Computers and Structures 110–111 (2012) 151–166

Contents lists available at SciVerse ScienceDirect

Computers and Structures

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

Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems

Hadi Eskandar [a], Ali Sadollah [b], Ardeshir Bahreininejad [b,*], Mohd Hamdi [b]

[a] Faculty of Engineering, Semnan University, Semnan, Iran
[b] Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 21 June 2012
Accepted 27 July 2012
Available online 26 August 2012

Keywords:
Water cycle algorithm
Metaheuristic
Global optimization
Constrained problems
Engineering design
Constraint handling

ABSTRACT

This paper presents a new optimization technique called water cycle algorithm (WCA) which is applied to a number of constrained optimization and engineering design problems. The fundamental concepts and ideas which underlie the proposed method is inspired from nature and based on the observation of water cycle process and how rivers and streams flow to the sea in the real world. A comparative study has been carried out to show the effectiveness of the WCA over other well-known optimizers in terms of computational effort (measures as number of function evaluations) and function value (accuracy) in this paper.

1. Introduction

Over the last decades, various algorithms have been developed to solve a variety of constrained engineering optimization problems. Most of such algorithms are based on numerical linear and nonlinear programming methods that may require substantial gradient information and usually seek to improve the solution in the neighborhood of a starting point. These numerical optimization algorithms provide a useful strategy to obtain the global optimum solution for simple and ideal models.

Many real-world engineering optimization problems, however, are very complex in nature and quite difficult to solve. If there is more than one local optimum in the problem, the results may depend on the selection of the starting point for which the obtained optimal solution may not necessarily be the global optimum. Furthermore, the gradient search may become unstable when the objective function and constraints have multiple or sharp peaks.

The drawbacks (efficiency and accuracy) of existing numerical methods have encouraged researchers to rely on metaheuristic algorithms based on simulations and nature inspired methods to solve engineering optimization problems. Metaheuristic algorithms commonly operate by combining rules and randomness to imitate natural phenomena [1].

The phenomena may include the biological evolutionary process such as genetic algorithms (GAs) proposed by Holland [2] and Goldberg [3], animal behavior such as particle swarm optimization (PSO) proposed by Kennedy and Eberhart [4], and the physical annealing which is generally known as simulated annealing (SA) proposed by Kirkpatrick et al. [5].

Among the optimization methods, the evolutionary algorithms (EAs) which are generally known as general purpose optimization algorithms are known to be capable of finding the near-optimum solution to the numerical real-valued test problems. EAs have been very successfully applied to constrained optimization problems [6].

GAs are based on the genetic process of biological organisms [2,3]. Over many generations, natural populations evolve according to the principles of natural selections (i.e. survival of the fittest). In GAs, a potential solution to a problem is represented as a set of parameters. Each independent design variable is represented by a gene. Combining the genes, a chromosome is produced which represents an individual (solution).

The efficiency of the different architectures of evolutionary algorithms in comparison to other heuristic techniques has been tested in both generic [7–9] and engineering design [10] problems. Recently, Chootinan and Chen [11] proposed a constraint-handling technique by taking a gradient-based repair method. The proposed technique is embedded into GA as a special operator.

PSO is a recently developed metaheuristic technique inspired by choreography of a bird flock developed by Kennedy and Eberhart [4]. The approach can be viewed as a distributed behavioral algorithm that performs a multidimensional search. It makes use of a

* Corresponding author. Tel.: +60 379675266; fax: +60 379675330.
E-mail addresses: hadi.eskandar@yahoo.com (H. Eskandar), ali_sadollah@yahoo.com (A. Sadollah), bahreininejad@um.edu.my (A. Bahreininejad), hamdi@um.edu.my (M. Hamdi).

0045-7949/$ - see front matter
http://dx.doi.org/10.1016/j.compstruc.2012.07.010

velocity vector to update the current position of each particle in the swarm.

In Ref. [12], there are some suggestions for choosing the parameters used in PSO. He and Wang [13] proposed an effective co-evolutionary PSO for constrained problems, where PSO was applied to evolve both decision factors and penalty factors. In these methods, the penalty factors were treated as searching variables and evolved by GA or PSO to the optimal values. Recently, Gomes [14] applied PSO on truss optimization using dynamic constraints.

