Applied Soft Computing 30 (2015) 58–71

Contents lists available at ScienceDirect

Applied Soft Computing

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

Water cycle algorithm with evaporation rate for solving constrained and unconstrained optimization problems

Ali Sadollah[a], Hadi Eskandar[b], Ardeshir Bahreininejad[c,*], Joong Hoon Kim[a]

[a] School of Civil, Environmental and Architectural Engineering, Korea University, 136-713 Seoul, South Korea
[b] Faculty of Engineering, University of Semnan, Semnan, Iran
[c] Faculty of Engineering, Institut Teknologi Brunei, BE1410 Bandar Seri Begawan, Brunei

ARTICLE INFO

Article history:
Received 13 February 2012
Received in revised form 17 January 2015
Accepted 24 January 2015
Available online 2 February 2015

Keywords:
Water cycle algorithm
Metaheuristics
Global optimization
Unconstrained
Constrained
Benchmark functions

ABSTRACT

This paper presents a modified version of the water cycle algorithm (WCA). The fundamental concepts and ideas which underlie the WCA are inspired based on the observation of water cycle process and how rivers and streams flow to the sea. New concept of evaporation rate for different rivers and streams is defined so called evaporation rate based WCA (ER-WCA), which offers improvement in search. Furthermore, the evaporation condition is also applied for streams that directly flow to sea based on the new approach. The ER-WCA shows a better balance between exploration and exploitation phases compared to the standard WCA. It is shown that the ER-WCA offers high potential in finding all global optima of multimodal and benchmark functions. The WCA and ER-WCA are tested using several multimodal benchmark functions and the obtained optimization results show that in most cases the ER-WCA converges to the global solution faster and offers more accurate results than the WCA and other considered optimizers. Based on the performance of ER-WCA on a number of well-known benchmark functions, the efficiency of the proposed method with respect to the number of function evaluations (computational effort) and accuracy of function value are represented.

1. Introduction

Among optimization methods, metaheuristic algorithms have proved their potentials in finding near optimal solutions to real life problems where exact methods may not reach the optimum solution within a reasonable computational time, especially, when the global minimum is surrounded by many local minima. The concepts of such optimizer are usually motivated by observing natural phenomena. Genetic algorithms (GAs), simulated annealing (SA), particle swarm optimization (PSO), ant colony optimization (ACO), and so forth are based on natural phenomena.

The GAs is inspired by the genetic process of biological organisms [1]. After many generations, natural populations evolve according to the principles of natural selections (i.e., survival of the fittest). In the GAs, a candidate solution is considered as a set of factors. Each design variable is symbolized by a gene. Combining the genes, a chromosome (an individual) is created which denotes a potential solution.

During the reproduction phase, the individuals are selected from the population and recombined. Parents are randomly chosen from the population using a scheme which favors the fitter individuals. By randomly selecting two parents, their chromosomes are recombined; using crossover and mutation mechanisms. For the exploration purposes, mutation is applied to some individuals, to guarantee population diversity [2]. The GAs with floating-point representation (GAF) consists of three genetic operators (selection, crossover, and mutation). Details of the GAF operators are given in the literature [3–5].

The PSO is a computation technique for solving optimization problems introduced by Kennedy and Eberhart [6]. The PSO is based on individual improvement copped with population cooperation and competition. Researchers have discovered that the synchrony of animal's behavior is through maintaining optimal distances among individual members and their neighbors [7].

Artificial bee colony (ABC) introduced by Karaboga [8] is encouraged by the behavior of honey bees when seeking for food source. Also, the ACO was inspired by the foraging behavior of real ants [9]. This behavior aids ants to find the shortest path between food sources and their nest. This performance of real ant colonies is considered in artificial ant colonies for tackling optimization problems [10,11].

The origins of SA lay in the analogy of optimization and a physical annealing process [12]. In physics, annealing is a thermal process for obtaining low-energy states of a solid in a heat bath. The idea behind the grenade explosion method (GEM) is based on observation of a grenade explosion, in which the thrown pieces of shrapnel destruct the objects near the explosion location [13].

Geem et al. [14] developed the harmony search (HS) algorithm motivated by the musical process of searching for a perfect state of harmony. The harmony in music is analogous to the optimization solution vector, and the musician's improvisations are resembled to local and global search schemes in optimization methods [15].

Bacterial foraging optimization (BFO) is based on the foraging (i.e. searching food) strategy of *Escherichia coli* bacteria [16]. In the BFO, the optimization uses the advantages of chemo-taxis, swarming, reproduction, elimination, and dispersal events to reach global minima.

* Corresponding author. Tel.: +673 2 461020; fax: +673 2 461035.
E-mail address: ardeshir.bahreininejad@itb.edu.bn (A. Bahreininejad).

http://dx.doi.org/10.1016/j.asoc.2015.01.050
1568-4946/

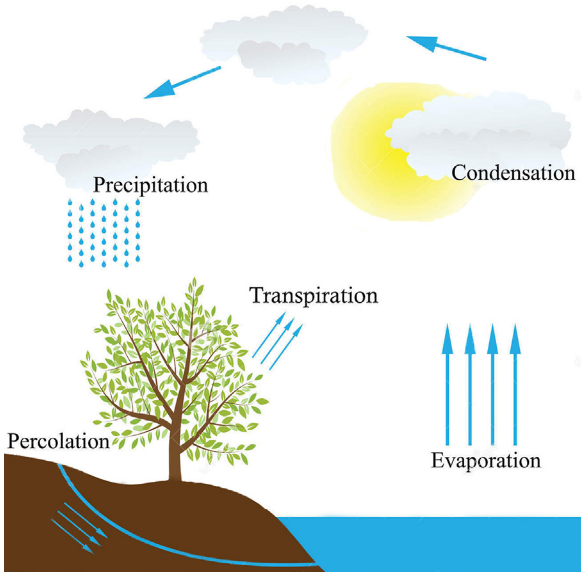

Fig. 1. A schematic view of the hydrologic cycle (water cycle process).

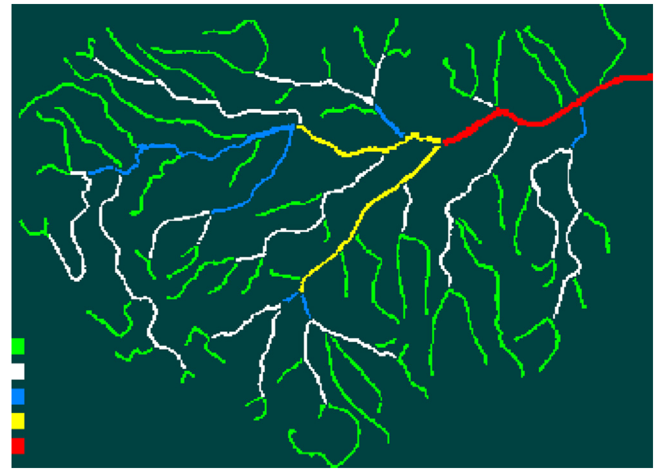

Fig. 2. Schematic diagram of streams having different orders flowing to a river. (For interpretation of the references to color in text, the reader is referred to the web version of the article.)

Adopted from U.S. Geological Service.

Shuffled complex evolution algorithm (SCE-UA) is a global optimization algorithm developed to evolve the traditional best parameter set and its underlying posterior distribution within a single optimization run [17,18].

Modified shuffled complex evolution algorithm (MSCE) has been utilized in the differential evolution (DE) to be used with the adaptation of the downhill simplex. The DE, a population-based optimizer, shows an overall superiority for a wide range of benchmark functions [19–21]. The DE combines simple operators such as recombination, mutation, and selection to progress from a randomly generated initial population.

This paper presents a modified version of water cycle algorithm (WCA) for optimizing continuous problems so called evaporation rate based WCA (ER-WCA). The WCA is based on the observation of water cycle process and how rivers and streams flow to the sea as in nature [22].

This paper is organized as follows. In Section 2, the proposed modified method and its concepts are introduced in detail. Validation of the ER-WCA is given in Section 3. In this section several unconstrained and constrained benchmark functions have been examined using the ER-WCA as follows:

1. The obtained optimization results have been compared with other reported optimizers in terms of number of function evaluations (NFEs) and function value.
2. Finding the global minimum among several local minimums (multimodal functions), and,
3. Finding all the global minima of functions having several global minima.
4. Finding the optimal feasible solutions of nonlinear constrained and engineering problems.

Finally, conclusions are drawn in Section 4.

2. The modified water cycle algorithm

2.1. Inspired idea

The idea of the WCA is inspired by nature and based on the observation of water cycle process and how rivers and streams flow downhill toward the sea in nature [22]. To further clarify, some basics of how rivers are created and water travels down to the sea are provided as follows.

A river, or a stream, is formed whenever water moves downhill from one place to another. This means that most rivers are formed high up in the mountains, where snow from the winter or ancient glaciers is melting.

Fig. 1 is a schematic view of the water cycle process, so called the hydrologic cycle. Water in rivers is evaporated, while plants discharge (transpire) water through photosynthesis process. The evaporated water is carried into the air to produce clouds which then condenses in the colder weather. Afterwards, the water returns to earth in the form of rain. This natural procedure is known as the hydrologic cycle [23].

