the substrate interaction energy: $\exp(E_{sub})$. As mentioned before, in this phase ($q > 0.4e$), as q increases, the substrate will have more energy and as a result we will have less evaporation. Let us consider -0.5 and -3.5 as the maximum (E_{max}) and minimum (E_{min}) values of E_{sub}, respectively, for each tth iteration of the algorithm until half the number of algorithm iterations. These values are based on Fig. 2(b). Fig. 3 shows the monolayer evaporation probability for various values of substrate energy between -3.5 and -0.5.

In each iteration, the objective function of individuals Fit_i^t is scaled to the interval $[-3.5, -0.5]$ and represents the corresponding $E_{sub}(i)^t$ inserted to each individual (substrate energy vector), via the following scaling function:

$$E_{sub}(i)^t = \frac{(E_{max} - E_{min}) \times (Fit_i^t - \mathrm{Min}(Fit))}{(\mathrm{Max}(Fit) - \mathrm{Min}(Fit))} + E_{min} \tag{5}$$

where Min and Max are the minimum and maximum functions, respectively. After generating the substrate energy vector, the Monolayer Evaporation Probability matrix (MEP) is constructed by the following equation:

$$MEP_{ij}^t = \begin{cases} 1 & if\ rand_{ij} < \exp(E_{sub}(i)^t) \\ 0 & if\ rand_{ij} \geqslant \exp(E_{sub}(i)^t) \end{cases} \tag{6}$$

where MEP_{ij}^t is the updating probability for the jth variable of the ith individual or water molecule in the tth iteration of the algorithm. In this way an individual with better objective function (considering the minimization problem) is more likely to remain unchanged in the search space. In detail we can say that in each iteration the best and worst candidate solutions will be updated by the probability equal to $\exp(-3.5) = 0.03$ and $\exp(-0.5) = 0.6$, respectively. In other words we can consider these values as minimum (MEP_{min}) and maximum (MEP_{max}) values of monolayer evaporation probability. Our simulation results shows that considering $MEP_{min} = 0.03$ and $MEP_{max} = 0.6$ based on the simulation results (Fig. 2(b)) is logical. However, these values can be considered as the first two parameters of the algorithm.

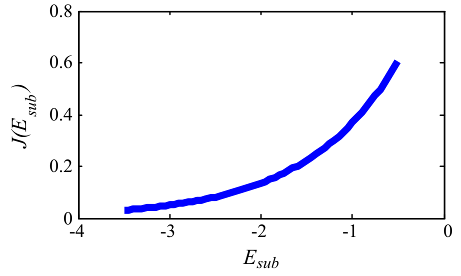

Fig. 3. Monolayer evaporation flux with different substrate energy for the WEO.

2.2.2. Droplet evaporation phase

In the droplet evaporation phase, using Eqs. (2) and (3) the evaporation flux is as the following:

$$J(\theta) = J_0 P_0 \left(\frac{2}{3} + \frac{\cos^3\theta}{3} - \cos\theta\right)^{-2/3} (1 - \cos\theta) \tag{7}$$

where J_0 and P_0 are constant values. As it is mentioned before, in this phase ($q < 0.4e$), since q is smaller, the contact angle is greater and as a result we will have less evaporation. According to Fig. 2(a) the maximum and minimum values of contact angle are 50° and 0°, respectively. Based on the MD simulations results, the variation of the evaporation flux perfectly fitted to the experimental results in the range $20° < \theta < 50°$. It can be interpreted that for $\theta < 20°$ the water droplet is no longer observed like a perfect sessile spherical cap. Fig. 4(a) and (b) illustrate this evaporation flux functions neglecting the constant values J_0 and P_0 for various contact angles between $0° < \theta < 50°$ and $20° < \theta < 50°$, respectively.

Our simulation results show that considering contact angle between $20° < \theta < 50°$ is quite suitable for WEO. Based on Fig. 4 (b), the maximum value for droplet evaporation probability is 2.6. Considering $J_0 \times P_0$ equal to $\frac{1}{2.6}$ for limiting the upper bound of droplet evaporation probability to 1, and considering -20 and -50 as the maximum (θ_{max}) and minimum (θ_{min}) values of contact angle, the droplet evaporation probability for various contact angles between $-50° < \theta < -20°$ is shown in Fig. 5. For all iterations in the second half of the algorithm, the objective function of individuals Fit_i^t is scaled to the interval $[-50°, -20°]$ via the following scaling function which represents the corresponding contact angle $\theta(i)^t$ (contact angle vector):

$$\theta(i)^t = \frac{(\theta_{max} - \theta_{min}) \times (Fit_i^t - \mathrm{Min}(Fit))}{(\mathrm{Max}(Fit) - \mathrm{Min}(Fit))} + \theta_{min} \tag{8}$$

where the Min and Max are the minimum and maximum functions. Such an assumption is consistent with MD simulations as depicted in Fig. 2(a) and results in a good performance of the WEO. Negative values have no effect on computations (cosine is an even function). In this way, the best and worst individuals have the smaller and bigger updating probability like the evaporation speed of a droplet on a substrate with less ($q = 0.0e$) and more ($q = 0.4e$) wettability, respectively. In other words we can have the droplet evaporation probability matrix with minimum (DEP_{min}) and maximum (DEP_{max}) values of droplet evaporation probability equal to 0.6 and 1, respectively as shown in Fig. 5. Our algorithm performance evaluation results show that these values are suitable. However these parameters can be considered as the next two parameters of the algorithm.

After generating contact angle vector $\theta(i)^t$, the Droplet Evaporation Probability matrix (DEP) is constructed by the following equation:

$$DEP_{ij}^t = \begin{cases} 1 & if\ rand_{ij} < J(\theta_i^{(t)}) \\ 0 & if\ rand_{ij} \geqslant J(\theta_i^{(t)}) \end{cases} \tag{9}$$

where DEP_{ij}^t is the updating probability for the jth variable of the ith individual or water molecule in the tth iteration of the algorithm.

2.2.3. Updating water molecules

In the MD simulations, the number of evaporated water molecules in the entire simulation process is considered negligible compared to the total number of the water molecules resulting in a

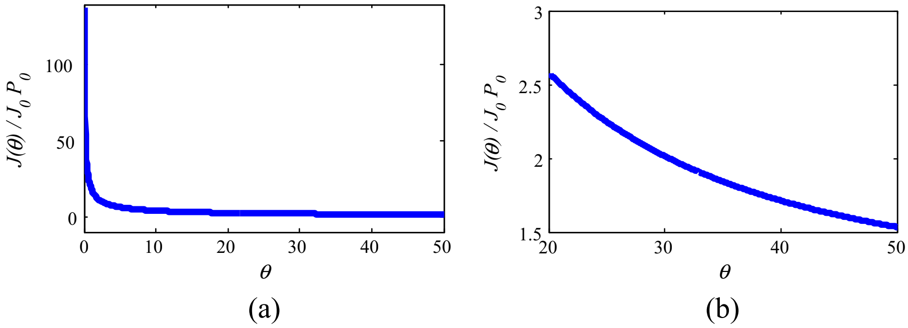

Fig. 4. Droplet evaporation flux based on the MD simulations.

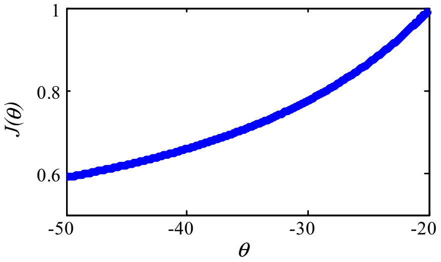

Fig. 5. Droplet evaporation flux with different contact angles considered for the WEO.

constant total number of molecules. In the WEO also the number of algorithm individuals or number of the water molecules (nWM) is considered constant, in all tth algorithm iterations. nWM is the algorithm parameter like other population based algorithms. t is the number of current iteration. Considering a maximum value for algorithm iterations (t_{max}) is essential for WEO to determine the evaporation phase of the algorithm, and also use as the stopping criterion. Such stopping criterion is utilized in many of optimization algorithms. When a water molecule is evaporated it should be renewed. Updating or evaporation of the current water molecules is made with the aim of improving objective function. The best strategy for regenerating the evaporated water molecules is using the current set of water molecules ($WM^{(t)}$). In this way a random permutation based step size can be considered for possible modification of individuals as:

$$S = rand \cdot (WM^{(t)}[permute1(i)(j)] - WM^{(t)}[permute2(i)(j)]) \tag{10}$$

where $rand$ is a random number in [0, 1] range, $permute1$ and $permute2$ are different rows permutation functions. i is the number of water molecule, j is the number of dimensions of the problem at hand. The next set of molecules ($WM^{(t+1)}$) is generated by adding this random permutation based step size multiplied by the corresponding updating probability (monolayer evaporation and droplet evaporation probability) and can be stated mathematically as:

$$WM^{(t+1)} = WM^{(t)} + S \times \begin{cases} MEP^{(t)} & t \leqslant t_{max}/2 \\ DEP^{(t)} & t > t_{max}/2 \end{cases} \tag{11}$$

Each water molecule is compared and replaced by the corresponding renewed molecule based on the objective function. It should be noted that random permutation based step size can help us in two aspects. In the first phase, water molecules are more far from each other than the second phase. In this way the generated permutation based step size will guarantee global and local search capability in each phase. The random part will guarantee the algorithm to be sufficiently dynamic. It should also be noted that these two aspects are guaranteed by considering two specific evaporation probability mechanisms for each phase. In both evaporation phase, as it is clear from Figs. 3 and 5, good water molecules are renewed locally (with less evaporation probability) while bad quality molecules are renewed globally (with more evaporation probability).

3. The proposed WEO algorithm

In this section, we present the proposed WEO algorithm. The flowchart of WEO is illustrated in Fig. 6 and the steps involved are as follows:

Step 1: Initialization

Algorithm parameters are determined in the first step. These parameters are the number of water molecules (nWM), maximum number of algorithm iterations (t_{max}), minimum (MEP_{min}) and maximum (MEP_{max}) values of monolayer evaporation probability, minimum (DEP_{min}) and maximum (DEP_{max}) values of droplet evaporation probability. As mentioned before, evaporation probability parameters are determined efficiently for WEO based on the MD simulations results. The initial positions of all water molecules are generated randomly within the n-dimensional search space ($WM^{(0)}$), and are evaluated based on the objective function of the problem at hand.