The present paper introduces a novel metaheuristic algorithm for optimizing constrained functions and engineering problems. The main objective of this paper is to present a new global optimization algorithm for solving the constrained optimization problems. Therefore, a new population-based algorithm named as the water cycle algorithm (WCA), is proposed. The performance of the WCA is tested on several constrained optimization problems and the obtained results are compared with other optimizers in terms of best function value and the number of function evaluations.

The remaining of this paper is organized as follows: in Section 2, the proposed WCA and the concepts behind it are introduced in details. In Section 3, the performance of the proposed optimizer is validated on different constrained optimization and engineering design problems. Finally, conclusions are given in Section 4.

2. Water cycle algorithm

2.1. Basic concepts

The idea of the proposed WCA is inspired from nature and based on the observation of water cycle and how rivers and streams flow downhill towards the sea in the real world. To understand this further, an explanation on the basics of how rivers are created and water travels down to the sea is given 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 or ancient glaciers melt. The rivers always flow downhill. On their downhill journey and eventually ending up to a sea, water is collected from rain and other streams.

Fig. 1 is a simplified diagram for part of the hydrologic cycle. Water in rivers and lakes is evaporated while plants give off (transpire) water during photosynthesis. The evaporated water is carried into the atmosphere to generate clouds which then condenses in the colder atmosphere, releasing the water back to the earth in the form of rain or precipitation. This process is called the hydrologic cycle (water cycle) [15].

In the real world, as snow melts and rain falls, most of water enters the aquifer. There are vast fields of water reserves underground. The aquifer is sometimes called groundwater (see percolation arrow in Fig. 1). The water in the aquifer then flows beneath the land the same way water would flow on the ground surface (downward). The underground water may be discharged into a stream (marsh or lake). Water evaporates from the streams and rivers, in addition to being transpired from the trees and other greenery, hence, bringing more clouds and thus more rain as this cycle counties [15].

Fig. 2 is a schematic diagram of how streams flow to the rivers and rivers flow to the sea. Fig. 2 resembles a tree or roots of a tree. The smallest river branches, (twigs of tree shaped figure in Fig. 2 shown in bright green[1]), are the small streams where the rivers begins to form. These tiny streams are called first-order streams (shown in Fig. 2 in green colors).

[1] For interpretation of color in Fig. 2, the reader is referred to the web version of this article.

Wherever two first-order streams join, they make a second-order stream (shown in Fig. 2 in white colors). Where two second-order streams join, a third-order stream is formed (shown in Fig. 2 in blue colors), and so on until the rivers finally flow out into the sea (the most downhill place in the assumed world) [16].

Fig. 3 shows the Arkhangelsk city on the Dvina River. Arkhangelsk (Archangel in English) is a city in Russia that drapes both banks of the Dvina River, near where it flows into the White Sea. A typical real life stream, river, sea formation (Dvina River) is shown in Fig. 3 resembling the shape in Fig. 2.

2.2. The proposed WCA

Similar to other metaheuristic algorithms, the proposed method begins with an initial population so called the raindrops. First, we assume that we have rain or precipitation. The best individual (best raindrop) is chosen as a sea. Then, a number of good raindrops are chosen as a river and the rest of the raindrops are considered as streams which flow to the rivers and sea.

Depending on their magnitude of flow which will be described in the following subsections, each river absorbs water from the streams. In fact, the amount of water in a stream entering a rivers and/or sea varies from other streams. In addition, rivers flow to the sea which is the most downhill location.