Fig. 2 is a schematic diagram of how streams having deferent orders flow to the river. The smallest river branches are the small streams where the rivers begin to form. These tiny streams are called first-order streams (shown in Fig. 2 in green colors).

A second-order stream (shown in Fig. 2 in white colors) is produced when two first-order streams are joined. A third-order stream is formed where two second-order streams join, (shown in Fig. 2 in blue colors), and such process continues until the river flows out into the sea [24]. Finally, all of the rivers flow to the sea (i.e., the most downhill place).

2.2. Proposed evaporation rate based WCA

In the WCA, it is assumed that there are some rain or precipitation phenomena. Streams are created using the water from the rain. The WCA is a population-based algorithm; therefore, an initial population of designs variables (i.e., streams) is randomly generated between upper (UB) and lower (LB) bounds. The best individual, classified in terms of having the minimum cost function (or maximum fitness), is chosen as the sea.

Then, a number of good individuals (i.e., cost function values close to the current best solution) are chosen as rivers, while all other streams are called streams which flow to rivers and sea. In an N dimensional optimization problem, a stream is an array of $1 \times N$. This array is defined as follows:

$$\text{A Candidate stream} = [x_1,\ x_2,\ x_3,\ \cdots,\ x_N], \tag{1}$$

where N is the number of design variables (i.e., problem dimension). To initiate an optimization algorithm, an initial population demonstrating a matrix of individuals of size $N_{pop} \times N$ is created.

Therefore, the matrix X which is randomly generated is given as follows:

$$\text{Total population} = \begin{bmatrix} \text{Sea} \\ \text{River}_1 \\ \text{River}_2 \\ \text{River}_3 \\ \vdots \\ \text{Stream}_{N_{sr+1}} \\ \text{Stream}_{N_{sr+2}} \\ \text{Stream}_{N_{sr+3}} \\ \vdots \\ \text{Stream}_{N_{pop}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{N_{pop}} & x_2^{N_{pop}} & x_3^{N_{pop}} & \cdots & x_N^{N_{pop}} \end{bmatrix}, \quad (2)$$

where N_{pop} is the population size. Depending on the value of cost function (fitness function), the intensity of flow for each stream is calculated as follows:

$$\text{Cost}_i = f(x_1^i, x_2^i, \cdots, x_N^i) \quad i = 1, 2, 3, \ldots, N_{pop}. \quad (3)$$

As the first step, N_{pop} individual are generated. A number of N_{sr} from the best individuals (minimum values) are chosen as a sea and rivers. The individual which has the minimum value among others is considered as the sea. In fact, N_{sr} is the summation of number of rivers (i.e., defined by the user) and a single sea as given in Eq. (4). The rest of the population (i.e., streams which flow to the rivers or may directly flow to the sea) is calculated using Eq. (5) as given follows:

$$N_{sr} = \text{Number of rivers} + \underbrace{1}_{\text{Sea}}, \quad (4)$$

$$N_{\text{Streams}} = N_{pop} - N_{sr}. \quad (5)$$

Eq. (6) shows the population of streams which flow to the rivers or sea. Indeed, Eq. (6) is part of Eq. (2) (i.e., all individuals in population):

$$\text{Population of streams} = \begin{bmatrix} \text{Stream}_1 \\ \text{Stream}_2 \\ \text{Stream}_3 \\ \vdots \\ \text{Stream}_{N_{\text{Stream}}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{N_{\text{Stream}}} & x_2^{N_{\text{Stream}}} & x_3^{N_{\text{Stream}}} & \cdots & x_N^{N_{\text{Stream}}} \end{bmatrix}, \quad (6)$$

where N_{Streams} is the number of streams which flow directly or indirectly to the rivers or sea. Depending on flow magnitude, sea and each river absorbs streams. Hence, the amount of water entering a river and/or the sea varies. The designated streams for sea and each river are calculated using the following equations [22]:

$$C_n = \text{Cost}_n - \text{Cost}_{N_{sr+1}} \quad n = 1, 2, 3, \ldots, N_{sr}, \quad (7)$$

$$NS_n = \text{round}\left\{ \left| \frac{C_n}{\sum_{n=1}^{N_{sr}} C_n} \right| \times N_{\text{Streams}} \right\}, \quad n = 1, 2, \ldots, N_{sr}, \quad (8)$$

where NS_n is the number of streams which flow to the specific rivers or sea. Indeed, in coding WCA, the costs of sea and each river have been deducted by the cost of an individual (i.e., $N_{sr}+1$) in stream population (Eq. (6)) as can be seen in Eq. (7). Streams tend to move toward sea and rivers based on their magnitude of flow (intensity of flow). It means more streams flow into the sea than rivers. Hence, one of the best approaches to hand out the streams among sea and rivers in proportional way is to use the cost functions (fitness functions) of the sea and rivers.

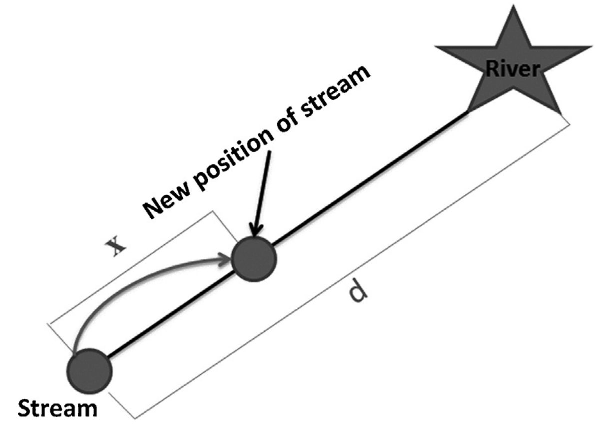

Fig. 3. Schematic view of stream's flow to the specific river (star and circle represent river and stream, respectively).

Using Eqs. (7) and (8), the best solution (i.e., sea) will be able to control and possess more streams. It is worth pointing out that streams will be randomly selected among the population of streams. Each stream is controlled by one of the best individual (i.e., sea or a river). Hence, it is not possible to assign a stream to more than one best individual.

However, for rare situations, the sum of the NS_n (Eq. (8)) may not equal to the N_{Streams}. In the source code of WCA, this issue has been solved by coding modification. In this condition, the number of streams considered for the rivers and sea is randomly increased or reduced by adding or removing value of one (i.e., ±1). Therefore, the total number of the streams assigned to the rivers and sea will become exactly equal to the N_{Streams}.

As it happens in nature, streams are created from the waters and join each other to generate new rivers. Some streams may even flow directly to the sea. All rivers and streams end up in the sea that corresponds to the current best individual.

We assume that there are N_{pop} individual of which $N_{sr} - 1$ are selected as rivers and one is selected as the sea. Fig. 3 shows the schematic view of a stream flowing toward a specific river along their connecting line.

The distance X between the stream and the river may be randomly updated as the following relation:

$$X \in (0, C \times d), \quad C > 1, \quad (9)$$

where $1 < C < 2$ and the best value for C may be chosen as 2; d is the current distance between stream and river. The value of X in relation (9) corresponds to a random number (uniformly distributed or determined from any appropriate distribution) between 0 and $(C \times d)$. Setting $C > 1$ allows streams to flow in different distances toward rivers. This concept may also be used to describe rivers flowing to the sea. Therefore, as the exploitation phase in the WCA, the new position for streams and rivers can be given in the following equations [22]:

$$\vec{X}_{\text{Stream}}(t+1) = \vec{X}_{\text{Stream}}(t) + rand \times C \times (\vec{X}_{\text{River}}(t) - \vec{X}_{\text{Stream}}(t)), \quad (10)$$

$$\vec{X}_{\text{Stream}}(t+1) = \vec{X}_{\text{Stream}}(t) + rand \times C \times (\vec{X}_{\text{Sea}}(t) - \vec{X}_{\text{Stream}}(t)), \quad (11)$$

$$\vec{X}_{\text{River}}(t+1) = \vec{X}_{\text{River}}(t) + rand \times C \times (\vec{X}_{\text{Sea}}(t) - \vec{X}_{\text{River}}(t)), \quad (12)$$

where *rand* is an uniformly distributed random number ([0,1]). Eqs. (10) and (11) are considered as updating equations for new

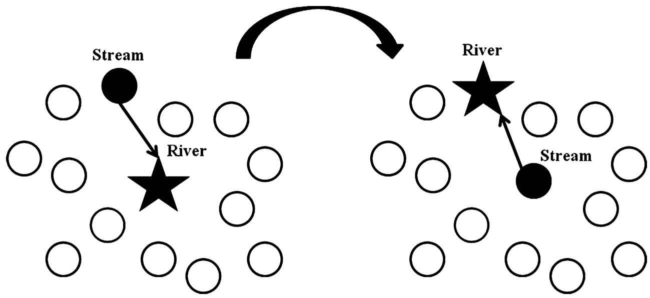

Fig. 4. Exchanging positions of the stream and river where star represents river and black circle shows the best stream among other streams.

positions of streams who flow to rivers and sea, respectively, while Eq. (12) is the updated equation for the rivers who pour to the sea. Notations having vector sign corresponds vector values, otherwise the rest of notations and parameters consider as scalar values.