Step 2: Generating water evaporation matrix

Every water molecule follows the evaporation probability rules specified for each phase of the algorithm based on the Eqs. (6) and (9). For $t \leqslant t_{max}/2$, water molecules are globally evaporated based on the monolayer evaporation probability (MEP) rule (Eq. (6)); for $t > t_{max}/2$, evaporation occurs based on the droplet evaporation probability (DEP) rule (Eq. (9)). It should be noted that for generating monolayer and droplet evaporation probability matrices, it is necessary to generate the correspondent substrate energy vector (Eq. (5)) and contact angle vector (Eq. (8)), respectively.

Step 3: Generating random permutation based step size matrix

A random permutation based step size matrix is generated according to Eq. (10).

Step 4: Generating evaporated water molecules and updating the matrix of water molecules

The evaporated set of water molecules $VM^{(t+1)}$ is generated by adding the product of step size matrix and evaporation probability matrix to the current set of molecules $VM^{(t)}$ according to Eq. (11).

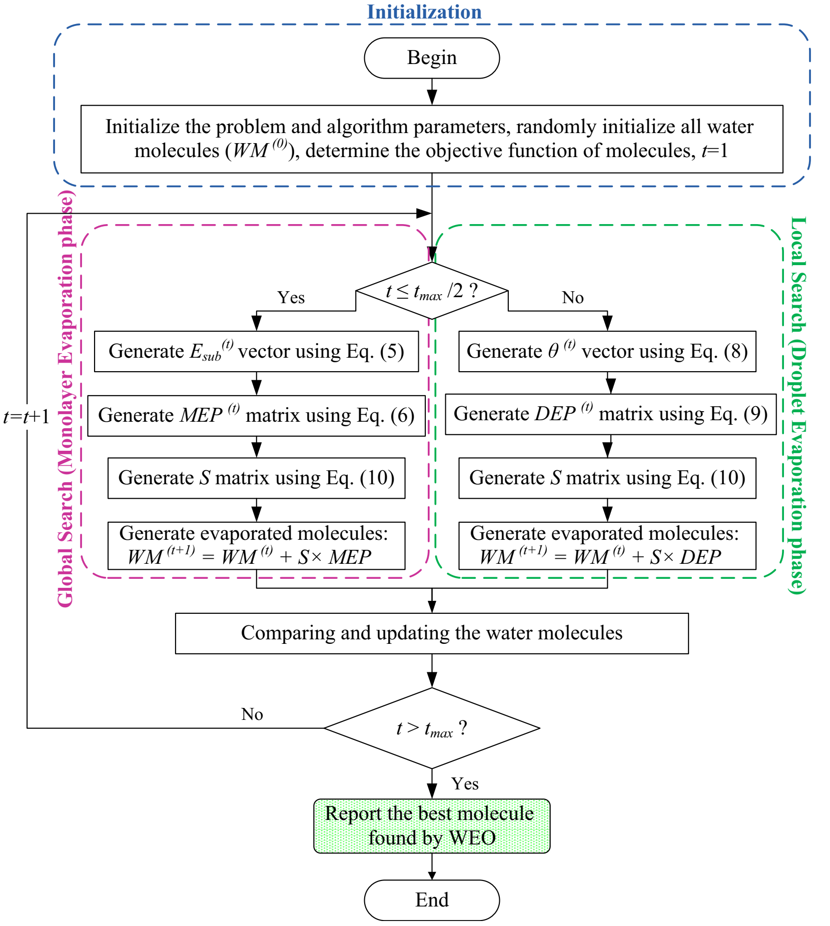
Initialization
Begin
Initialize the problem and algorithm parameters, randomly initialize all water molecules ($WM^{(0)}$), determine the objective function of molecules, t=1
Global Search (Monolayer Evaporation phase)
Local Search (Droplet Evaporation phase)
$t \le t_{max}/2$?
Yes
No
Generate $E_{sub}^{(t)}$ vector using Eq. (5)
Generate $MEP^{(t)}$ matrix using Eq. (6)
Generate S matrix using Eq. (10)
Generate evaporated molecules: $WM^{(t+1)} = WM^{(t)} + S \times MEP$
Generate $\theta^{(t)}$ vector using Eq. (8)
Generate $DEP^{(t)}$ matrix using Eq. (9)
Generate S matrix using Eq. (10)
Generate evaporated molecules: $WM^{(t+1)} = WM^{(t)} + S \times DEP$
t=t+1
Comparing and updating the water molecules
$t > t_{max}$?
No
Yes
Report the best molecule found by WEO
End

Fig. 6. Flowchart of the WEO algorithm.

These molecules are evaluated based on the objective function. For the molecule i ($i = 1, 2, \ldots, nWM$) if the newly generated molecule is better than the current one, the latter should be replaced. Return the best water molecule as the output of the algorithm.

Step 5: Terminating condition check

If the number of iteration of the algorithm (t) becomes larger than the maximum number of iterations (t_{max}), the algorithm terminates. Otherwise go to Step 2.

4. Test experiments

4.1. Experimental results and discussion on unconstrained problems

4.1.1. Benchmark and experiments settings

It is common in optimization frameworks to benchmark the performance of new algorithms in comparison with other algorithms on a set of mathematical functions with known global optima so-called global optimization problems or real parameter optimization problems. Many optimization algorithms have been proposed some of which were reviewed in the introduction section. Therefore some inevitable subjective selection of algorithms is needed for evaluating the WEO. Probably the most frequent ways to select algorithms for comparison are [46]: (i) choosing only algorithms that are variants of one original method; (ii) putting major attention to variants of one original method, but including into comparison also some much different approaches; (iii) choosing algorithms that have very different roots. The second and third approaches are followed here.

PSO and ABC are probably the most rapidly developed metaheuristics during recent decade. These algorithms are swarm based optimization algorithms. Chen et al. [47] very recently developed an efficient variant of PSO with an aging leader and challengers (ALC-PSO). By comparing ALC-PSO with eight popular PSO variants, it was found that ALC-PSO has a good balance between global and local exploration abilities which allows better optimized results to be found. ALC-PSO also enjoyed many attentions in the past three years that was developed. Therefore, ALC-PSO is taken as an effective comparison basis to evaluate the performance of proposed WEO. These algorithms are employed on a set of 17 benchmark functions as test bed. These problems are used frequently in the significant past studies in this area and considered herein in order to make a fair and unbiased comparison. There are a number of collections of benchmark problems based on artificially-created functions such as the CEC 2005 [48] for unified evaluation of optimization algorithms. Five of the functions of this set are from the CEC 2005. Apart from ALC-PSO, a modified ABC algorithm [49] (denoted as ABC/best) inspired by differential evolution (DE) is selected for evaluating the WEO. Also a particle swarm inspired multi-elitist ABC algorithm named PS-MEABC [50] which is the efficient hybridized version of the PSO and ABC algorithms is considered. These algorithms are also enjoyed more attention in the global optimization area because of their efficiency. Five and eight of our desired functions are used in the original work of these algorithms respectively, and we will use the available results for comparison.

The test functions can be divided to three groups [47]. The first five functions f_1–f_5 are unimodal functions with less complexity. These functions can evaluate the WEO for its well converging behavior. The next 7 functions f_6–f_{12} are multimodal functions with many local optima. These functions can test the global search ability of the algorithm in avoiding premature convergence. Finally the last five functions f_{13}–f_{17} are shifted and rotated functions introduced by Suganthan et al. [48]. These functions are more complex and can test the WEO more precisely. Table 1 summarizes the optimization problems reporting the number of design variables n (set equal to 30), the known global optima f_{opt}, the bounds of design variables, predefined value of error for the algorithm (Δf) and the denomination of each test problem in the optimization literature. Since the search process is governed by random rules, each test problem was solved by carrying out 30 independent optimization runs to obtain statistically significant results. During each run, the maximum number of 2×10^5 function evaluations is used. The maximum number of algorithm iterations (t_{max}) is equal to the result of dividing the maximum number of function evaluations to the number of water molecules (nWM). The performance assessment of the algorithm is carried out based on three metrics: Accuracy, reliability and speed of convergence. Accuracy will be presented by statistical results (i.e. best, worst, mean, and standard deviation). For assessing the reliability and speed of convergence of algorithm, a target objective function value (f_{target}) is considered for each problem using a predefined value (Δf) as error of the algorithm: $f_{opt} = f_{target} + \Delta f$. The reliability of algorithm can be reflected by the success rate (suc%) defined as the number of successful runs over the total number of 30 independent runs. A successful run refers to a run where the algorithm has reached the target objective function for the problem under study. The speed of the algorithm will be assessed by number of function evaluations that the algorithm needs to obtain an acceptable solution.

4.1.2. Search behavior of WEO on unconstrained functions

In this section, the search behavior of WEO is studied. The effects of each evaporation phase, the number of water molecules (nWM), minimum and maximum value of monolayer (MEP_{min} and MEP_{max}) and droplet (DEP_{min} and DEP_{max}) evaporation probabilities will be analyzed in detail. In particular the suitability of minimum and maximum value of monolayer and droplet evaporation probability based on the molecular dynamic simulation results ($MEP_{min} = 0.03$ and $MEP_{max} = 0.6$; $DEP_{min} = 0.6$ and $DEP_{max} = 1$) will be sought.

The search behavior of WEO is investigated using three benchmark functions: the unimodal Sphere function (f_1), the multimodal Schwefel's function (f_6), and the Shifted Rotated High Conditioned Elliptic function (f_{14}). The complexity increases as the index of function increases.

For a better study of the performance of the monolayer and droplet evaporation phases, Figs. 7–9 depict the convergence curves resulted by trial runs of WEO for these functions. Each function is solved two more times considering monolayer and droplet evaporation updating mechanisms alone for all t_{max} algorithm iterations which are reported as WEO-MEP and WEO–DEP, respectively. For a more precise monitoring of the results, the objective function axis of convergence curves is presented in the logarithmic scale. It should be noted that nWM is considered as 10, 20 and 10 for the above mentioned three functions, respectively. Based on these figures, it is clear that considering two phases in the content of WEO is inevitable to guarantee a good balance between global and local search ability and keep the dynamicity of the algorithm and preserves it from premature convergence. This effect becomes more apparent as the more problem gets harder.

Table 2 presents the statistical results of 10 independent runs for these three functions considering variable number of water molecules: 5, 10, 15, 20, 30, 50 and 100. For clarity the best observed results for each performance metric are marked in boldface. Considering nWM between 10 and 20, results in better

Table 1
Details of the mathematical optimization problems.