2.2.1. Create the initial population

In order to solve an optimization problem using population-based metaheuristic methods, it is necessary that the values of problem variables be formed as an array. In GA and PSO terminologies such array is called "Chromosome" and "Particle Position", respectively. Accordingly, in the proposed method it is called "Raindrop" for a single solution. In a N_{var} dimensional optimization problem, an raindrop is an array of $1 \times N_{\text{var}}$. This array is defined as follows:

$$Raindrop = [x_1, x_2, x_3, \ldots, x_N] \tag{1}$$

To start the optimization algorithm, a candidate representing a matrix of raindrops of size $N_{\text{pop}} \times N_{\text{var}}$ is generated (i.e. population of raindrops). Hence, the matrix X which is generated randomly is given as (rows and column are the number of population and the number of design variables, respectively):

$$\text{Population of raindrops} = \begin{bmatrix} \text{Raindrop}_1 \\ \text{Raindrop}_2 \\ \text{Raindrop}_3 \\ \vdots \\ \text{Raindrop}_{N_{\text{pop}}} \end{bmatrix} = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & \cdots & x_{N_{\text{var}}}^1 \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_{N_{\text{var}}}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{N_{\text{pop}}} & x_2^{N_{\text{pop}}} & x_3^{N_{\text{pop}}} & \cdots & x_{N_{\text{var}}}^{N_{\text{pop}}} \end{bmatrix} \tag{2}$$

Each of the decision variable values $(x_1, x_2, \ldots, x_{N_{\text{var}}})$ can be represented as floating point number (real values) or as a predefined set for continuous and discrete problems, respectively. The cost of a raindrop is obtained by the evaluation of cost function (C) given as:

$$C_i = \text{Cost}_i = f(x_1^i, x_2^i, \ldots, x_{N_{\text{var}}}^i) \quad i = 1, 2, 3, \ldots, N_{\text{pop}} \tag{3}$$

where N_{pop} and N_{vars} are the number of raindrops (initial population) and the number of design variables, respectively. For the first step, N_{pop} raindrops are created. A number of N_{sr} from the best

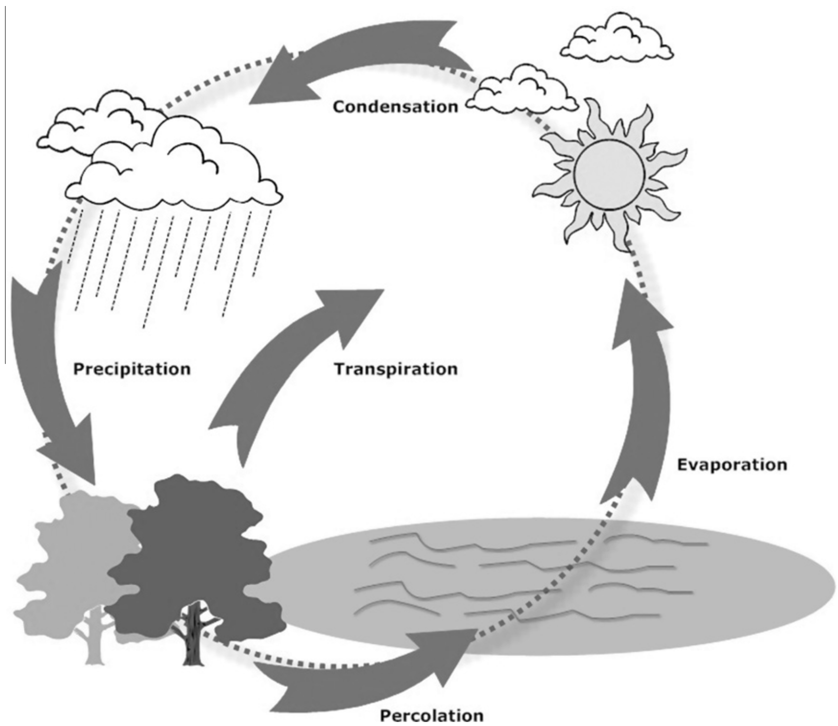

Fig. 1. Simplified diagram of the hydrologic cycle (water cycle).

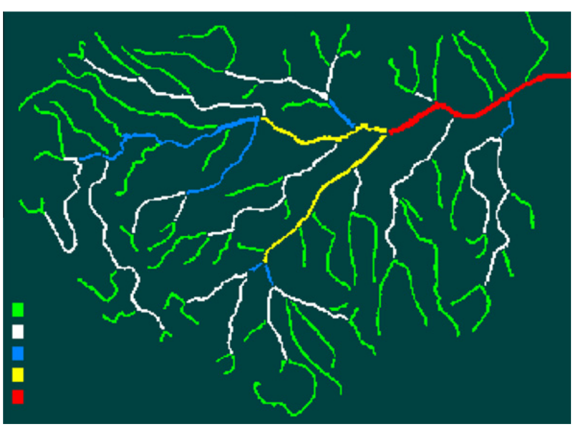

Fig. 2. Schematic diagram of how streams flow to the rivers and also rivers flow to the sea.