If the solution obtained by a stream is improved than its connecting river, the positions of river and stream are switched (i.e., the stream becomes a river and the river becomes a stream). Therefore, the previous river act as a stream and the better stream (in term of cost function value) assigns as a new river. Indeed, the new river controls all previous streams. A similar interchange can be carried out for a river and the sea and for the streams and sea as well.

In this condition (same as interchange between river and stream), the improved river will act as new sea and the previous sea will be a new river with its own streams which was directly moved toward it. Hence, the streams related to the previous river (which is currently new updated sea) will act as the streams moving directly toward the new sea. Fig. 4 shows the exchange of a stream and river in the population.

The development of the WCA optimization process is illustrated by Fig. 5 (which also contains the concept given in Fig. 3) where circles, stars, and the diamond correspond to streams, rivers, and sea, respectively. The white (empty) shapes denote the new positions taken by streams and rivers.

Also, evaporation and raining process are defined in the original WCA to enhance its capability. To be more precise, the evaporation concept operates similar to mutation in GAs and it causes the algorithm to escape from local optima. In fact, this concept enables the algorithm to avoid premature (immature) convergence to local optima [22]. Basically, evaporation causes sea water to evaporate as rivers/streams flow into the sea. This leads to new precipitations. The evaporator operator is applied for both rivers and streams that flow into the sea. Therefore, rivers and streams should be checked whether they are close enough to the sea to make the evaporation process occur. For that purpose, the following criterion is utilized:

$$S1: \begin{array}{l} \text{if } ||\vec{X}_{\text{Sea}} - \vec{X}^i_{\text{River}}|| < d_{\max} \quad \text{or} \quad rand < 0.1 \quad i = 1, 2, \ldots, N_{sr} - 1 \\ \quad \text{Perform raining process by Eq. (13)} \\ \text{end} \end{array},$$

where $d_{\max}$ is a small number close to zero. After evaporation, the raining process is applied and new streams in the different locations are formed. For specifying the new locations of the newly formed streams, the following equation is used:

$$\vec{X}^{\text{New}}_{\text{Stream}}(t+1) = \vec{LB} + rand \times (\vec{UB} - \vec{LB}), \tag{13}$$

In the WCA, this condition should also check for the streams belong to the sea and if the evaporation condition is satisfied for a specified stream, it will move to the new location based on Eq. (13).

In the evaporation rate based WCA (ER-WCA) compared with the WCA, different raining approach is defined for streams that directly flow to the sea (see pseudo-code S2). For that purpose, the following criterion is utilized:

$$S2: \begin{array}{l} \text{if } ||\vec{X}_{\text{Sea}} - \vec{X}^i_{\text{Streams}}|| < d_{\max} \quad i = 1, 2, \ldots, NS_1 \\ \quad \text{Perform raining process by Eq. (14)} \\ \text{end} \end{array},$$

$$\vec{X}^{\text{New}}_{\text{Stream}}(t+1) = \vec{X}_{\text{Sea}}(t) + \sqrt{\mu} \times randn(1, N), \tag{14}$$

where μ is a coefficient which shows the range of searching region near the sea. *randn(1,N)* is a vector of *N* (i.e., design variables) standard Gaussian numbers. The smaller value for μ leads the algorithm to search in smaller region near the sea. A suitable value for μ is set to 0.1.

In mathematical point of view, the μ defines the concept of variance. Using such concept, the generated streams with variance μ after evaporation condition are distributed around the best obtained optimum point (sea) in the ER-WCA. In fact, this approach enhances WCA's exploitation to perform a better search around the sea.

A large value for $d_{\max}$ prevents more searches and small values encourage the search intensity near the sea. Therefore, $d_{\max}$ controls the search intensity near the sea. The value of $d_{\max}$ adaptively decreases as follows:

$$d_{\max}(t+1) = d_{\max}(t) - \frac{d_{\max}(t)}{\text{Max. iteration}} \tag{15}$$

Also, as can be seen in the nature, some rivers have low flow because a few streams pour in to these rivers. Hence, they will not be able to decrease their distance to the sea and they will evaporate after having some movements.

In order to define this concept, In the ER-WCA, the evaporation process is modified by adding the concept of evaporation rate. Hence, based on the assigned number of streams to rivers, the evaporation rate (ER) can be calculated using the following equation:

$$ER = \frac{\text{Sum}(NS_n)}{N_{sr} - 1} \times rand \quad n = 2, \ldots, N_{sr} \tag{16}$$

Eq. (16) shows the evaporation rate only used for streams and rivers. As can be seen in Eq. (16), clamorous rivers (i.e., quality solutions) with better streams have lower evaporation rate while, stagnant streams/river (i.e., low quality solutions) have higher evaporation rate. Therefore, rivers with more assigned number of streams have lower chance to evaporate compared with other rivers and their streams. Furthermore, the value of ER can be changed (slightly) at each iteration which also gives stochastic nature to the ER.

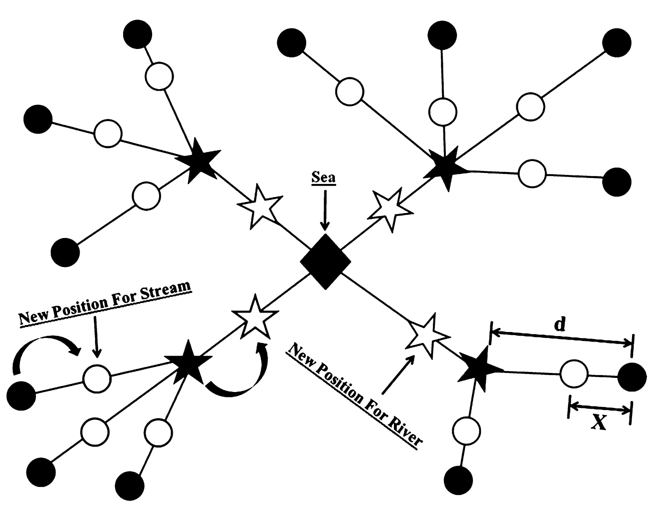

Fig. 5. A schematic view of the approaches used in the WCA.

Hence, in the ER-WCA, two types of evaporation is introduced: evaporation used between sea and streams (pseudo-code S2)/rivers (pseudo-code S1), and evaporation among rivers having a few streams (pseudo-code S3). Regarding this evaporation concept, the ER is not the only evaporation condition to be satisfied. The reason behind the *pseudo-code S3* is that rivers and their corresponding streams having low quality solutions have to be given more chances to move and flow to other high quality solutions or to find better regions in terms of better objective function value which is given as follows:

$$S3: \begin{array}{l} \text{for} \quad i = 2 : N_{sr} - 1 \\ \quad \text{if} \;\; (\exp(-k/\max\_it) < rand) \& (NS_i < ER) \\ \quad\quad \text{Perform raining process by Eq. (13)} \\ \quad \text{end} \\ \text{end} \end{array},$$

where k is the iteration index. In the ER-WCA, for evaporation among rivers, the probability of evaporation is high for low quality solutions which undergo heavy explorations at early iterations. The probability is decreased as the iterations move toward the final iteration.

To further clarify, if evaporation condition is satisfied for any river, the corresponding river together with its streams will be removed (i.e., will be evaporated). Next, the new streams, equal to the number of previous streams and a river, will be generated in new positions using Eq. (13).

In the ER-WCA, as for WCA, rivers are not considered as fixed points (see Eq. (12)) and move toward the sea (the best solution). This procedure (moving streams to the rivers and, then moving rivers to the sea) leads to indirect movement toward the best solution. In brief, the ER-WCA forces new generated streams to search near sea using the concept of variance. Variable evaporation rate used in the ER-WCA utilizes the adaptive water evaporation in the streams and rivers. Indeed, the WCA is a special case of ER-WCA having fixed evaporation rate for all rivers and streams. The occurrence of evaporation condition is decreased as the iterations continue in the ER-WCA. As for constrained handling strategy, the ER-WCA uses the same approach as used in WCA [22].

2.2.1. The steps and flowchart of ER-WCA

The steps of the ER-WCA are summarized as follows:

Step 1: Choose the initial parameters of the WCA: N_{sr}, d_{max}, N_{pop}, and maximum number of iteration.
Step 2: Generate random initial population and form the initial streams, rivers, and sea using Eqs. (2), (4), and (5).
Step 3: Calculate the cost value (for minimization problems) of each stream using Eq. (3).
Step 4: Determine the intensity of flow for rivers and sea using Eqs. (7) and (8).
Step 5: Streams flow to the rivers and sea by Eqs. (10) and (11), respectively.
Step 6: Exchange positions of rivers/sea with a stream which gives the best solution.
Step 7: Rivers flow to the sea using Eq. (12).
Step 8: Similar to Step 6, if a river finds better solution than the sea, the position of river is exchanged with the sea.
Step 9: Calculate the evaporation rate (ER) using Eq. (16).
Step 10: Check the evaporation condition among rivers and streams by pseudo-code S3 and calculate the new positions of rivers and stream using Eq. (13).
Step 11: Check the evaporation conditions between sea and streams/rivers by pseudo-codes S1 and S2 and calculate the new positions of streams and rivers using Eqs. (13) and (14).
Step 12: Reduce the d_{max} by Eq. (15).
Step 13: Check stopping condition. If the stopping criterion is satisfied, the algorithm will be stopped, otherwise return to Step 5.