Test function	n	f_{opt}	Domain	Δf	Name
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	0	±100	0.01	Sphere
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	0	±10	0.01	Schwefel's P2.22
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} x_j \right)^2$	30	0	±100	100	Quadric
$f_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	0	±10	100	Rosenbrock
$f_5(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	30	0	±100	0	Step
$f_6(x) = 12569.5 + \sum_{i=1}^{n} - x_i \sin\left(\sqrt{x_i}\right)$	30	0	±500	2000	Schwefel's
$f_7(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	0	±5.12	10	Rastrigin
[a]$f_8(x) = \sum_{i=1}^{n} [y_i^2 - 10\cos(2\pi y_i) + 10]$	30	0	±5.12	10	Non-continuous Rastrigin
$f_9(x) = -20\exp\left(-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^2}\right) - \exp\left(1/n\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	30	0	±32	0.01	Ackley
$f_{10}(x) = 1/4000\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(x_i/\sqrt{i}\right) + 1$	30	0	±600	0.01	Griewank
$f_{11}(x) = \frac{\pi}{n}\left\{1 - \sin^2(\pi y_1) + \sum_{i=1}^{n-1}((y_i - 1)^2[1 + 10\sin^2 \pi y_{i+1}]) + (y_n - 1)^2\right\}$ $+\sum_{i=1}^{n} u(x_i, 10, 100, 4), y_i = 1 + \frac{1}{4}(x_i + 1)$	30	0	±50	0.01	Generalized penalized
$f_{12}(x) = \frac{1}{10}\left\{10\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})]\right\}$	30	0	±50	0.01	Generalized penalized
[b]$f_{13}(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} z_j \right)^2 - 450, z = x - o$	30	-450	±100	100	Shifted Schwefel's P1.2
$f_{14}(x) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} z_i^2 - 450, z = (x - o) \times M$	30	-450	±100	1e + 7	Shifted Rotated High Conditioned Elliptic
$f_{15}(x) = \sum_{i=1}^{n-1} [100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2] + 390, z = x - o + 1$	30	390	±100	100	Shifted Rosenbrock's
$f_{16}(x) = \sum_{i=1}^{n} [z_i^2 - 10\cos(2\pi z_i) + 10] - 330, z = (x - o) \times M$	30	-330	±5	200	Shifted Rotated Rastrigin's
[b]$f_{17}(x) = \sum_{i=1}^{n} \left(\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (z_i + 0.5))] \right) - n\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k \times 0.5)]$ $+90, a = 0.5, b = 3, k_{max} = 20, z = (x - o) \times M$	30	90	±0.5	30	Shifted Rotated Weierstrass

[a] In f_8, $y_i = \begin{cases} x_i & \lvert x_i \rvert < 0.5 \\ \frac{round(2x_i)}{2} & \lvert x_i \rvert \geqslant 0.5 \end{cases}$. In f_{11} and $f_{12} u(x_j, a, k, m) = \begin{cases} k(x_j - a)^m & \lvert x_i \rvert < a \\ 0 & -a \leqslant \lvert x_i \rvert \leqslant a \\ k(-x_j - a)^m & \lvert x_i \rvert < -a \end{cases}$

[b] In f_{13}–f_{17}, o is a shifted vector and M is a transformation matrix [48].

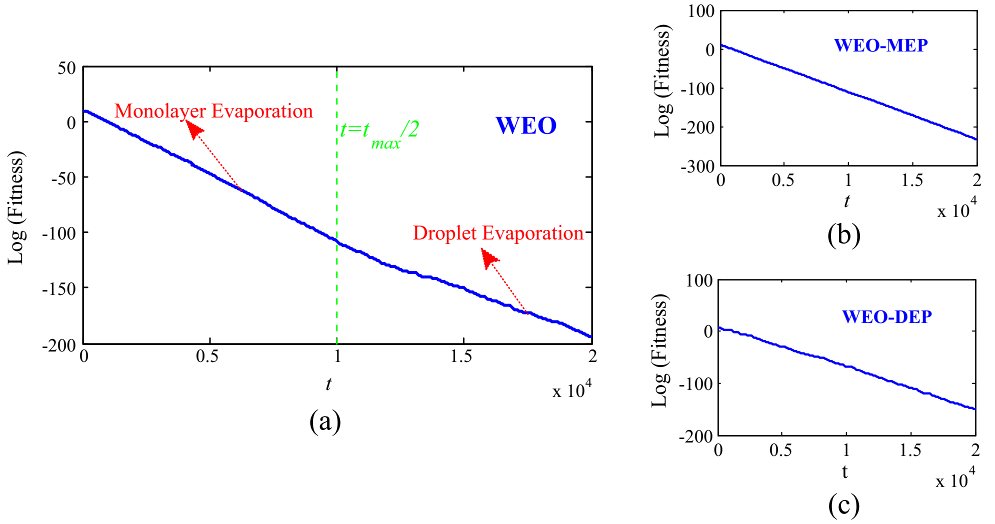

Fig. 7. Search behavior of the WEO on unimodal function f_1.

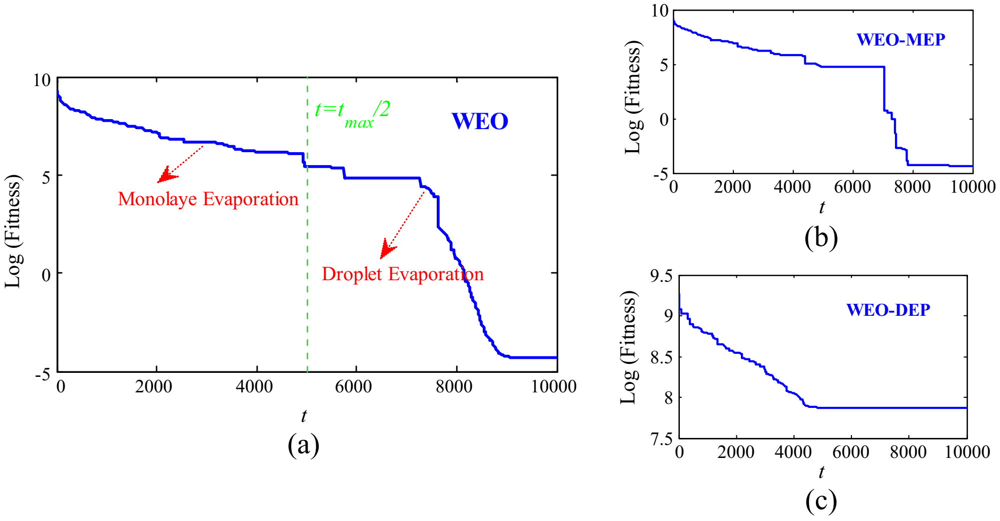

Fig. 8. Search behavior of the WEO on multimodal function f_6.

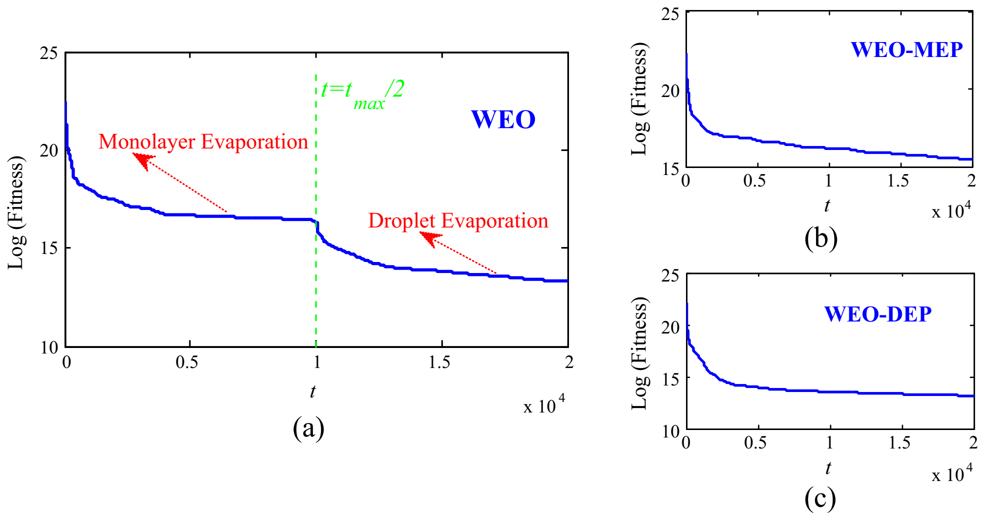

Fig. 9. Search behavior of the WEO on shifted rotated function f_{14}.

Table 2
Statistical and reliability results of the *nWM* parameter tuning analysis carried out for the problems f_1, f_6 and f_{14} (best observed results are marked in bold-face).

Function	*nWM*							
		5	10	15	20	30	50	100
f_1	Mean	989.86	**7.941e−99**	3.369e−61	9.158e−41	1.082e−22	6.386e−11	0.0038
	Best	94.27	**5.96e−101**	6.716e−62	1.052e−41	6.077e−23	2.359e−11	0.0021
	Worst	2404.18	**2.161e−98**	9.761e−61	2.717e−40	2.453e−22	9.162e−11	0.0046
	SD	796.29	**9.301e−99**	3.326e−61	1.049e−40	6.927e−23	2.647e−11	0.0009
	Suc%	0	100	100	100	100	100	100
	FEs	–	**12,270**	20,505	29,140	50,010	85,350	18,0400
f_6	Mean	1142.2	23.70	**1.338e−2**	0.40637	135.054	439.861	1081.773
	Best	473.76	1.338e−2	**1.338e−2**	1.338e−2	1.822	354.563	902.645
	Worst	1658.2	118.45	**1.338e−2**	1.9209	237.063	568.096	1188.139
	SD	383.29	47.38	**0**	0.7576	78.310	76.137	103.717
	Suc%	100	100	100	100	100	100	100
	FEs	6545	12,600	18,060	14,860	33,510	48,900	76,300
f_{14}	Mean	5.07e+6	**5.37e+5**	9.01e+5	1.22e+6	7.59e+5	7.29e+5	1.05e+7
	Best	2.92 e+6	**2.26 e+5**	4.04 e+5	8.09 e+5	4.21 e+5	2.50 e+5	8.11 e+6
	Worst	7.31e+6	**9.53e+5**	1.53e+6	2.21e+6	1.68e+6	1.06e+6	1.41e+7
	SD	1.39e+5	**2.49e+5**	4.31e+5	5.12e+5	4.70e+5	3.00e+5	2.08e+6
	Suc%	100	100	100	100	100	100	60
	FEs	89,610	**9330**	10,635	11,520	30,420	51,350	17,7100

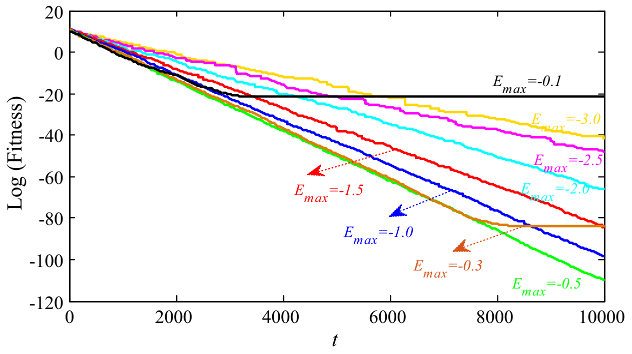

Fig. 10. Convergence of the WEO for $MEP_{min} = 0.03$ and different values of MEP_{max} on unimodal function f_1.

performance of the algorithm. *nWM* will be considered 10 for all functions except the function f_6 in which it will be taken as 15. Increasing the number of water molecules, results in poor performance of the algorithm in aspects of accuracy and speed.