Fig. 3. Arkhangelsk city on the Dvina River (adopted from NASA, Image Source: http://asterweb.jpl.nasa.gov/gallery-detail.asp?name=Arkhangelsk).

individuals (minimum values) are selected as sea and rivers. The raindrop which has the minimum value among others is considered as a sea. In fact, N_{sr} is the summation of *Number of Rivers* (which is a user parameter) and a single sea as given in Eq. (4). The rest of the population (raindrops form the streams which flow to the rivers or may directly flow to the sea) is calculated using Eq. (5).

$$N_{\mathrm{sr}} = \text{Number of Rivers} + \underbrace{1}_{\text{Sea}} \tag{4}$$

$$N_{\mathrm{Raindrops}} = N_{\mathrm{pop}} - N_{\mathrm{sr}} \tag{5}$$

In order to designate/assign raindrops to the rivers and sea depending on the intensity of the flow, the following equation is given:

$$NS_n = \mathrm{round}\left\{\left|\frac{\mathrm{Cost}_n}{\sum_{i=1}^{N_{\mathrm{sr}}}\mathrm{Cost}_i}\right| \times N_{\mathrm{Raindrops}}\right\}, \quad n = 1, 2, \ldots, N_{\mathrm{sr}} \tag{6}$$

where NS_n is the number of streams which flow to the specific rivers or sea.

2.2.2. How does a stream flow to the rivers or sea?

As mentioned in subsection 2.1, the streams are created from the raindrops and join each other to form new rivers. Some of the streams may also flow directly to the sea. All rivers and streams end up in sea (best optimal point). Fig. 4 shows the schematic view of stream's flow towards a specific river.

As illustrated in Fig. 4, a stream flows to the river along the connecting line between them using a randomly chosen distance given as follow:

$$X \in (0, C \times d), \quad C > 1 \tag{7}$$

where C is a value between 1 and 2 (near to 2). The best value for C may be chosen as 2. The current distance between stream and river is represented as d. The value of X in Eq. (7) corresponds to a distributed random number (uniformly or may be any appropriate distribution) between 0 and ($C \times d$). The value of C being greater than one enables streams to flow in different directions towards the rivers.

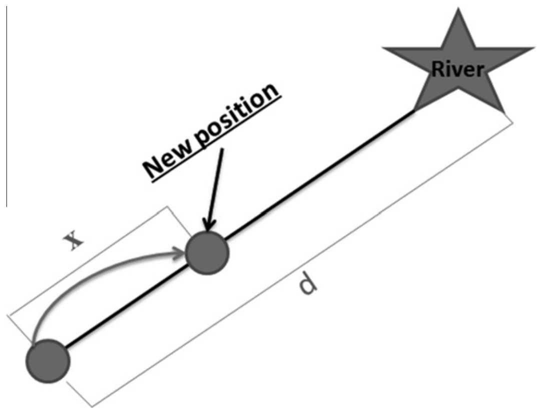

Fig. 4. Schematic view of stream's flow to a specific river (star and circle represent river and stream respectively).

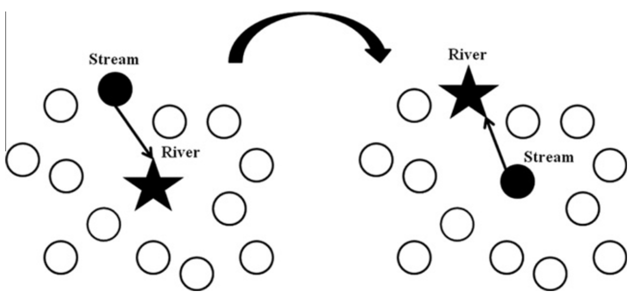

Fig. 5. Exchanging the positions of the stream and the river where star represents river and black color circle shows the best stream among other streams.