2.3. WCA versus other nature-inspired algorithms

In this section, similarities and differences of WCA compared with other nature-inspired metaheuristic methods including the intelligent water drops (IWD) [25] and water wave optimization (WWO) [26] are highlighted. Every metaheuristic algorithm has its own approach and methodology in finding global optimum solution.

The three algorithms, WCA, WWO and IWD are categorized as population-based and nature-inspired metaheuristic algorithms; population of water waves in WWO, population of water drops in IDW, and population of streams in WCA. However, user parameters and operators seem different. The IWD is inspired by interaction and behavior of natural water drops in river beds [25], while the WWO's concept is based on the phenomena of water waves, such as propagation, refraction, and breaking, and inspired by shallow water wave models [26]. However, the WCA's notions are derived by the water cycle process in nature and the observation of how streams and rivers flow to the sea. The updating formulations and generation of new candidates in WCA differ from the those used in IWD and WWO.

In IWD, the idea is based on the paths with lower soils (more velocity) on its beds to the paths with higher soils (less velocity) on its beds. Also, IWD assumes water drop velocity and exchange of soil amount between river bed and water drops [25].

However, WCA does not consider stream and river beds and their amounts of soils. On the contrary, WCA utilizes a simple concept of surface runoff (i.e., the movement of streams toward rivers and rivers to the sea) as we can see in nature and water cycle process.

On other hand, the WWO considers three types of operations on the waves including propagation, refraction, and breaking. In general, the propagation and refraction operators are mostly used to create new solutions and perform more exploration in WWO, while the breaking operator conducts local search (i.e., exploitation in search space) near the best obtained solution. In WWO, different strategies and formulations are employed for using the operators in order to form new candidate solution [26] compared with the WCA.

Literary, the WCA utilizes the concept of moving indirectly from streams to the rivers and from rivers to the sea (best solution). In the WCA, rivers (a number of best selected solutions except the best one (sea)) (Eq. (12)), act as guiding individuals for directing other individuals in the population (streams) toward better positions (see Fig. 5). Rivers also move toward the sea and they are not considered as fixed individuals in the population (see Eq. (12)). In fact, this procedure, (i.e., moving streams to the rivers and, then moving rivers to the sea), leads to indirect movements toward the best solution (sea) by the WCA.

In addition, a number of near-best to best selected solutions (rivers + sea) attract other individuals of population (streams) based on their goodness of the function values (i.e., intensity of flow) using Eqs. (7) and (8) in the WCA,. However, in the IWD and WWO, these processes and strategies are not assumed.

Another difference among the WCA and considered methods is the existence of evaporation condition and raining process which corresponds to the exploration phase in WCA. The evaporation condition and raining process provide an escape mechanism for the WCA to avoid getting trapped in local optimum solutions, while in the IWD and WWO, the exploration mechanisms (formulation and concept) are different.

In terms of number of user parameters, in addition to the population size and the maximum number of iteration (i.e., assumed as common user parameters), WCA has two user-defined parameters (i.e., $d_{\max}$ and N_{sr}), while IWD and WWO have 10 (i.e., a_v, b_v, c_v, a_s, b_s, c_s, ρ_n, ρ_{IWD}, initial soil, and initial velocity) [25] and four (i.e., $h_{\max}$, α, β, and $k_{\max}$) [26] initial parameters to tune, respectively. As can be seen with respect to the number of user parameter, WCA has less user parameters which make tuning phase easier compared with the IWD and WWO.

3. Validation of the proposed algorithms

MATLAB programming software was used for coding and implementation purposes. The task of optimization was carried out on Pentium IV system 2500 GHz CPU with 4 GB RAM. For validating the proposed ER-WCA, the following criteria are considered:

- To compare the WCA with other optimization methods in terms of NFEs and best obtained function value.
- To find the global minimum among many local minima (multi-modal functions).
- To find the optimum feasible solutions for constrained problems.

In this study, a number of minimization benchmark functions have been investigated to demonstrate the efficiency of the ER-WCA and the solutions obtained are compared with the results obtained from the literature. These examples include 19 well-known unconstrained benchmark functions and 3 constrained benchmark problems. Twenty five independent optimization runs were carried out for each test problem in order to have statistically significant results. All benchmark functions (except constrained problems, see [22]) with their optimum values used in this paper are given in Appendix in Table A.1.

3.1. Benchmarks in terms of NFEs and best obtained solutions

The NFEs determine the computational time and the robustness of an algorithm. Having fewer NFEs corresponds to less time needed to reach the global optimum. This feature returns back to the structure and feature of the algorithm. The best solution represents the capability of exploration and exploitation of an optimization method. The NFEs and the best obtained solution are dependent on each other.

The NFEs is wildly used in the literature for evaluating and assessing the computational effort (time) among optimizers. It shows how many times an algorithm uses the objective function for evaluating the cost (fitness) function. Lower NFEs is more desirable which corresponds to less CPU time and, therefore, faster results.

Table 1 represents the characteristics of seven benchmark functions. In this subsection, the optimization process terminates when the difference between the maximum fitness obtained and the global optimum value is less than predefined value (i.e., 0.1% of the optimum value, or less than 0.001) [27].

Table 1
Specifications of seven unconstrained benchmark functions presented in [13].

No.	Function	N	Interval
1	De Jong	2	$[-2.048, 2.048]^N$
2	Goldstein and Price I	2	$[-2, 2]^N$
3	Branin	2	$[-5, 10]^N$
4	Martin and Gaddy	2	$[0, 10]^N$
5a	Rosenbrock	2	$[-1.2, 1.2]^N$
5b	Rosenbrock	2	$[-10, 10]^N$
5c	Rosenbrock	4	$[-1.2, 1.2]^N$
6	Hyper sphere	6	$[-5.12, 5.12]^N$
7	Shaffer	2	$[-100, 100]^N$

Table 2
Internal parameters of ER-WCA (WCA) used for optimization of benchmark functions presented in Table 1.

No.	N_{pop}	N_{sr}	$d_{\max}$	Max. iteration
1	10	3	0.01	200
2	10	3	0.01	200
3	10	3	0.01	100
4	5	2	0.01	50
5a	5	2	0.01	100
5b	10	3	0.01	500
5c	50	4	0.01	50,000
6	10	3	0.01	100
7	50	4	0.01	1000

Fig. 6 illustrates the surface plot and contour lines for the seven benchmark functions given in Table 1. Table 2 gives the initial optimization parameters used for the ER-WCA (WCA) for the considered functions in Table 1.

Tables 3 and 4 show the statistical optimization results including the worst, average, best solutions, and standard deviation (SD) for the reported benchmark functions using the WCA and ER-WCA, respectively. By observing Tables 3 and 4, ER-WCA compared with the WCA has found better statistical solutions for almost all cases.

Table 5 gives the optimization results obtained from the WCA, ER-WCA, in addition to, the deterministic simplex method (SIMPSA), the stochastic simulated annealing optimization procedure (NE-SIMPSA), the GA, the ant colony system (ACS), the ABC [27], and the GEM [13]. The best obtained results for all algorithms, (except for the WCA and ER-WCA), were directly extracted from the literature [13,27,28]. The best function evaluation in each case has been highlighted in bold in Table 5.

The "%" sign given in Table 5 refers to the percentage of closeness of the solutions obtained by optimizers compared with the optimal solution. For instance, if the "%" is equal to 100, it means the algorithm reaches the global optimum point for each independent run.

The comparison of the best obtained results reveals that in six out of nine problems, the ER-WCA has found the minimum value with the desired accuracy faster than other optimization

Table 3
Statistical optimization results for benchmark functions given in Table 1 using the WCA.

No.	Worst solution	Average solution	Best solution	SD[a]	Optimum solution
1	3906.121239	3905.940137	3905.930000	3.82E−02	3905.93
2	3.000968	3.000561	3.000020	2.55E−04	3
3	0.398717	0.398272	0.397731	3.20E−04	0.397727
4	9.29E−04	4.16E−04	5.01E−06	3.11E−04	0
5a	1.46E−02	1.34E−03	2.81E−05	3.11E−03	0
5b	9.86E−04	4.32E−04	1.12E−06	3.20E−04	0
5c	7.98E−04	2.12E−04	3.23E−07	2.29E−04	0
6	9.22E−03	6.01E−03	1.34E−07	1.81E−03	0
7	9.71E−03	1.16E−03	2.61E−05	2.58E−03	0

[a] Standard deviation.