To study the suitability of monolayer and droplet evaporation probability values ($MEP_{min} = 0.03$ and $MEP_{max} = 0.6$; $DEP_{min} = 0.6$ and $DEP_{max} = 1$) legislated based on the MD simulations results, the algorithm employed for these three functions several times and it is found that these values lead to efficient performance of WEO. Let us consider [−3.5, −3, −2.5, −2, −1.5, −1] as minimum values of E_{sub} which results in MEP_{min} between [0.03, 0.6]. For each MEP_{min} different values are considered for MEP_{max} and trial run of algorithm (only monolayer evaporation phase) is conducted. For example considering E_{sub} equal to −3.5 ($MEP_{min} = 0.03$) different values of MEP_{max} will be obtained considering [−3, −2.5, −2, −1.5, −1, −0.5, −0.3, −0.1] as maximum values for E_{sub}. Fig. 10 depicts the algorithm convergence curves monitored for f_1 function with $MEP_{min} = -0.03$ and different values of MEP_{max}. Obviously considering maximum value of substrate energy equal to −0.5 which is equivalent to $MEP_{max} = 0.6$ provides better performance of the algorithm. This behavior was seen for E_{max} equal to −0.5 or near values regardless of the problem and considered value of MEP_{min}. Simulation results also show that considering MEP_{min} as 0.03, leads the efficiency of the algorithm. Such a behavioral study of the algorithm is also performed for different values of DEP_{min} and DEP_{max}, and it is found that $DEP_{min} = 0.6$ and $DEP_{max} = 1$ are suitable values.

4.1.3. Comparison results with ALC-PSO

Tables 3–5 compare the results obtained by WEO with those of ALC-PSO in relation with accuracy, reliability and speed of convergence for unimodal, multimodal and shifted rotated functions, respectively. From accuracy point of view, WEO is better than the ALC-PSO in 5 functions ($f_2, f_6, f_{10}, f_{11}, f_{14}$). It should be noted that WEO is able to reach the near results to the ALC-PSO in almost all functions. From reliability and speed of convergence point of view, WEO is comparable with the ALC-PSO.

Fig. 11 firstly presents the convergence history of WEO for the best observed results and the average of the 30 independent runs.

Table 3
Comparing results of the WEO with ALC-PSO [47] on unimodal functions.

Function		Mean	Best	Worst	SD	SUC%	FEs
f_1	ALC-PSO	1.677×10^{-161}	1.135×10^{-172}	–	8.206×10^{-161}	100	7491
	WEO	6.243×10^{-89}	1.407×10^{-98}	1.872×10^{-87}	3.361×10^{-88}	100	9880
f_2	ALC-PSO	1.161×10^{-90}	1.121×10^{-98}	–	4.146×10^{-90}	100	8180
	WEO	0	0	0	0	100	670
f_3	ALC-PSO	1.792×10^{-11}	8.267×10^{-14}	–	3.538×10^{-11}	100	23,839
	WEO	1.449×10^{-05}	6.291×10^{-07}	1.177×10^{-04}	2.435×10^{-05}	100	36,370
f_4	ALC-PSO	7.613	3.729×10^{-7}	–	6.658	100	6406
	WEO	22.042	1.347×10^{-4}	75.256	16.765	100	6530
f_5	ALC-PSO	0	0	–	0	100	13,176
	WEO	0.400	0	3	0.757	73	7930

Table 4
Comparing results of the WEO with ALC-PSO [47] on multimodal functions.

Function		Mean	Best	Worst	SD	SUC%	FEs
f_6	ALC-PSO	21.028	1.338×10^{-2}	–	54.103	100	46,698
	WEO	1.338×10^{-2}	1.338×10^{-2}	1.338×10^{-2}	0	100	14,505
f_7	ALC-PSO	2.528×10^{-14}	7.105×10^{-15}	–	1.376×10^{-14}	100	74,206
	WEO	0.265	0	3.979	0.768	100	20,120
f_8	ALC-PSO	1.251×10^{-11}	0	–	6.752×10^{-11}	100	58,900
	WEO	0.233	0	2	0.496	100	18,580
f_9	ALC-PSO	1.148×10^{-14}	7.694×10^{-15}	–	2.941×10^{-15}	100	58,900
	WEO	0.081	1.510×10^{-14}	1.155	0.266	90	15,260
f_{10}	ALC-PSO	1.221×10^{-2}	0	–	1.577×10^{-2}	100	10,161
	WEO	1.110×10^{-17}	0	2.220×10^{-16}	4.394×10^{-17}	100	12,890
f_{11}	ALC-PSO	4.393×10^{-32}	1.570×10^{-32}	–	7.602×10^{-32}	100	28,869
	WEO	2.190×10^{-32}	1.571×10^{-32}	2.016×10^{-31}	3.336×10^{-32}	100	11,140
f_{12}	ALC-PSO	3.167×10^{-31}	1.350×10^{-31}	–	6.918×10^{-31}	100	42,466
	WEO	0.00366	1.350×10^{-31}	0.10987	0.01972	97	10,050

Table 5
Comparing results of the WEO with ALC-PSO [47] on shifted rotated functions.

Function		Mean	Best	Worst	SD	SUC%	FEs
f_{13}	ALC-PSO	7.342×10^{-10}	2.728×10^{-12}	–	2.694×10^{-9}	100	30,736
	WEO	0.00097	5.559×10^{-05}	0.00808	0.00155	100	70,900
f_{14}	ALC-PSO	888,006	394,882	–	414,729	100	9277
	WEO	786468.4	417143.5	1452454.1	255628.6	100	29,820
f_{15}	ALC-PSO	21.839	4.916×10^{-4}	–	40.414	90	34,007
	WEO	36.52	0.03597	149.71	43.43	93	18,470
f_{16}	ALC-PSO	111.633	62.682	–	28.375	100	17,121
	WEO	156.27	95.52	226.85	32.42	87	9500
f_{17}	ALC-PSO	23.444	16.114	–	3.618	50	36,270
	WEO	26.97	21.91	31.74	2.99	77	7120

Plots are relative to test problems f_1, f_6, f_{14}. The presence of continuous steplike movements indicates how WEO can escape from local optima and find better designs thus avoiding premature convergence. In order to further analyze convergence behavior, optimization histories of the best and worst water molecules and the average optimization history of all water molecules corresponding to a single independent run of the WEO are also plotted in this figure. It appears that WEO reaches to an effective balance between global search and local search. It can be always seen that the difference between the average of all particles and the worst particle is smaller than the difference between average and the best particle. Another important aspect to be underlined is that in the early optimization cycles where randomness plays the main rule, the distance between the best particle diagram and both average and worst particle diagrams is small. WEO tries to increase this distance in the next iterations: this demonstrates the diversification capability of the present optimization technique. The distance then decreases again as the search process progresses and finally becomes negligible upon reaching the optimum design: this demonstrates the intensification ability of WEO. Such a behavior was observed in all test problems. The previous discussion demonstrates that WEO achieves an effective balance between diversification and intensification.

4.1.4. Comparison results with ABC/best and PS-MEABC

The performance on the solution accuracy of WEO is compared with those of ABC/best and PS-MEABC in Tables 6 and 7, respectively. For clarity the best observed results are marked in boldface. Obviously WEO shows comparable performance with the two efficient modified variants of PSO and ABC algorithms.

4.2. Experimental results and discussion on constrained problems

4.2.1. Benchmark and experiments settings

To evaluate the performance of the proposed WEO on continuous constrained optimization problems (COPs), 13 well-known benchmark problems are used. These problems have been first proposed by Hock and Schittkowski [51] and later extended by Michalewicz and Schoenauer [52] and Runarsson and Yao [53]. These test suit also recently used by Gandomi et al. [54] for evaluating the Bat Algorithm (BA). In order to further investigate its performance for practical problems, WEO was used to deal with engineering design problems in Kaveh and Talatahari [55]. Since the search process is governed by random rules, each test problem is solved by carrying out 50 independent optimization runs to obtain statistically significant results. During each run, the maximum number of objective function evaluations of 2×10^4 is used. The maximum number of algorithm iterations (t_{max}) is equal to the result of dividing the maximum number of structural analysis to the number of water molecules (nWM).