This concept may also be used in flowing rivers to the sea. Therefore, the new position for streams and rivers may be given as:

$$X^{i+1}_{\text{Stream}} = X^{i}_{\text{Stream}} + rand \times C \times \left(X^{i}_{\text{River}} - X^{i}_{\text{Stream}}\right) \tag{8}$$

$$X^{i+1}_{\text{River}} = X^{i}_{\text{River}} + rand \times C \times \left(X^{i}_{\text{Sea}} - X^{i}_{\text{River}}\right) \tag{9}$$

where *rand* is a uniformly distributed random number between 0 and 1. If the solution given by a stream is better than its connecting river, the positions of river and stream are exchanged (i.e. stream becomes river and river becomes stream). Such exchange can similarly happen for rivers and sea. Fig. 5 depicts the exchange of a stream which is best solution among other streams and the river.

2.2.3. Evaporation condition

Evaporation is one of the most important factors that can prevent the algorithm from rapid convergence (immature convergence). As can be seen in nature, water evaporates from rivers and lakes while plants give off (transpire) water during photosynthesis. The evaporated water is carried into the atmosphere to form clouds which then condenses in the colder atmosphere, releasing the water back to earth in the form of rain. The rain creates the new streams and the new streams flow to the rivers which flow to the sea [15].

This cycle which was mentioned in subsection 2.1 is called water cycle. In the proposed method, the evaporation process causes the sea water to evaporate as rivers/streams flow to the sea. This assumption is proposed in order to avoid getting trapped in local optima. The following Psuocode shows how to determine whether or not river flows to the sea.

if $\left|X^{i}_{\text{Sea}} - X^{i}_{\text{River}}\right| < d_{\max} \quad i = 1, 2, 3, \ldots, N_{\text{sr}} - 1$

Evaporation and raining process

end

where $d_{\max}$ is a small number (close to zero). Therefore, if the distance between a river and sea is less than $d_{\max}$, it indicates that the river has reached/joined the sea. In this situation, the evaporation process is applied and as seen in the nature after some adequate evaporation the raining (precipitation) will start. A large value for $d_{\max}$ reduces the search while a small value encourages the search intensity near the sea. Therefore, $d_{\max}$ controls the search intensity near the sea (the optimum solution). The value of $d_{\max}$ adaptively decreases as:

$$d^{i+1}_{\max} = d^{i}_{\max} - \frac{d^{i}_{\max}}{\text{max iteration}} \tag{10}$$

2.2.4. Raining process

After satisfying the evaporation process, the raining process is applied. In the raining process, the new raindrops form streams in the different locations (acting similar to mutation operator in GA). For specifying the new locations of the newly formed streams, the following equation is used:

$$X^{\text{new}}_{\text{Stream}} = LB + rand \times (UB - LB) \tag{11}$$

where LB and UB are lower and upper bounds defined by the given problem, respectively.

Again, the best newly formed raindrop is considered as a river flowing to the sea. The rest of new raindrops are assumed to form new streams which flow to the rivers or may directly flow to the sea.

In order to enhance the convergence rate and computational performance of the algorithm for constrained problems, Eq. (12) is used only for the streams which directly flow to the sea. This equation aims to encourage the generation of streams which directly flow to the sea in order to improve the exploration near sea (the optimum solution) in the feasible region for constrained problems.

$$X^{\text{new}}_{\text{stream}} = X_{\text{sea}} + \sqrt{\mu} \times randn(1, N_{\text{var}}) \tag{12}$$

where μ is a coefficient which shows the range of searching region near the sea. *Randn* is the normally distributed random number. The larger value for μ increases the possibility to exit from feasible region. On the other hand, 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 term $\sqrt{\mu}$ in Eq. (12) represents the standard deviation and, accordingly, μ defines the concept of variance. Using these concepts, the generated individuals with variance μ are distributed around the best obtained optimum point (sea).