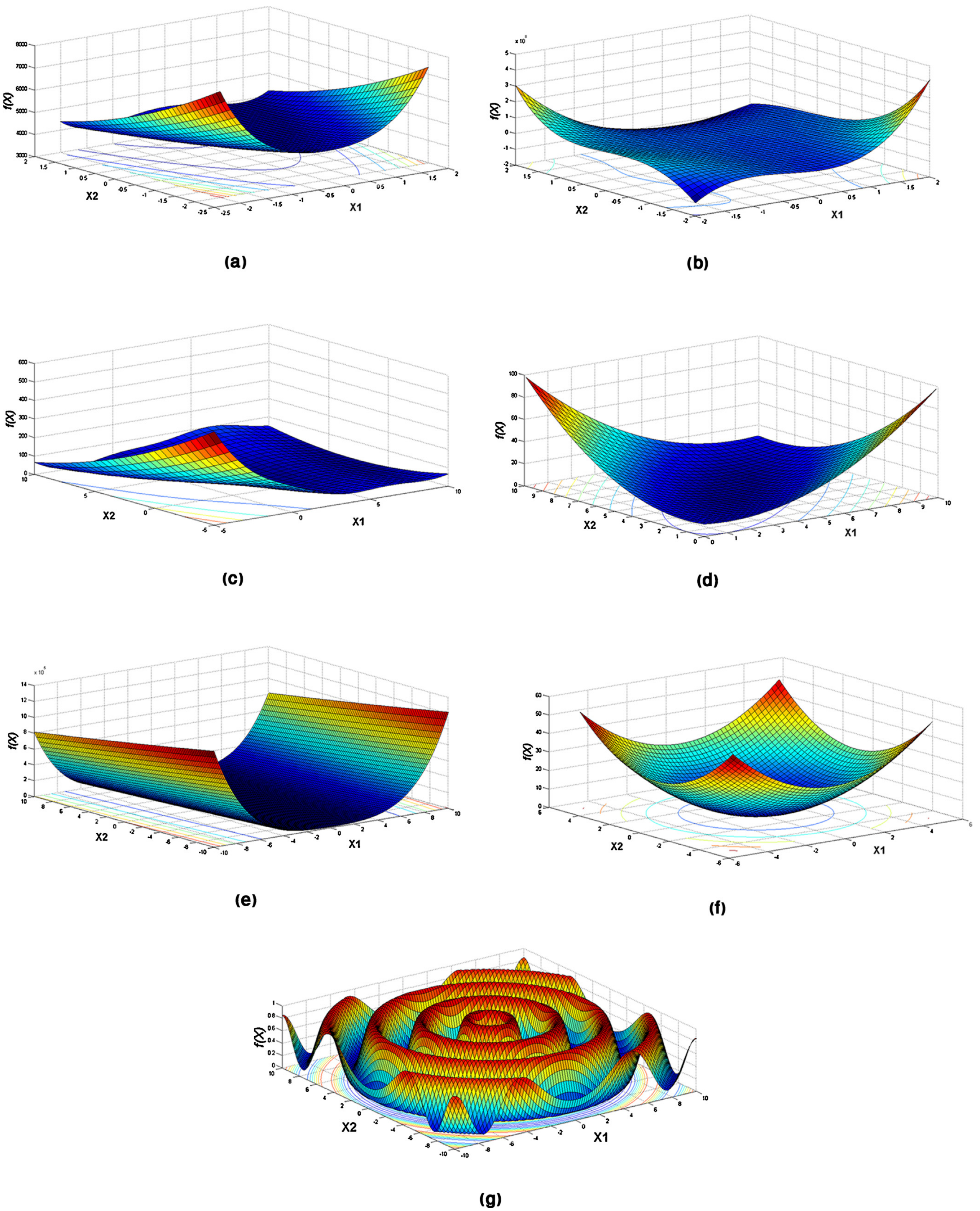

Fig. 6. Surface plot and contour lines for the seven benchmark functions presented in Table 1: (a) De Jong, (b) Goldstein and Price I, (c) Branin, (d) Martin and Gaddy, (e) Rosenbrock, (f) Hyper Sphere, and (g) Shaffer.

Table 4
Statistical optimization results for benchmark functions given in Table 1 using the ER-WCA.

No.	Worst solution	Average solution	Best solution	SD	Optimum solution
1	3905.980776	3905.934711	3905.930037	9.83E−03	3905.93
2	3.000940	3.000494	3.000013	3.08E−04	3
3	0.398693	0.398193	0.397729	2.94E−04	0.397727
4	8.75E−04	2.45E−04	2.14E−06	2.55E−04	0
5a	1.84E−02	9.35E−04	8.72E−09	3.32E−03	0
5b	9.58E−04	2.18E−04	9.66E−08	2.57E−04	0
5c	8.44E−04	2.84E−04	2.32E−09	2.65E−04	0
6	9.45E−04	2.79E−04	8.30E−09	3.02E−04	0
7	8.08E−04	2.46E−04	7.56E−07	2.62E−04	0

Table 5
Comparison of optimization results for benchmark functions presented in Table 1.

No.	SIMPSA		NE-SIMPSA		GA		ACS		ABC		GEM		WCA		ER-WCA	
	%[c]	ANFEs[b]	%	ANFEs	%	ANFEs	%	ANFEs	%	ANFEs	%	ANFEs	%	ANFEs	%	ANFEs
1	N/A[a]	N/A	N/A	N/A	100	10,160	100	6000	100	868	100	746	100	**684**	100	1220
2	N/A	N/A	N/A	N/A	100	5662	100	5330	100	999	100	701	100	980	100	**480**
3	N/A	N/A	N/A	N/A	100	7325	100	1936	100	1657	100	689	100	377	100	**160**
4	N/A	N/A	N/A	N/A	100	2488	100	1688	100	526	100	258	100	57	100	100
5a	100	10,780	100	4508	100	10,212	100	6842	100	631	100	572	100	174	100	**73**
5b	100	12,500	100	5007	N/A	N/A	100	7505	100	2306	100	2289	100	623	100	**94**
5c	99	21,177	94	3053	N/A	N/A	100	8471	100	28,529	100	82,188	100	**266**	100	300
6	N/A	N/A	N/A	N/A	100	15,468	100	22,050	100	7113	100	423	100	101	100	**91**
7	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	100	8456	100	9481	100	8942	100	**1110**

[a] Not available.
[b] Average NFEs.
[c] Success rate.

Table 6
Specification of seven unconstrained benchmark functions presented in [15].

No.	Function	Interval	N
1	Rosenbrock	[−10, 10]	2
2	Goldstein and price I	[−5, 5]	2
3	Goldstein and price II	[−5, 5]	2
4	Six hump camel back	[−10, 10]	2
5	Easton and fenton	[0, 10]	2
6	Wood	[−5, 5]	4
7	Powell quartic	[−5, 5]	4

methods (i.e., minimum ANFEs), while the WCA has been placed in the second rank. This superiority is more evident for functions 3–7.

One of the advantages of the ER-WCA/WCA is that the function values are reduced to near optimum solution in the early iterations. This may be due to the searching criteria and approach of ER-WCA/WCA where it searches a wide region of problem domain and quickly focuses on the near optimum solutions. In the following subsections, this advantage is shown for functions having higher dimensions and complexity.

Another comparison is provided to show the superiority of the proposed modified method over other optimizers. Table 6 shows the properties of seven other functions optimized using the HS [15]. The user parameters of the WCA for the functions given in Table 6 are given in Table 7.

The statistical optimization results for the seven benchmark functions from Table 6 are tabulated in Tables 8 and 9 for the WCA and ER-WCA, respectively. From Tables 8 and 9, it can be seen that the ER-WCA has found equal or more accurate solutions than the WCA.

Table 7
User parameters of ER-WCA (WCA) for benchmark functions given in Table 6.

No.	N_{pop}	N_{sr}	d_{max}	Max_iteration
1	20	4	1E−03	1000
2	10	3	1E−03	300
3	50	4	1E−03	1000
4	15	3	1E−03	500
5	10	3	1E−03	100
6	50	4	1E−03	1000
7	50	4	1E−03	1000

Table 10 demonstrates the results of optimization with respect to the NFEs and the best function value using the HS, WCA, and ER-WCA. The best solution and function evaluation in each case have been highlighted in bold in Table 10. For all functions given in Table 10, the WCA and ER-WCA show superiority over the HS in terms of NFEs except function 3. In terms of functions values, the WCA and ER-WCA have found better values than the HS for all cases.

In terms of performances of WCA and ER-WCA, looking at Table 10, the ER-WCA (with respect to the best solution accuracy) surpassed the WCA. In terms of NFEs, both algorithms have presented competitive results as shown in Table 10.

4. Multimodal benchmarks

The ER-WCA is capable of finding the global minimum of functions having many local minima without being trapped in local minima. In Section 3.1, the optimization results proved this advantage for the WCA and ER-WCA for two-dimensional problems. A further six well-known unconstrained benchmark functions are optimized using the proposed modified method to highlight the performance of the proposed method.

The multimodal functions considered are the Schwefel function, Ackley function, Rastrigin function, Sphere function, Rosenbrock function, and Zakharov function with 30 independent variables [13,29]. Table 11 presents the specifications of these benchmark functions. The global optimum point of all benchmark functions is given in Table 11 which is zero.

Functions 1–6 are high-dimensional problems. Schwefel, Ackley, Rastrigin, and Rosenbrock functions are multimodal (various optima) functions where the number of local minima increases exponentially with the problem dimension. They are considered as being the most difficult class of problems for many optimizers. It is important to mention here that the Rosenbrock function can be treated as a multimodal problem [30].

Table 8
Statistical optimization results for the benchmark functions provided in Table 6 using the WCA.

No.	Worst solution	Average solution	Best solution	SD	Optimum SOLUTION
1	4.78E−09	9.54E−10	4.52E−11	1.06E−09	0
2	3.0000	3.0000	3.0000	9.81E−07	3
3	1.1291	1.0118	1.0000	3.60E−02	1
4	−1.0316	−1.0316	−1.0316	1.38E−08	−1.0316
5	1.7441	1.7441	1.7441	1.96E−06	1.74
6	3.81E−05	1.58E−06	1.30E−10	7.60E−06	0
7	2.87E−09	6.09E−10	1.12E−11	8.29E−10	0

Table 9
Statistical optimization results for the benchmark functions provided in Table 6 using the ER-WCA.