From the implementation point of view, a major barrier in solving COPs is proper handling of nonlinear constraints [56]. There are various methods for constraint handling. The penalty method is used in this study [54]. The idea is to define a penalty function so that the constrained problem is transformed into an unconstrained problem. Considering nonlinear equality constraints φ_i and inequality constraints ψ_i, the following equation can be defined:

$$\Pi(x, \mu_i, \nu_i) = f(x) + \sum_{i=1}^{n} \mu_i \phi_i^2(x) + \sum_{j=1}^{m} \nu_j \psi_j^2(x) \qquad (12)$$

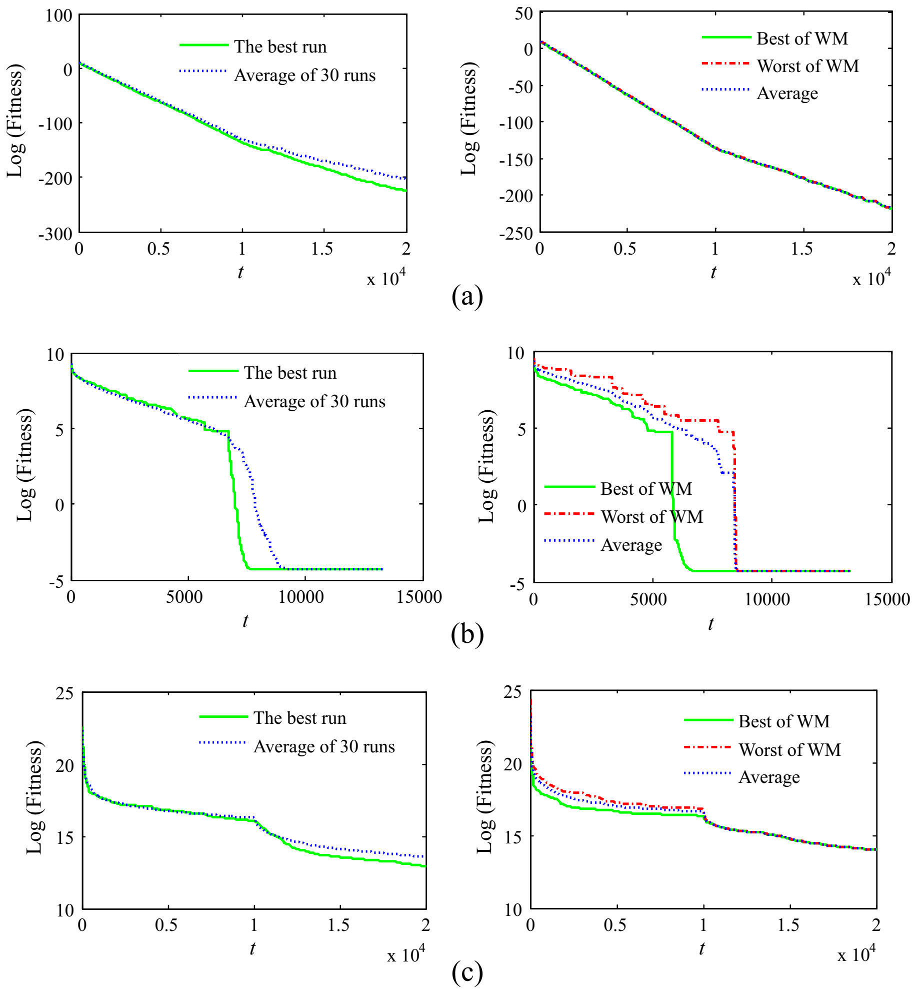

Fig. 11. Comparison of convergence curves relative to the best design and average optimization run and Global search speed and local search speed of the WEO: (a) f_1; (b) f_6; and (c) f_{14}.

Table 6
Comparison of the results of the WEO with those of ABC/best (best observed results are marked in bold-face).

Function	ABC/best [49]		WEO (present study)	
	Mean	SD	Mean	SD
f_1	1.57×10^{-27}	1.14×10^{-27}	$\mathbf{6.243 \times 10^{-89}}$	$\mathbf{3.361 \times 10^{-88}}$
f_2	3.45×10^{-15}	8.79×10^{-16}	**0**	**0**
f_5	0	**0**	0	0.757
f_7	0	**0**	0	0.768
f_8	0	**0**	0	0.496
f_9	$\mathbf{1.26 \times 10^{-13}}$	$\mathbf{3.48 \times 10^{-14}}$	0.081	0.266
f_{10}	4.23×10^{-11}	2.16×10^{-11}	$\mathbf{1.110 \times 10^{-17}}$	$\mathbf{4.394 \times 10^{-17}}$
f_{11}	2.85×10^{-30}	2.19×10^{-30}	$\mathbf{2.190 \times 10^{-32}}$	$\mathbf{3.336 \times 10^{-32}}$

where $1 \leqslant \mu_i$ and $0 \leqslant \upsilon_i$ which should be large enough, depending on the solution quality needed. As can be seen, when an equality constraint is met, its contribution to Π is zero. However, when it is violated, it is heavily penalized as it significantly increases Π. Similarly, it is true when inequality constraints become tight or exact.

In the following, first the search behavior of WEO is studied in detail for COPs. In the next two subsections WEO is applied to 13 benchmark constrained functions and three well-known engineering design problems and evaluated by comparing with other efficient metaheuristics available in the optimization literature.

4.2.2. Search behavior of WEO on COPs

In this subsection, the search behavior of WEO is studied on COPs. The effects of each evaporation phase, the number of water molecules (nWM), minimum and maximum value of monolayer (MEP_{min} and MEP_{max}) and droplet (DEP_{min} and DEP_{max}) evaporation probabilities are analyzed in detail. In particular the suitability of

Table 7
Comparison of the results of the WEO with those of PS-MEABC (best observed results are marked in bold-face).

Function	PS-MEABC [50]			WEO (present study		
	Mean	SD	Best	Mean	SD	Best
f_{13}	427.6	267.2	84.11	$\mathbf{9.7 \times 10^{-4}}$	$\mathbf{2.694 \times 10^{-9}}$	$\mathbf{5.559 \times 10^{-05}}$
f_{14}	3.550×10^6	1.756×10^6	1.458×10^6	$\mathbf{0.786 \times 10^6}$	$\mathbf{0.256 \times 10^6}$	$\mathbf{0.417 \times 10^6}$
f_{15}	**11.96**	**24.49**	$\mathbf{3.161 \times 10^{-05}}$	36.52	43.43	3.597×10^{-02}
f_{16}	**56.46**	**10.23**	**39.18**	156.27	32.42	95.52
f_{17}	27.02	**1.61**	23.77	**26.97**	2.99	**21.91**

minimum and maximum value of monolayer and droplet evaporation probability based on the molecular dynamic simulation results ($MEP_{min} = 0.03$ and $MEP_{max} = 0.6$; $DEP_{min} = 0.6$ and $DEP_{max} = 1$) are investigated.

The search behavior of WEO is studied using first two constrained functions (G_{01} and G_{02}) which contain 13 and 20 number of design variables and 9 and 2 number of constraints, respectively. This behavior is also investigated using one of the engineering design problems (a welded beam design). For a better study of the performance of the monolayer and droplet evaporation phases, Fig. 12 depicts the average convergence curves resulted by 10 independent runs of WEO for these problems. Each test case is solved two more series of 10 independent runs considering monolayer and droplet evaporation updating mechanisms alone for all t_{max} algorithm iterations which are reported as WEO-MEP and WEO–DEP, respectively. It should be noted that *nWM* is considered as 10. For a more precise monitoring of the results, the not penalized objective function convergence histories are divided into two parts by taking the 400th iteration as the separating point. Based on these figures, it is clear that considering two phases in the content of WEO is inevitable to guarantee a good balance between global and local search ability and keep the dynamicity of the algorithm and preserves it from premature convergence. The most interesting observed result is the way the WEO math WEO-MEP in the first half of the iterations and then tries to coincide with the WEO-DEP, especially in the engineering design problem. It should be noted that in the welded beam design problem WEO-DEP results in almost infeasible solutions. In this regards WEO-DEP was conducted many times to reach the feasible optimum designs for 10 times.

Table 8 presents the statistical results of 10 independent runs for these three test problems considering variable number of water molecules: 5, 10, 15, 20, 30, 50 and 100, in which Mean, Best, Worst and SD stands for average, best observed, worst observed, and standard deviation of optimum designs resulted from 10 independent runs, respectively. For clarity, the best observed results for each performance metric are marked in boldface. Considering *nWM* between 10 and 20, results in better performance of the algorithm. Although 15 number of water molecules results in robust performance of the algorithm for G_{02} and welded beam design problem, *nWM* will be considered equal to 10 for all problems by which the best result is achievable. Increasing the number of water molecules, results in poor performance of the algorithm in the aspect of accuracy and robustness.

To study the suitability of monolayer and droplet evaporation probability values ($MEP_{min} = 0.03$ and $MEP_{max} = 0.6$; $DEP_{min} = 0.6$ and $DEP_{max} = 1$) legislated based on the MD simulation results, the algorithm is employed for these three problems several times and it is found that these values lead to efficient performance of WEO. Let us consider [−3.5, −3, −2.5, −2, −1.5, −1] as minimum values of E_{sub} which results in MEP_{min} between [0.03, 0.6]. For each MEP_{min} different values are considered for MEP_{max} and 10 independent runs of algorithm (only monolayer evaporation phase) is conducted. For example considering E_{sub} equal to −3.5 ($MEP_{min} = 0.03$) different values of MEP_{max} will be obtained considering [−3, −2.5, −2, −1.5, −1, −0.5, −0.3, −0.1] as maximum values for E_{sub}. The obtained mean weight of 10 independent runs for different sets of monolayer evaporation probabilities for the G_{01} function is depicted in the Fig. 13. Algorithm performance is almost unchanged for different values of the MEP_{min} and considering

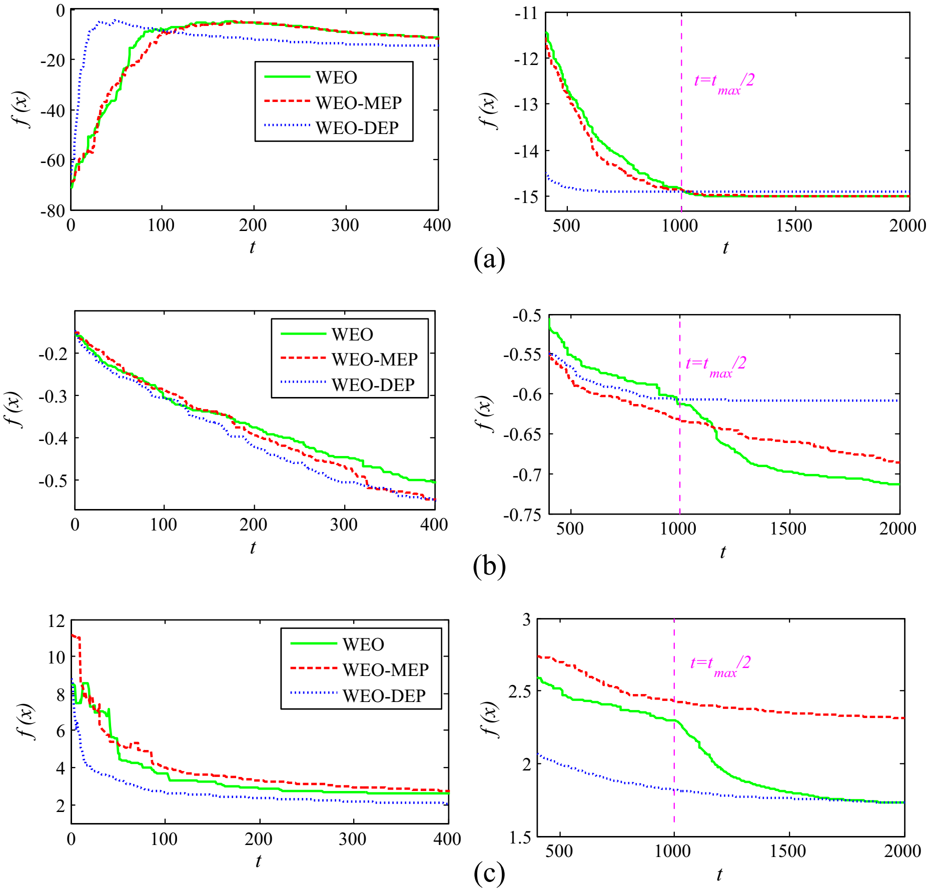

Fig. 12. Search behavior of the WEO on: (a) G_{01}, (b) G_{02}, (c) welded beam design problem.