2.2.5. Constraint handling

In the search space, streams and rivers may violate either the problem specific constraints or the limits of the design variables. In the current work, a modified feasible-based mechanism is used to handle the problem specific constraints based on the following four rules [17]:

- Rule 1: Any feasible solution is preferred to any infeasible solution.
- Rule 2: Infeasible solutions containing slight violation of the constraints (from 0.01 in the first iteration to 0.001 in the last iteration) are considered as feasible solutions.

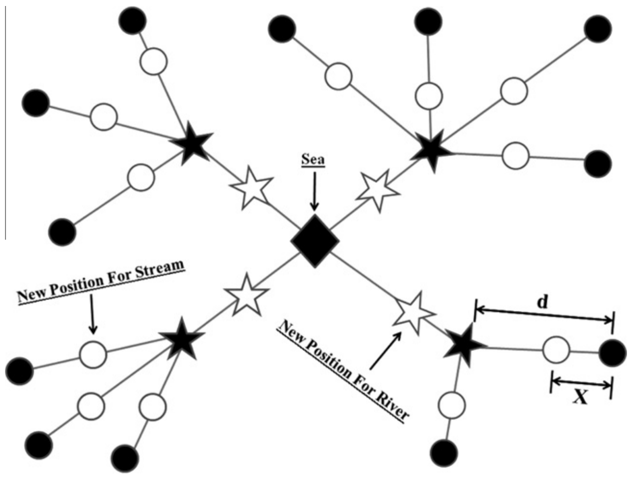

Fig. 6. Schematic view of WCA.

- Rule 3: Between two feasible solutions, the one having the better objective function value is preferred.
- Rule 4: Between two infeasible solutions, the one having the smaller sum of constraint violation is preferred.

Using the first and fourth rules, the search is oriented to the feasible region rather than the infeasible region. Applying the third rule guides the search to the feasible region with good solutions [17]. For most structural optimization problems, the global minimum locates on or close to the boundary of a feasible design space. By applying rule 2, the streams and rivers approach the boundaries and can reach the global minimum with a higher probability[18].

2.2.6. Convergence criteria

For termination criteria, as commonly considered in metaheuristic algorithms, the best result is calculated where the termination condition may be assumed as the maximum number of iterations, CPU time, or ε which is a small non-negative value and is defined as an allowable tolerance between the last two results. The WCA proceeds until the maximum number of iterations as a convergence criterion is satisfied.

2.2.7. Steps and flowchart of WCA

The steps of WCA are summarized as follows:

- Step 1: Choose the initial parameters of the WCA: N_{sr}, d_{max}, N_{pop}, max_iteration.
- Step 2: Generate random initial population and form the initial streams (raindrops), rivers, and sea using Eqs. (2), (4), and (5).
- Step 3: Calculate the value (cost) of each raindrops using Eq. (3).
- Step 4: Determine the intensity of flow for rivers and sea using Eq. (6).
- Step 5: The streams flow to the rivers by Eq. (8).
- Step 6: The rivers flow to the sea which is the most downhill place using Eq. (9).
- Step 7: Exchange positions of river with a stream which gives the best solution, as shown in Fig. 5.
- Step 8: Similar to Step 7, if a river finds better solution than the sea, the position of river is exchanged with the sea (see Fig. 5).
- Step 9: Check the evaporation condition using the Psuocode in subsection 2.2.3.
- Step 10: If the evaporation condition is satisfied, the raining process will occur using Eqs. (11) and (12).
- Step 11: Reduce the value of d_{max} which is user defined parameter using Eq. (10).
- Step 12: Check the convergence criteria. If the stopping criterion is satisfied, the algorithm will be stopped, otherwise return to Step 5.

The schematic view of the proposed method is illustrated in Fig. 6 where circles, stars, and the diamond correspond to streams, rivers, and sea, respectively. From Fig. 6, the white (empty) shapes refer to the new positions found by streams and rivers. Fig. 6 is an extension of Fig. 4.

The following paragraphs aim to offer some clarifications on the differences between the proposed WCA and other metaheuristics methods such as PSO for example.

In the proposed method, rivers (a number of best selected points except the best one (sea)) act as "guidance points" for guiding other individuals in the population towards better positions (as shown in Fig. 6) in addition to minimize or prevent searching in inappropriate regions in near-optimum solutions (see Eq. (8)).