No.	Worst solution	Average solution	Best solution	SD	Optimum solution
1	1.24E−30	3.73E−31	0.00E+0	5.80E−31	0
2	3.00E+0	3.00E+0	3.00E+0	6.56E−07	3
3	1.0374	1.0237	1.0000	1.83E−02	1
4	−1.0316	−1.0316	−1.0316	7.89E−10	−1.0316
5	1.7487	1.7444	1.7441	1.08E−03	1.74
6	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0
7	1.36E−27	4.57E−29	9.79E−35	2.48E−28	0

Table 10
Comparison of optimization results for the problems represented in Table 6.

No.	HS		WCA		ER-WCA	
	Best solution	NFEs	Best solution	NFEs	Best solution	NFEs
1	5.68E−10	50,000	4.52E−11	12,680	0.00E+0	**5780**
2	3.0000	40,000	3.0000	**2400**	3.00E+0	2960
3	1.0000	**45,000**	1.0000	47,500	1.00E+0	47,650
4	−1.0316	4870	−1.0316	**3105**	−1.0316	6825
5	1.7441	800	1.7441	**650**	1.7441	550
6	4.85E−09	70,000	1.30E−10	15,650	0.00E+0	**15,050**
7	1.25E−11	100,000	1.12E−11	23,500	9.79E−35	**19,900**

Table 11
Specifications of six unconstrained benchmark functions.

No.	Function	N	Interval
1	Schwefel	30	[−500, 500]
2	Ackley	30	[−32, 32]
3	Rastrigin	30	[−5.12, 5.12]
4	Sphere	30	[−5.12, 5.12]
5	Rosenbrock	30	[−30, 30]
6	Zakharov	30	[−10, 10]

The Rosenbrock function has a narrow parabolic-shaped deep valley from the perceived local optima to the global optimum. To find the valley is trivial, but to achieve convergence to the global minimum is a difficult task.

In order to show the complexity and difficulty of mentioned benchmark functions, Fig. 7 is given for only two-dimensional mode. By observing Fig. 7, the global minimum is surrounded among many local minima, even for the two-dimensional mode (e.g., Fig. 7a–c).

The performance of WCA and ER-WCA are compared against the GAF, the SCE-UA, the MSCE [29], the DE, the gregarious particle swarm optimizer (GPSO), and the synchronous bacterial foraging optimization (SBFO) [31]. The number of rivers (N_{sr}), population size (N_{pop}), d_{max}, and maximum NFEs for the all functions in Table 11 are 4, 50, 1E−05, and 25,000, respectively, for the ER-WCA and WCA. Table 12 shows the statistical optimization results for each benchmark function given in Table 11.

Table 13 represents the statistical optimization results for the GAF, SCE-UA and MSCE for optimization of six functions, respectively [29]. By comparing Tables 12 and 13, the WCA and ER-WCA have surpassed other algorithms in terms of NFEs for all problems.

The MSCE method used 120,000 function evaluations to found the global optimum point for functions 4 and 6 with SD equal to zero, while the ER-WCA reached the optimal solution with 34-digit and 31-digit accuracies, respectively. On the other hand, for functions 4 and 6, the NFEs for the ER-WCA were 7750 and 15,650, respectively.

The only method that could compete against the ER-WCA and WCA in terms of function value for some problems is the MSCE. Comparative performance between ER-WCA and WCA shows superiority of ER-WCA over the WCA in terms of NFEs and solutions are shown in Table 12.

Table 12
Statistical optimization results of the WCA and ER-WCA for six benchmark functions given in Table 11.

No.	Worst solution		Average solution		Best solution		SD		NFEs	
	WCA	ER-WCA	WCA	ER-WCA	WCA	ER-WCA	WCA	ER-WCA	WCA	ER-WCA
1	3.87E−04	3.81E−04	3.82E−04	3.81E−04	3.81E−04	3.81E−04	1.05E−06	8.91E−13	3050	2850
2	4.44E−15	2.91E−10	1.03E−15	3.48E−11	8.88E−16	8.88E−16	7.10E−16	7.88E−11	1900	10,700
3	4.99E−06	0.00E+0	2.00E−07	0.00E+0	2.21E−12	0.00E+0	9.99E−07	0.00E+0	20,350	7400
4	1.05E−17	1.89E−16	8.44E−19	6.66E−18	2.68E−37	1.85E−34	2.92E−18	3.45E−17	8000	7750
5	1.75E−04	6.80E−16	7.00E−06	4.25E−17	3.01E−14	0.00E+0	3.50E−05	1.39E−16	18,150	12,200
6	4.64E−11	4.23E−17	1.93E−12	1.65E−18	2.26E−36	5.50E−31	9.48E−12	7.72E−18	17,750	15,650

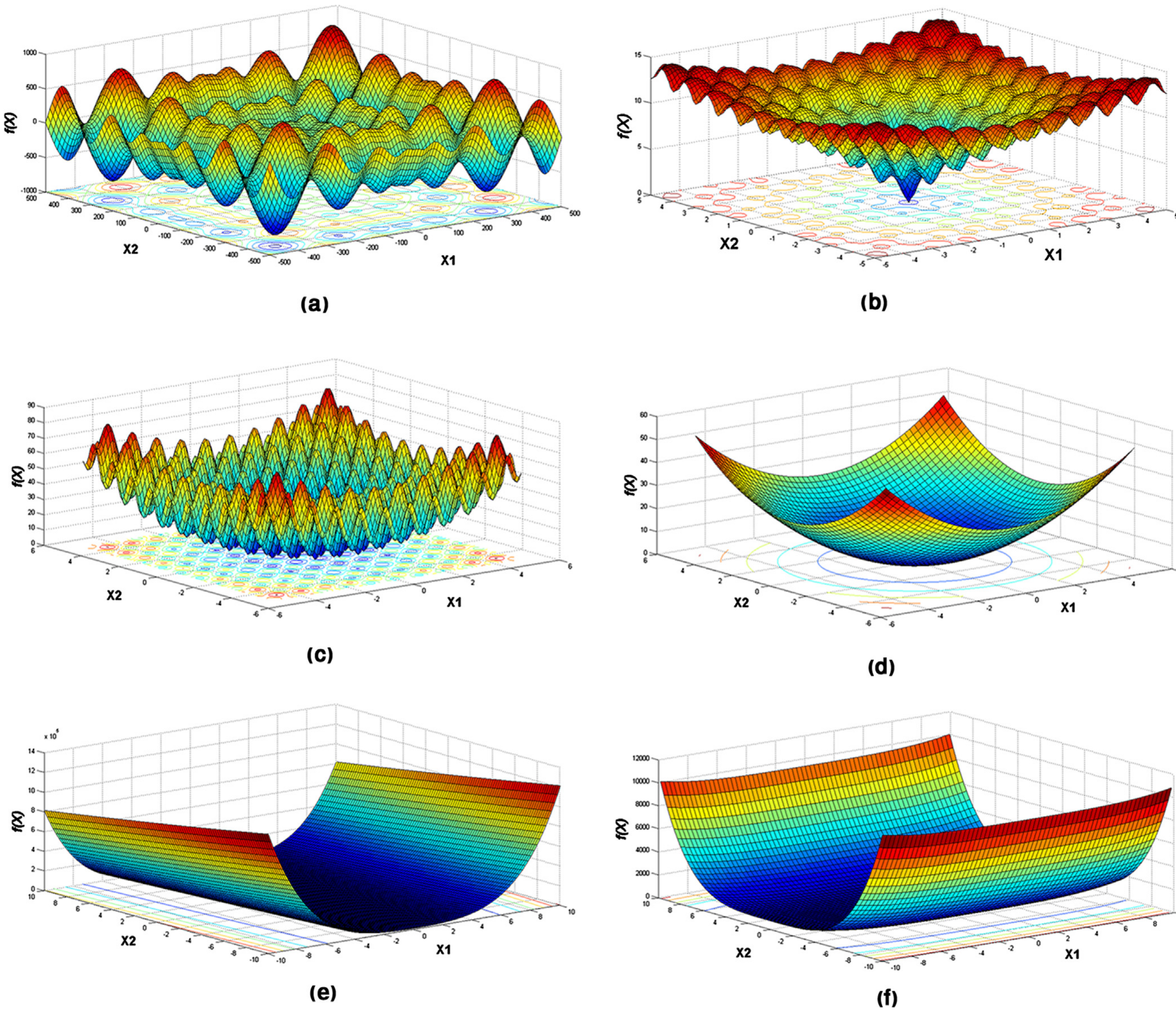

Fig. 7. Mesh plot and contour lines for six benchmark functions in two-dimension: (a) Schwefel, (b) Ackley, (c) Rastrigin, (d) Hyper Sphere, (e) Rosenbrock, and (f) Zakharov.

Table 13
Statistical optimization results using the different optimizers in [29] for the functions given in Table 11.