Table 8
Statistical and reliability results of the *nWM* parameter tuning analysis carried out for the G_{01}, G_{02}, and Welded beam design problems (best observed results are marked in bold-face).

Function		*nWM*						
		5	10	15	20	30	50	100
G01	Mean	−11.742	**−14.9999999735053**	−14.999879	−14.99642	−14.91	−14.19	−12.41
	Best	−14.9999998	**−14.9999999981958**	−14.999992	−14.99959	−14.96	−14.74	−13.58
	Worst	−9.00	**−14.9999998691365**	−14.999363	−14.9891	−14.86	−13.40	−11.33
	SD	2.327	**3.6802e−08**	0.0001703	0.0033	0.035	0.381	0.713
G02	Mean	−0.5269404	−0.7039884	**−0.7259382**	−0.6843342	−0.6428178	−0.5602645	−0.4545068
	Best	−0.6360198	**−0.7466322**	−0.7363992	−0.7419955	−0.6779439	−0.6137082	−0.5206279
	Worst	−0.4094870	−0.6413440	**−0.7111326**	−0.6332578	−0.6002330	−0.4952313	−0.4068388
	SD	0.0762186	0.0386825	**0.0088275**	0.0302885	0.0275776	0.0319804	0.0299364
Welded beam	Mean	2.6812157	1.7254234	**1.7251959**	1.7273127	1.7486177	1.8352480	1.8990273
	Best	1.9822340	**1.7248523**	1.7248779	1.7252087	1.7337169	1.7793478	1.8085300
	Worst	3.3015363	1.7295733	**1.7263152**	1.7329101	1.7679601	1.9200787	2.1067792
	SD	0.3784889	0.0014106	**0.0005041**	0.0024157	0.0101151	0.0386435	0.0752280

maximum value of substrate energy equal to −0.5 which is equivalent to $MEP_{max} = 0.6$. Fig. 14 depicts the penalized objective function convergence curves of a single trial run monitored for G_{01} function with $MEP_{min} = 0.03$ and different values of MEP_{max}. For a more precise monitoring of the results, the penalized objective function convergence histories in the logarithmic scale are divided into two parts by taking $\Pi(x) = 5$ as the separating point. Considering $MEP_{max} = 0.6$ is efficient for the dynamicity and well converging behavior of the algorithm. Such a behavioral study of the algorithm is also performed for different values of DEP_{min} and DEP_{max}, and it is found that $DEP_{min} = 0.6$ and $DEP_{max} = 1$ are suitable values.

To conclude, the results in this subsection reveal that WEO can be found effective and stable in solving COPs, considering *nWM* equal to 10 and considering monolayer and droplet evaporation probability values as: $MEP_{min} = 0.03$, $MEP_{max} = 0.6$, $DEP_{min} = 0.6$, and $DEP_{max} = 1$ which are legislated based on the MD simulation results.

4.2.3. Implementation of WEO for 13 benchmark constrained functions

WEO was benchmarked using 13 well-known constrained problems (G_{01}–G_{13}). The main characteristics of the employed problems are presented in Table 9 [54]. Of the 13 benchmark problems G_{02}, G_{03}, and G_{08} are maximization problems, which are transformed into minimization problems using $-f(x)$, and the others are minimization problems. Problems G_{03}, G_{05}, G_{11} and G_{13} contain equality constraints. The details of these problems can be found in Gandomi et al. [54].

Gandomi et al. [54] have used Bat Algorithm (BA) for this set of functions and found that BA is able to find global optimum in all cases with a very good performance in comparison to many other metaheuristic algorithms and numerical methods. Elsayed et al. [57] recently have proposed a mix of particle swarm variants in one algorithmic structure with a self-adaptive mechanism (SAM-PSO) and used for this set of test problems and found that SAM-PSO shows more consistent performances than each of PSO individual variants, and is always either better than, or the same as, the state-of-the-art algorithms. BA and SAM-PSO was selected for comparing with the proposed WEO for achieving an efficient evaluating. Table 10 presents the statistical results obtained for 50 independent runs carried out from different initial populations randomly generated for WEO besides BA and SAM-PSO algorithms. As it is clear WEO is able to find the global optimum in all test problems and has competitive performance to the BA algorithm. The best results are obtained by SAM-PSO.

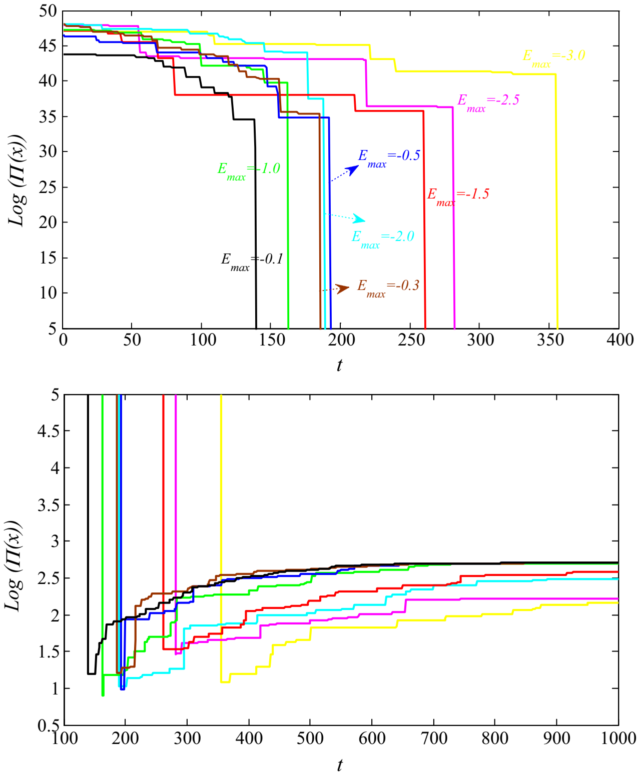

Fig. 14. Convergence of the WEO for $MEP_{min} = 0.03$ and different values of MEP_{max} on G_{01} problem.

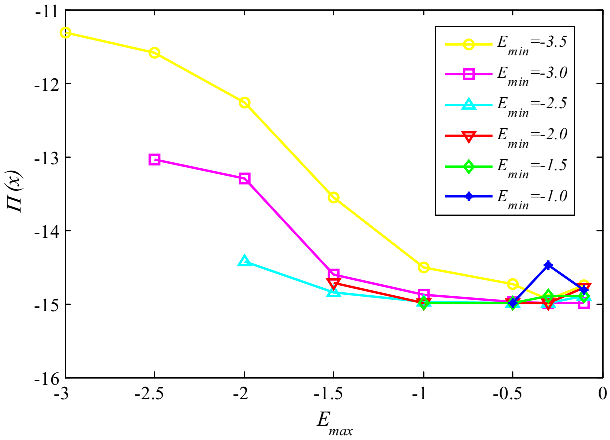

Fig. 13. Mean penalized objective function of 10 independent runs for different values of MEP_{min} and MEP_{max} on G_{01} function.

Table 9
Characteristics of the benchmark problems.

ID	Function type	Problem type	No. of variables	No. of constraints	Global optimum
G_{01}	Quadratic	Min	13	9 [I] + 0 [E]	−15
G_{02}	Nonlinear	Max	20	2 [I] + 0 [E]	−0.803619
G_{03}	Polynomial	Max	10	0 [I] + 1 [E]	1
G_{04}	Quadratic	Min	5	6 [I] + 0 [E]	−30665.539
G_{05}	Cubic	Min	4	2 [I] + 3 [E]	5126.4981
G_{06}	Cubic	Min	2	2 [I] + 0 [E]	−6961.81388
G_{07}	Quadratic	Min	10	8 [I] + 0 [E]	24.30621
G_{08}	Nonlinear	Max	2	2 [I] + 0 [E]	0.095825
G_{09}	Polynomial	Min	7	4 [I] + 0 [E]	680.6300573
G_{10}	Linear	Min	8	6 [I] + 0 [E]	7049.3307
G_{11}	Quadratic	Min	2	0 [I] + 1[E]	0.75
G_{12}	Quadratic	Min	3	9^3 [I] + 0[E]	1
G_{13}	Exponential	Min	5	0 [I] + 3 [E]	0.0539498

4.2.4. Implementation of WEO for engineering design problems

4.2.4.1. A tension/compression spring design problem. One of the well-known engineering optimization problems is the design of a tension–compression spring for a minimum weight subject to constraints on shear stress, surge frequency, and minimum deflection as shown in Fig. 15. The design variables are the mean coil diameter D; the wire diameter d, and the number of active coils N which are denoted by x_1, x_2 and x_3, respectively. The problem can be stated with the cost function:

$$f_{cost}(x) = (x_3 + 2)x_2x_1^2 \tag{13}$$

to be minimized and constrained with:

$$\begin{aligned} g_1(x) &= 1 - \frac{x_2^3x_3}{71785x_1^4} \leqslant 0 \\ g_2(x) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leqslant 0 \\ g_3(x) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leqslant 0 \\ g_4(x) &= \frac{x_1 + x_2}{1.5} - 1 \leqslant 0 \end{aligned} \tag{14}$$

The design space is bounded by $0.05 \leqslant x_1 \leqslant 2$, $0.25 \leqslant x_2 \leqslant 1.3$, $2 \leqslant x_3 \leqslant 15$.

This problem is optimized by many researchers using various mathematical, numerical and metaheuristic optimization methods. The best results for this problem, based on the author's knowledge obtained almost by metaheuristic algorithms. From them GA-based [58], co-evolutionary particle swarm optimization (CPSO)

Table 10
Comparison of accuracy, robustness and reliability of WEO with BA and SAM-PSO on 13 benchmark functions.