Furthermore, rivers are not fixed points 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 move toward the best solution. The procedure for the proposed WCA is shown in Fig. 7 in the form of a flowchart.

In contrast, only individuals (particles) in PSO, find the best solution and the searching approach based on the personal and best experiences.

The proposed WCA also uses "evaporation and raining conditions" which may resemble the mutation operator in GA. The evaporation and raining conditions can prevent WCA algorithm from getting trapped in local solutions. However, PSO does not seem to posses such criteria/mechanism.

3. Numerical examples and test problems

In this section, the performance of the proposed WCA is tested by solving several constrained optimization problems. In order to validate the proposed method for constraint problems, first, four constrained benchmark problems have been applied and then, the performance of the WCA for seven engineering design problems (widely used in literatures) were examined and the optimization results were compared with other optimizers.

The benchmark problems include the objective functions of various types (quadratic, cubic, polynomial and nonlinear) with various number of the design variables, different types and number of inequality and equality constraints. The proposed algorithm was coded in MATLAB programming software and simulations were run on a Pentium V 2.53 GHz with 4 GB RAM. The task of optimizing each of the test functions was executed using 25 independent runs. The maximization problems were transformed into minimization ones as $-f(x)$. All equality constraints were converted into inequality ones, $|h(x)| - \delta \leqslant 0$ using the degree of violation $\delta = 2.2E{-}16$ that was taken from MATLAB.

For all benchmark problems (except of multiple disc clutch brake problem), the initial parameters for WCA, (N_{total}, N_{sr} and d_{max}) were chosen as 50, 8, and 1E−03, respectively. For the multiple disc clutch brake design problem, the predefined WCA user parameters were chosen as 20, 4, and 1E−03. Different iteration numbers were used for each benchmark function, with smaller iteration number for smaller number of design variables and moderate functions, while larger iteration number for large number of desicion variables and complex problems. The mathematical formulations for constrained benchmark functions (problems 1–4) are given in Appendix A. Similarly, the mathematical objective function and their constraints for the mechanical engineering design problems are presented in Appendix B.