No.	Worst solution	Average solution	Best solution	SD	NFEs
			GAF		
1	6219.6	5434.8	3987.9	552.3	120,000
2	3.1669	1.8585	0.1209	0.6483	120,000
3	1.9902	0.2655	2.13E−13	0.5183	120,000
4	2.29E−4	4.83E−05	9.56E−11	4.29E−5	120,000
5	23.0082	51.7613	27.7946	50.6304	120,000
6	52.8072	30.9811	13.7928	10.5527	120,000
			SCE-UA		
1	8594.3853	8042.6031	7394.4199	288.5129	120,000
2	1.6462	0.1068	1.66E−04	0.3407	120,000
3	3.9798	1.5588	5.51E−09	1.1294	120,000
4	5.97E−11	5.92E−12	3.48E−16	1.212E−11	120,000
5	28.2745	27.0576	25.3911	0.6330	120,000
6	0.0393	0.0116	2.60E−04	0.0112	120,000
			MSCE		
1	6.1420	1.5598	0.1072	1.4026	120,000
2	8.88E−16	8.88E−16	8.88E−16	1.01E−15	120,000
3	3.9095	1.5270	5.32E−09	1.1216	120,000
4	0.00E+0	0.00E+0	0.00E+0	0.00E+0	120,000
5	25.9221	23.4675	20.3137	1.2133	120,000
6	0.00E+0	0.00E+0	0.00E+0	0.00E+0	120,000

Table 14
Comparison of the best obtained solution using different methods for four functions given in Table 11.

No.	SBFO		GPSO		DE		WCA		ER-WCA	
	Best	NFEs	Best	NFEs	Best	NFEs	Best	NFEs	Best	NFEs
2	5.18E−04	100,000	3.70E−02	200,000	8E−04	200,000	8.88E−16	25,000	8.88E−16	25,000
3	4.68E−04	100,000	0.13	200,000	27.43	200,000	2.21E−12	25,000	0.00E+0	25,000
4	4.68E−04	100,000	6.60E−02	200,000	3.50E−03	200,000	2.68E−37	25,000	1.85E−34	25,000
5	27.6329	100,000	2.46	200,000	34.35	200,000	3.01E−14	25,000	0.00E+0	25,000

The ER-WCA and WCA are also compared with the DE, the GPSO, and the SBFO [31]. The obtained results are compared in terms of the best obtained solution and the NFEs. Table 14 shows the comparison of results for the proposed modified method against other algorithms.

From Table 14, the results of all methods except the WCA and ER-WCA are obtained from [31]. As shown in Table 14, the WCA and ER-WCA, under predefined NFEs (i.e., 25,000), could obtain considerably better accurate solutions compared with other optimizers. Fig. 8 illustrates the reduction history for function values

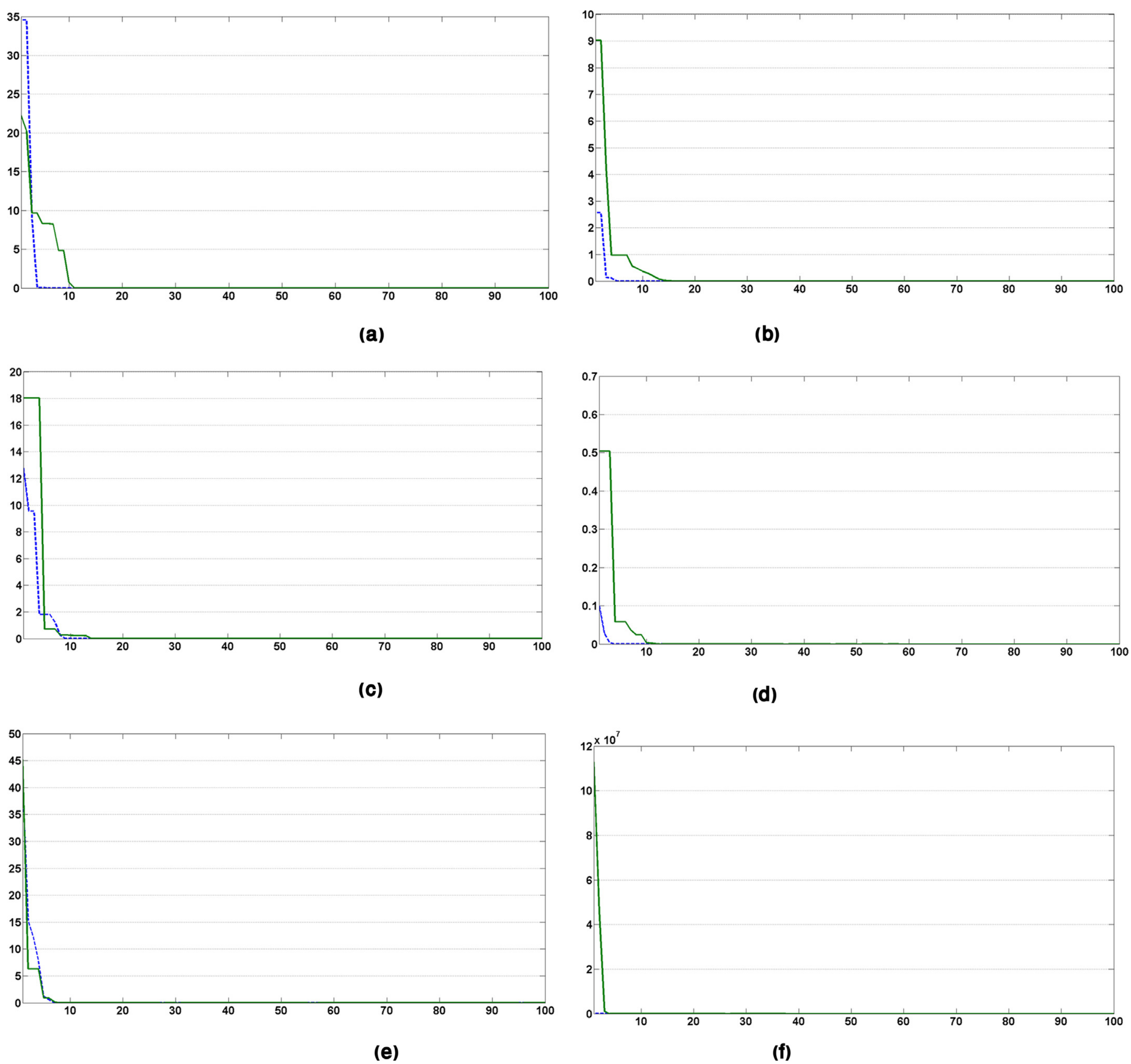

Fig. 8. Function values versus the number of iterations for six benchmark functions given in Table 11 using the WCA (dashed lines) and ER-WCA (solid lines): (a) Schwefel, (b) Ackley, (c) Rastrigin, (d) Hyper Sphere, (e) Rosenbrock, and (f) Zakharov (vertical and horizontal axes are function values and number of iterations, respectively).

Table 15
Comparison of statistical results given by various algorithms for constrained problem 3.

Methods	Worst solution	Mean solution	Best solution	SD	NFEs
CRGA	−0.9931	−0.9975	−0.9997	1.4E−03	67,600
SR	−1.0000	−1.0000	−1.0000	1.9E−04	229,000
SMES	−1.000	−1.000	−1.000	2.09E−04	240,000
DELC	−1.000	−1.000	−1.000	2.1E−06	200,000
DEDS	−1.0005	−1.0005	−1.0005	1.9E−08	225,000
HEAA	−1.000	−1.000	−1.000	5.2E−15	200,000
WCA	−0.999171	−0.999806	−0.999981	−1.91E−04	103,900
ER-WCA	−0.999837	−0.999930	−0.999994	4.30E−05	98,050

Table 16
Comparison of statistical results given by different optimizers for the pressure vessel problem.

Methods	Worst solution	Mean solution	Best solution	SD	NFEs
GA	6469.3220	6177.2533	6059.9463	130.92	80,000
HPSO	6288.6770	6099.9323	6059.7143	86.20	81,000
NM-PSO	5960.0557	5946.7901	5930.3137	9.161	80,000
WCA	6590.2129	6198.6172	5885.3327	213.04	27,500
ER-WCA	6326.7716	5933.1276	5885.3327	153.43	21,200

Table 17
Comparison of statistical results using several optimizers for the rolling element bearing problem (maximization problem).

Methods	Worst solution	Mean solution	Best solution	SD	NFEs
GA	N/A	N/A	81843.3	N/A	225,000
ABC	78,897.81	81,496	81,859.74	0.69	10,000
TLBO	80,807.85	81,438.98	81,859.74	0.66	10,000
WCA	83,942.71	83,847.16	85,538.48	488.30	3950
ER-WCA	84,780.72	85,452.93	85,539.19	175.57	2850

with respect to the number of iterations for the six benchmark functions presented in Table 11. For all six benchmark functions the first 100 iterations are depicted to show the convergence rate for the WCA and ER-WCA.

Interestingly, as can be seen in Fig. 8, the ER-WCA started from higher cost functions for almost all functions compared with the WCA. However, in near final iterations, the ER-WCA obtained better quality solutions. Indeed, the worst results are plotted in Fig. 8 for the ER-WCA, to show the exploration capability of the ER-WCA versus the WCA (best results are plotted for the WCA).