Prob	Algorithm	Mean	Best	Worst	SD
G_{01}	BA	−14.65818	−15.00000	−12.45309	0.70457
	SAM-PSO	−15.0000	−15.0000	NA	0.00E+00
	WEO	−14.97702	−15.00000	−13.8281	0.16248
G_{02}	BA	0.7534700	0.8036191	0.5625529	0.0540384
	SAM-PSO	−0.79606	−0.803619	NA	5.3420E−03
	WEO	−0.7195518	−0.803619	−0.6060756	0.04312859
G_{03}	BA	0.98955	0.99998	0.93641	0.01706
	SAM-PSO	−1.0005	−1.0005	NA	0.00E+00
	WEO	−0.98976	−0.99998	−0.93641	0.01876
G_{04}	BA	−30665.539	−30665.539	−30665.539	0.000000000
	SAM-PSO	−30665.539	−30665.539	NA	0.00E+00
	WEO	−30665.4134	−30665.539	−30659.5104	0.84389
G_{05}	BA	5129.4248	5126.4987	5181.4744	10.89206
	SAM-PSO	5126.4967	5126.4967	NA	1.3169E−10
	WEO	5126.5991	5126.4985	5130.6872	0.585324
G_{06}	BA	−6961.81387	−6961.81388	−6961.81388	0.000000000
	SAM-PSO	−6961.8139	−6961.8139	NA	0.00E+00
	WEO	−6961.79559	−6961.81388	−6961.265062	0.0785323
G_{07}	BA	24.30650	24.30622	24.30799	0.00038
	SAM-PSO	24.306209	24.306209	NA	1.9289E−08
	WEO	24.306647	24.306236	24.30969	0.00039
G_{08}	BA	0.0958250	0.0958250	0.0958250	0.000000000
	SAM-PSO	−0.09582504	−0.09582504	NA	0.00E+00
	WEO	−0.0811551	−0.095825041	−0.029143	0.02762
G_{09}	BA	680.6300574	680.6300574	680.6300574	0.000000000
	SAM-PSO	680.630057	680.630057	NA	0.00E+00
	WEO	680.6657831	680.631073	680.8036	0.041
G_{10}	BA	7049.4713	7049.2606	7051.7822	0.4913431
	SAM-PSO	7049.248	7049.248	NA	1.5064E−05
	WEO	7105.68761	7049.32234	7945.85180	141.13235
G_{11}	BA	0.75000001	0.750000	0.750000012	0.00000001
	SAM-PSO	0.7499	0.7499	NA	0.00E+00
	WEO	0.7500055	0.7500000	0.750077	0.00000169
G_{12}	BA	1.0000000	1.0000000	1.0000000	0.000000000
	SAM-PSO	−1.0000	−1.0000	NA	0.00E+00
	WEO	−1.0000000	−1.0000000	−1.0000000	0.000000000
G_{13}	BA	0.0539520	0.0539499	0.0539723	0.000005646
	SAM-PSO	0.053942	0.053942	NA	0.00E+00
	WEO	0.0539548	0.0539499	0.0539873	0.000008693

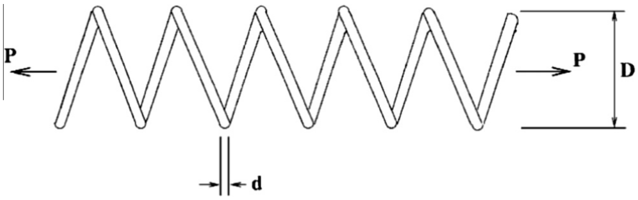

Fig. 15. Schematic of the tension/compression spring.

Table 11
Comparison of accuracy, robustness and reliability of WEO with other metaheuristics on the spring design problem.

Algorithm	Best	Mean	Worst	SD
GA	0.012681	0.012742	0.012973	5.90E−05
CPSO	0.012675	0.01273	0.012924	5.20E−05
ESs	0.012698	0.013461	0.16485	9.66E−04
IACO	0.012643	0.01272	0.012884	3.49E−05
CSS	0.012638	0.012852	0.013626	8.36E−05
BA	0.012665	0.013501	0.016895	0.00142
WEO	0.012665	0.012669	0.012697	0.000006

Table 12
Optimum designs obtained by WEO and various metaheuristic algorithms for the spring design problem.

Methods	Optimal design variables			$f_{cost}(x)$
	$x_1(d)$	$x_2(D)$	$x_3(N)$	
GA	0.051989	0.363965	10.89052	0.012681
CPSO	0.051728	0.357644	11.24454	0.012675
ESs	0.051643	0.35536	11.39793	0.012698
IACO	0.051865	0.3615	11	0.012643
CSS	0.051744	0.358532	11.1657	0.012638
BA	0.05169	0.35673	11.2885	0.01266522
WEO	0.051685	0.356630	11.294103	0.012665

[59], evolution strategies (ESs) [60], an Improved Ant Colony Optimization (IACO) [61], Charged System Search (CSS) algorithm [55], and Bat Algorithm (BA) [54] are used successfully for this problem. Table 11 tabulates the statistical results yielded by WEO for this problem besides the ones obtained by other metaheuristics. The best result is obtained by the CSS algorithm. As it is clear WEO is competitive with others in the aspect of accuracy and but shows the most robust results. Table 12 presents the optimum design obtained by WEO and other metaheuristic algorithms.

4.2.4.2. A welded beam design. WEO is also applied to the optimum design of a well-known welded beam problem as shown in the Fig. 16. The aim is to find the minimum fabricating cost of the welded beam subjected to constraints on shear stress, bending stress, buckling load, end deflection, and side constraint. There are four design variables, namely h (=x_1), l (=x_2), t (=x_3), and b (=x_4). The objective function of the problem is expressed as follows:

$$f_{cost}(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \qquad (15)$$

which is subjected to 7 inequalities constraints. Additional information about the constraints and variable bounds can be found in [55].

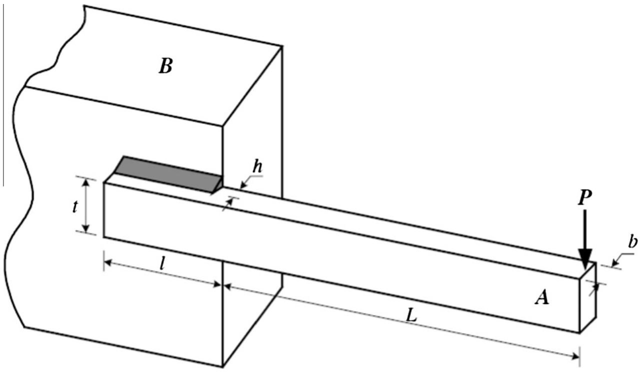

Fig. 16. A welded beam system.

Table 13
Comparison of accuracy, robustness and reliability of WEO with other metaheuristics on the welded beam design problem.

Algorithm	Best	Mean	Worst	SD
GA	1.728226	1.792654	1.993408	0.074713
CPSO	1.728024	1.748831	1.782143	0.012926
ESs	1.7373	1.81329	1.994651	0.0705
IACO	1.724918	1.729752	1.775961	0.0092
CSS	1.724866	1.739654	1.759479	0.008064
WEO	1.724852	1.739437	1.818454	0.02314

Table 14
Optimum designs obtained by WEO and various metaheuristic algorithms for the welded beam design problem.

Methods	Optimal design variables				$f_{cost}(x)$
	$x_1(h)$	$x_2(l)$	$x_3(t)$	$x_4(b)$	
GA	0.205986	3.471328	9.020224	0.20648	1.728226
CPSO	0.202369	3.544214	9.04821	0.205723	1.728024
ESs	0.199742	3.61206	9.0375	0.206082	1.7373
IACO	0.2057	3.471131	9.036683	0.205731	1.724918
CSS	0.20582	3.468109	9.038024	0.205723	1.724866
WEO	0.205730	3.470489	9.036624	0.205730	1.724852

Table 13 presents the statistical results yielded by WEO for this problem besides the ones obtained by a GA-based algorithm [58], co-evolutionary particle swarm optimization (CPSO) [59], evolution strategies (ESs) [60], an Improved Ant Colony Optimization (IACO) [61], and Charged System Search (CSS) algorithm [55]. The best result is obtained by the CSS algorithm. As it is clear WEO results the best design and is competitive with others in the aspect of robustness. Table 14 tabulates the optimum design obtained by WEO and other metaheuristic algorithms.

4.2.4.3. A pressure vessel design problem. Another well-known engineering optimization problem is the design of a pressure vessel. A cylindrical pressure vessel capped at both ends by hemispherical heads is presented in Fig. 17. The objective is to minimize the total cost, including the cost of material, forming and welding [55]:

$$f_{cost}(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \qquad (16)$$

where x_1 is the thickness of the shell (T_s), x_2 is the thickness of the head (T_h), x_3 is the inner radius (R), and x_4 is the length of cylindrical section of the vessel (L), not including the head. This problem is a mixed type of optimization. T_s and T_h are integer multiples of 0.0625 inch, the available thickness of the rolled steel plates, R and L are continuous. The constraints are as follows:

$$\begin{aligned}
&g_1(x) = -x_1 + 0.0193x_3 \leqslant 0\\
&g_2(x) = -x_2 + 0.00954x_3 \leqslant 0\\
&g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1{,}296{,}000 \leqslant 0\\
&g_4(x) = x_4 - 240 \leqslant 0
\end{aligned} \qquad (17)$$

The design space is bounded by $0 \leqslant x_1, x_2 \leqslant 99$, $10 \leqslant x_3, x_4 \leqslant 200$.

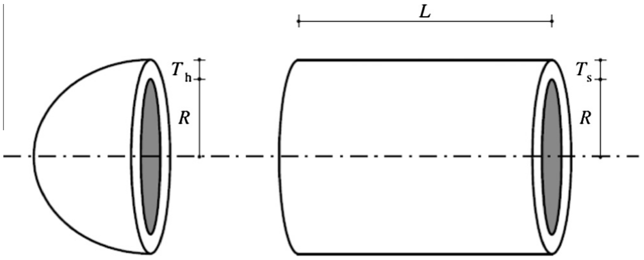

Fig. 17. Schematic of a pressure vessel.

Table 15
Comparison of accuracy, robustness and reliability of WEO with other metaheuristics on the pressure vessel design problem.