For instance, from Fig. 8f, it can be seen that the initial cost function for the ER-WCA is about 11.2E+07, while for the WCA is around 0.32. However, ER-WCA obtained better average solution compared with the WCA (see Table 12). In further iterations, the ER-WCA has tried to decrease the cost function value from order E+07 to order E−31. Fig. 8 shows a good cooperation between exploration and exploitation phases for the ER-WCA. This trend can be seen for all convergence plots obtained by the ER-WCA for the reported functions.

5. Constrained problems

The WCA has already been tested for solving constrained problems. The obtained results using the WCA for constrained problems were satisfactory and compatible compared with other considered optimizers in the literature [22]. The performance of ER-WCA for solving constrained problems is examined in this paper.

The three constrained problems used are exactly the same problems used in [22] (which are constrained problem 3, pressure vessel design problem, and rolling element bearing problem). The mathematical equations and schematic views of the considered problems are provided in the literature [22]. Tables 15–17 demonstrate statistical optimization results for the reported problems using the ER-WCA and other algorithms. All applied optimization algorithms in Tables 15–17 are explained in [22].

From Tables 15–17, the ER-WCA shows its potential for tackling constrained problems by providing better statistical optimization results in a fewer number of function evaluations compared with the considered optimizers.

6. Conclusions

This paper presented a modified version of water cycle algorithm (WCA) so called the evaporation rate based WCA (ER-WCA). The fundamental concepts behind the WCA are motivated by nature and are based on water cycle process. In brief, the new modifications applied to the WCA are listed as follows: (a) evaporation rate's concept for rivers and streams is suggested in the ER-WCA; (b) variable evaporation rate is utilized in the ER-WCA to adjust the evaporation of the water adaptively (offer more explorations in not good quality regions); (c) ER-WCA forces new generated streams to search near sea using the concept of variance; (d) In the ER-WCA, the occurrence of evaporation condition decreases as the iteration continues.

It is shown that this simple and robust algorithm has good potential in finding all global optima of multimodal and benchmark functions. The proposed modified method was tested using several unconstrained and constrained benchmark functions and the optimization results obtained from comparison show that in most cases, the ER-WCA has converged to the global minima faster and more accurate than other reported optimizers.

The ER-WCA was able to find the global minimum of multimodal functions with minimum possibility of getting trapped in local minima. As the ER-WCA is efficient in traveling toward optimal point, a hybridization of the ER-WCA with other optimization methods may be considered as further research.

Acknowledgment

This work was supported by a National Research Foundation of Korean (NRF) grant funded by the Korean government (MSIP) (NRF-2013R1A2A1A01013886).

Appendix.

Table A.1
Mathematical formulations and optimum values for considered benchmark functions.

Function	Formulation	Optimal value
Rastrigin	$\sum_{i=1}^{N}(x_i^2 - 10\cos(2\pi i x) + 10)$	0
Ackley	$-20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{N} x_i^2}{N}}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right) + 20 + e$	0
Zakharov	$\sum_{i=1}^{N} x_i^2 + \left(\sum_{i=1}^{N} 0.5 i x_i\right)^2 + \left(\sum_{i=1}^{N} 0.5 i x_i\right)^4$	0
Schwefel	$418.9829 \times N - \sum_{i=1}^{N} x_i \sin(\sqrt{\lvert x_i \rvert})$	0
Rosenbrock	$\sum_{i=1}^{N-1}\{100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\}$	0
Hyper Sphere	$\sum_{i=1}^{N} x_i^2$	0
Martin and Gaddy	$(x_1 - x_2)^2 + \left[\frac{(x_1 + x_2 - 10)}{3}\right]^2$	0
Branin	$\left(x_2 - \frac{5.1}{4}\left(\frac{7}{22}\right)^2 x_1^2 + \frac{35}{22}x_1 - 6\right)^2 + 10\left(1 - \frac{7}{176}\right)\cos(x_1) + 10$	0.397727
Goldstein and Price I	$[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [3 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3
Goldstein and Price II	$\exp\left\{\frac{1}{2}(x_1^2 + x_2^2 - 25)^2\right\} + \sin^4(4x_1 - 3x_2) + \frac{1}{2}(2x_1 + x_2 - 10)^2$	1
De Jong	$3905.93 - 100(x_1^2 - x_2)^2 - (1 - x_1)^2$	3905.93
Six Hump Camel Back	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	−1.031628
Himmelblau	$(x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$	0
Shaffer	$0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	0
Hansen	$\left(\sum_{i=1}^{5} i\cos((i-1)x_1 + i)\right) \times \left(\sum_{j=1}^{5} j\cos((j+1)x_2 + j)\right)$	−176.541793
Shubert	$\left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right) \times \left(\sum_{j=1}^{5} j\cos((j+1)x_2 + j)\right)$	−186.730908
Wood	$100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$	0
Powell Quartic	$(x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$	0
Easton and Fenton	$\left\{12 + x_1^2 + \frac{1 + x_2^2}{x_1^2} + \frac{x_1^2 x_2^2 + 100}{(x_1 x_2)^4}\right\}\left(\frac{1}{10}\right)$	1.74

References

[1] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.
[2] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[3] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, AI, Series, New York, NY, USA, 1992.
[4] Z. Michalewicz, T.D. Logan, S. Swaminathan, Evolutionary operators for continuous convex parameter spaces, in: Proceedings of 3rd Annual Conference on Evolutionary Programming, World Scientific, 1994, pp. 84–97.
[5] K.A. De Jong, An analysis of the behavior of a class of genetic adaptive systems (Ph.D. thesis), University of Michigan, Michigan, USA, 1975.
[6] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE IJCNN, Perth, Australia, 1995, pp. 1942–1948.
[7] J. Kennedy, R. Eberhart, A discrete binary version of the particle swarm algorithm, in: Proceedings of the World Multi-conference on Systematics, Cybernetics and Informatics, 1997, pp. 4104–4109.
[8] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report, Erciyes University, Engineering Faculty, Computer Engineering Department, Turkey, 2005.
[9] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evol. Comput. 1 (1997) 53–66.
[10] F. Aymerich, M. Serra, Optimization of laminate stacking sequence for maximum buckling load using the ant colony optimization (ACO) metaheuristic, Composites 39 (2008) 262–272.
[11] G.C. Luh, C.Y. Lin, Structural topology optimization using ant colony optimization algorithm, Appl. Soft Comput. 9 (2009) 1343–1353.
[12] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[13] A. Ahrari, A.A. Aatai, Grenade explosion method-A novel tool for optimization of multimodal functions, Appl. Soft Comput. 10 (2010) 1132–1140.
[14] Z. Geem, J. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2001) 60–68.
[15] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comput. Methods Appl. Mech. Eng. 194 (2005) 3902–3933.

[16] Y. Liu, K.M. Passino, Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors, J. Optimiz. Theory Appl. 115 (2002) 603–628.
[17] J.A. Vrugt, H.V. Gupta, W. Bouten, S. Sorooshian, A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrological model parameters, Water Resour. Res. 39 (2003) 1201–1218.
[18] Q. Duan, V.K. Gupta, S. Sorooshian, Effective and efficient global optimization for conceptual rainfall-runoff models, Water Resour. Res. 28 (1992) 1015–1031.
[19] M.M. Ali, Z. Kajee-Bagdadi, A local exploration-based differential evolution algorithm for constrained global optimization, Appl. Math. Comput. 208 (2009) 31–48.
[20] J. Vesterström, R. Thomsen, A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems, in: Proceedings of the Sixth Congress on Evolutionary Computation, Piscataway, NJ, USA, 2004, pp. 1980–1987.
[21] R. Ursem, R. Vadstrup, Parameter identification of induction motors using differential evolution, in: Proceedings of the 5th CEC '03, Piscataway, NJ, USA, 2003, pp. 790–796.
[22] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110–111 (2012) 151–166.
[23] S. David, The Water Cycle (John Yates, Illus), Thomson Learning, New York, 1993.
[24] A.N. Strahler, Dynamic basis of geomorphology, Geol. Soc. Am. Bull. 63 (1952) 923–938.
[25] H.S. Hosseini, Problem solving by intelligent water drops, in: IEEE CEC 2007, Singapore, 2007, pp. 3226–3231.
[26] Y.J. Zheng, Water wave optimization: a new nature-inspired metaheuristic, Comput. Oper. Res. 55 (2015) 1–11.
[27] D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, The Bees Algorithm – A Novel Tool for Complex Optimization Problems, iPROMS, Cardiff, England, 2006, pp. 454–461.
[28] A. Ahrari, M.R. Saadatmand, M. Shariat-Panahi, A.A. Atai, On the limitations of classical benchmark functions for evaluating robustness of evolutionary algorithms, Appl. Math. Comput. 215 (2010) 3222–3229.
[29] V.C. Mariani, L.G.J. Luvizotto, F.A. Guerra, L.D.S. Coelho, A hybrid shuffled complex evolution approach based on differential evolution for unconstrained optimization, Appl. Math. Comput. 217 (2011) 5822–5829.
[30] Y.W. Shang, Y.H. Qiu, A note on the extended rosenbrock function, Evol. Comput. 14 (2006) 119–126.
[31] K.M. Bakwad, S.S. Pattnaik, B.S. Sohi, S. Devi, B.K. Panigrahi, V.R.S. Gollapudi, Multimodal function optimization using synchronous bacterial foraging optimization technique, IETE J. Res. 56 (2010) 80–87.