Algorithm	Best	Mean	Worst	SD
GA	6059.95	6177.25	6469.32	130.9297
CPSO	6061.08	6147.13	6363.80	86.4545
ESs	6059.75	6850.00	7332.88	426
IACO	6059.73	6081.78	6150.13	67.2418
CSS	6059.09	6067.91	6085.48	10.2564
BA	6059.71	6179.13	6318.95	137.223
WEO	6059.71	6138.61	6410.19	129.90332

Table 16
Optimum designs obtained by WEO and various metaheuristic algorithms for the pressure vessel design problem.

Methods	Optimal design variables				$f_{cost}(x)$
	$x_1(T_s)$	$x_2(T_h)$	$x_3(R)$	$x_4(L)$	
GA	0.812500	0.437500	42.097398	176.654050	6059.9463
CPSO	0.812500	0.437500	42.091266	176.746500	6061.0777
ESs	0.812500	0.437500	42.098087	176.640518	6059.7456
IACO	0.812500	0.437500	42.098353	176.637751	6059.7258
CSS	0.812500	0.437500	42.103624	176.572656	6059.0888
BA	0.812500	0.437500	42.098445	176.636595	6059.7143
WEO	0.812500	0.437500	42.098444	176.636622	6059.71

This problem is optimized by many researchers using various mathematical, numerical and metaheuristic optimization methods. The best results for this problem, based on the author's knowledge obtained almost by metaheuristic algorithms. From them GA-based [58], co-evolutionary particle swarm optimization (CPSO) [59], evolution strategies (ESs) [60], an Improved Ant Colony Optimization (IACO) [61], Charged System Search (CSS) algorithm [55], and Bat Algorithm (BA) [54] are used successfully for this problem. Table 15 tabulates the statistical results yielded by WEO for this problem besides the ones obtained by other metaheuristics. The best results are obtained firstly by the CSS and then by BA algorithm. WEO is able to find the same best result obtained by the BA and the difference with the one obtained by CSS is negligible. Robustness of the WEO is competitive with other algorithms except the CSS. Table 16 presents the optimum designs obtained by WEO and other metaheuristic algorithms.

5. Conclusions

In this paper a new physically inspired population based metaheuristic is presented to solve global optimization problems. This algorithm is based on the evaporation of a tiny amount of water molecules on the solid surfaces with different wettability which can be studied by molecular dynamics simulations and is called as Water Evaporation Optimization (WEO). To test the performance of the proposed algorithm in an unbiased manner, WEO tested and analyzed in comparison to other existing methods on three sets of continuous test problems, a set of 17 benchmark unconstrained functions (consisting of three types of functions: unimodal, multimodal, and shifted and rotated functions), a set of 13 classical benchmark constraint functions, and three benchmark constraint engineering problems, reported in the specialized literature. The results obtained indicate that the proposed technique is highly competitive with other well-known efficient metaheuristics.

Mechanisms of WEO are carefully investigated and complete parametric studies are conducted. WEO is conceptually simple and is relatively simple to implement. In solving continuous real parameter optimization problems, both constrained and unconstrained, our proposed metaheuristic is an effective and comparable tool. The applicability of WEO to structural optimization problems (continuous or discrete optimization) are currently being investigated by the authors.

References

[1] Hare W, Nutini J, Tesfamariam S. A survey of non-gradient optimization methods in structural engineering. Adv Eng Softw 2013;59:19–28.

[2] Holland JH. Adaptation in natural and artificial systems. MIT Press; 1992. p. 211.

[3] Beyer H-G, Schwefel H-P. Evolution strategies – a comprehensive introduction. Nat Comput 2002;1(1):3–52.

[4] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220(4598):671–80.

[5] Glover F. Tabu search—Part I. ORSA J Comput 1989;1(3):190–206.

[6] Geem Zong Woo, Kim Joong Hoon, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76(2):60–8.

[7] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybernet Part B: Cybernet 1996;26(1):29–41.

[8] Kennedy J, Eberhart R. Particle swarm optimization. In: 1995 Proceedings of IEEE international conference on neural networks; 1995.

[9] Karaboga D. An idea based on honey bee swarm for numerical optimization. Erciyes University; 2005.

[10] Krishnanand KN, Ghose D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. Swarm Intell 2009;3(2):87–124.

[11] Yang X-S. Nature-inspired metaheuristic algorithms. Luniver Press; 2008. p. 128.

[12] Mucherino A, Seref O. Monkey search: a novel metaheuristic search for global optimization. AIP Conf Proc 2007;953(1):162–73.

[13] Gandomi AH, Alavi AH. Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 2012;17(12):4831–45.

[14] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw 2014;69:46–61.

[15] Askarzadeh A. Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. Commun Nonlinear Sci Numer Simul 2014;19(4):1213–28.

[16] Cuevas E et al. A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Syst Appl 2013;40(16):6374–84.

[17] Kamkar I, Akbarzadeh-T MR, Yaghoobi M. Intelligent water drops a new optimization algorithm for solving the Vehicle Routing Problem. In: 2010 IEEE international conference on systems man and cybernetics (SMC); 2010.

[18] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE congress on evolutionary computation, CEC 2007; 2007.

[19] Husseinzadeh Kashan A. League championship algorithm (LCA): an algorithm for global optimization inspired by sport championships. Appl Soft Comput 2014;16:171–200.

[20] Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179(13):2232–48.

[21] Shah-Hosseini H. Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. Int J Comput Sci Eng 2011;6(1/2):132–40.

[22] Tamura K, Yasuda K. Spiral optimization – a new multipoint search method. In: 2011 IEEE international conference on systems, man, and cybernetics (SMC); 2011.

[23] Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 2011;43(3):303–15.

[24] Erol OK, Eksin I. A new optimization method: Big Bang–Big Crunch. Adv Eng Softw 2006;37(2):106–11.

[25] Eskandar H et al. Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 2012;110–111:151–66.

[26] Ruttanateerawichien K, Kurutach W, Pichpibul T. An improved golden ball algorithm for the capacitated vehicle routing problem. In: Pan L et al., editors. Bio-inspired computing – theories and applications. Berlin, Heidelberg: Springer; 2014. p. 341–56.

[27] Eita MA, Fahmy MM. Group counseling optimization: a novel approach. In: Bramer M, Ellis R, Petridis M, editors. Research and development in intelligent systems XXVI. London: Springer; 2010. p. 195–208.

[28] Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 2009;87(5–6):267–83.

[29] Kaveh A, Talatahari S. Optimum design of skeletal structures using imperialist competitive algorithm. Comput Struct 2010;88(21–22):1220–9.

[30] Kaveh A, Bakhshpoori T, Afshari E. An efficient hybrid particle swarm and swallow swarm optimization algorithm. Comput Struct 2014;143:40–59.

[31] Kaveh A, Khayatazad M. A new meta-heuristic method: ray optimization. Comput Struct 2012;112–113:283–94.

[32] Kaveh A, Farhoudi N. A new optimization method: dolphin echolocation. Adv Eng Softw 2013;59:53–70.

[33] Kaveh A, Mahdavi VR. Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 2014;139:18–27.

[34] Kaveh A. Advances in metaheuristic algorithms for optimal design of structures. Springer International Publishing; 2014.

[35] Kaveh A, Mahdavi VR. Colliding bodies optimization. Springer International Publishing; 2015.

[36] Penman HL. Natural evaporation from open water. Bare Soil Grass 1948;193:120–45.

[37] Kohler MA, Parmele LH. Generalized estimates of free-water evaporation. Water Resour Res 1967;3(4):997–1005.

[38] Wang S et al. Evaporation of tiny water aggregation on solid surfaces with different wetting properties. J Phys Chem B 2012;116(47):13863–7.

[39] Eames IW, Marr NJ, Sabir H. The evaporation coefficient of water: a review. Int J Heat Mass Transf 1997;40(12):2963–73.

[40] Kimmel GA et al. Crystalline ice growth on Pt(111): observation of a hydrophobic water monolayer. Phys Rev Lett 2005;95(16):166102.

[41] Zarei G et al. A model for soil surface evaporation based on Campbell's retention curve. J Hydrol 2010;380(3–4):356–61.

[42] Lee M et al. Evaporation of water droplets from hydrophobic and hydrophilic nanoporous microcantilevers. Appl Phys Lett 2011;98(1):013107.

[43] Gelderblom H et al. How water droplets evaporate on a superhydrophobic substrate. Phys Rev E 2011;83(2):026306.

[44] Hong-Kai G, Hai-Ping F. Drop size dependence of the contact angle of nanodroplets. Chin Phys Lett 2005;22(4):787.

[45] Bond M, Struchtrup H. Mean evaporation and condensation coefficients based on energy dependent condensation probability. Phys Rev E 2004;70 (6):061605.

[46] Piotrowski AP. Regarding the rankings of optimization heuristics based on artificially-constructed benchmark functions. Inf Sci 2015;297: 191–201.

[47] Chen WN et al. Particle swarm optimization with an aging leader and challengers. IEEE Trans Evol Comput 2013;17(2):241–58.

[48] Suganthan PN, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real parameter optimization; 2005.

[49] Gao W, Liu S, Huang L. A global best artificial bee colony algorithm for global optimization. J Comput Appl Math 2012;236(11):2741–53.

[50] Xiang Y et al. A particle swarm inspired multi-elitist artificial bee colony algorithm for real-parameter optimization. Comput Optim Appl 2014;57 (2):493–516.

[51] Hock W, Schittkowski K. Test examples for nonlinear programming codes. J Optim Theory Appl 1980;30(1):127–9.

[52] Michalewicz Z, Schoenauer M. Evolutionary algorithms for constrained parameter optimization problems. Evol Comput 1996;4(1):1–32.

[53] Runarsson TP, Xin Y. Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 2000;4(3):284–94.

[54] Gandomi A et al. Bat algorithm for constrained optimization tasks. Neural Comput Appl 2013;22(6):1239–55.

[55] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech 2010;213(3–4):267–89.

[56] Coello Coello CA. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 2002;191(11–12):1245–87.

[57] Elsayed SM, Sarker RA, Mezura-Montes E. Self-adaptive mix of particle swarm methodologies for constrained optimization. Inf Sci 2014;277: 216–33.

[58] Coello Coello CA, Mezura Montes E. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform 2002;16(3):193–203.

[59] He Q, Wang L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 2007;20 (1):89–99.

[60] Mezura-Montes E, Coello CAC. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 2008;37(4):443–73.

[61] Kaveh A, Talatahari S. An improved ant colony optimization for constrained engineering design problems. Eng Comput 2010;27(1):155–82.