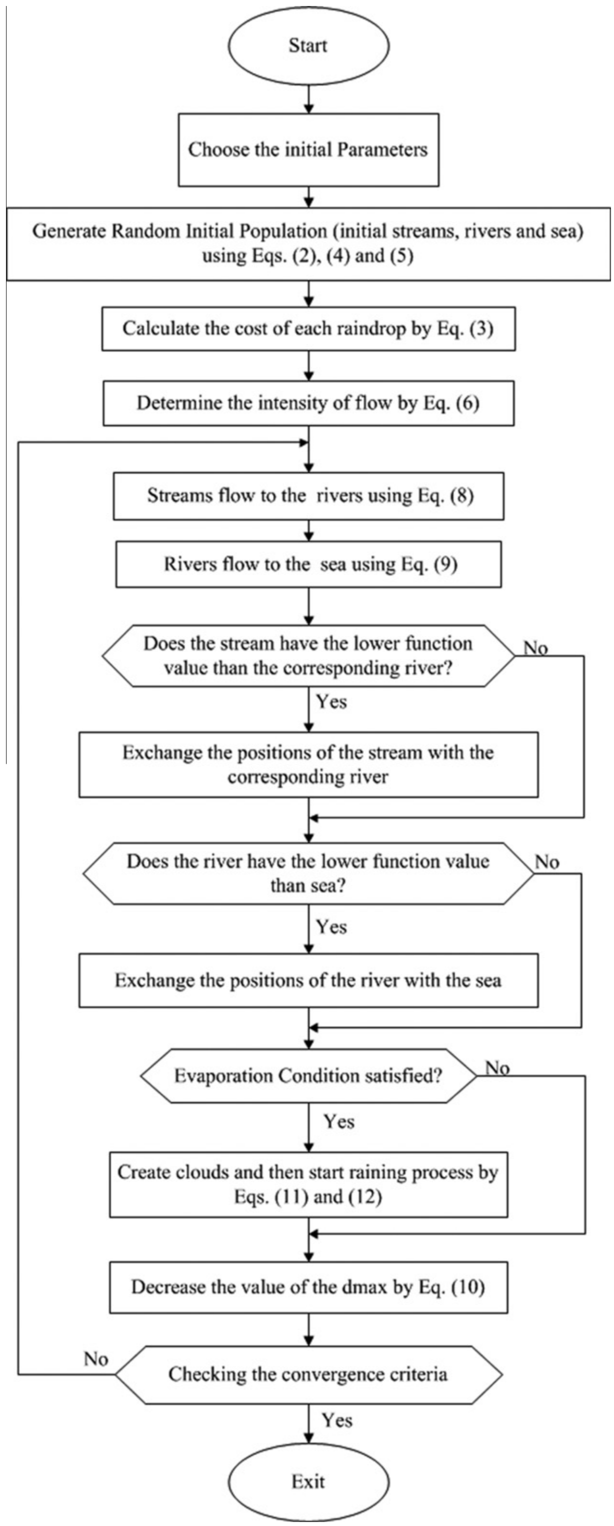

Fig. 7. Flowchart of the proposed WCA.

3.1. Constrained problem 1

For this minimization problem (see Appendix A.1), the best solution given by a number of optimizers was compared in Table 1. Table 2 presents the comparison of statistical results for the constrained problem 1 obtained using WCA and compared with previous statistical result reported by homomorphous mappings (HM) [19], adaptive segregational constraint handling evolutionary algorithm (ASCHEA) [20], improved genetic algorithm (IGA) [21], GA [11], GA1 [22], simple multi-membered evolution strategy (SMES) [23], self adaptive penalty function (SAPF) [24], PSO [25], stochastic ranking (SR) [26], differential evolution (DE) [27], cultured differential evolution (CULDE) [28], harmony search (HS) [1], coevolutionary particle swarm optimization using gaussian distribution (CPSO–GD) [29], differential evolution with dynamic stochastic selection (DEDS) [30], improved stochastic ranking (ISR) [31], hybrid evolutionary algorithm and adaptive constraint handling technique (HEAA) [32], the α constraint Simplex method (α Simplex) [33], particle evolutionary swarm optimization (PESO) [34], co-evolutionary differential evolution (CDE) [35], and artificial bee colony (ABC) [36].

Table 1
Comparison of the best solution given by previous studies for constrained problem 1.

DV	IGA [21]	HS [1]	WCA	Optimal
X_1	2.330499	2.323456	2.334238	2.330499
X_2	1.951372	1.951242	1.950249	1.951372
X_3	−0.477541	−0.448467	−0.474707	−0.477541
X_4	4.365726	4.361919	4.366854	4.365726
X_5	−0.624487	−0.630075	−0.619911	−0.624487
X_6	1.038131	1.03866	1.030400	1.038131
X_7	1.594227	1.605348	1.594891	1.594227
$g_1(X)$	4.46E−05	0.208928	1E−13	4.46E−05
$g_2(X)$	−252.561723	−252.878859	252.569346	−252.561723
$g_3(X)$	−144.878190	−145.123347	144.897817	−144.878190
$g_4(X)$	7.63E−06	−0.263414	2.2E−12	7.63E−06
$f(X)$	680.630060	680.641357	680.631178	680.630057

Table 2
Comparison of statistical results for various algorithms for constrained problem 1. “NFEs”, “SD” and “NA” stand for number of function evaluations, standard deviation, and not available, respectively.

Methods	Worst	Mean	Best	SD	NFEs
HM [19]	683.1800	681.1600	680.9100	4.11E−02	1400,000
ASCHEA [20]	NA	680.6410	680.6300	NA	1500,000
IGA [21]	680.6304	680.6302	680.6301	1.00E−05	NA
GA [11]	680.6538	680.6381	680.6303	6.61E−03	320,000
GA1 [22]	NA	NA	680.6420	NA	350,070
SAPF [24]	682.081	681.246	680.7730	0.322	500,000
SR [26]	680.7630	680.6560	680.6321	0.034	350,000
HS [1]	NA	NA	680.6413	NA	160,000
DE [27]	680.1440	680.5030	680.7710	0.67098	240,000
CULDE [28]	680.6300	680.6300	680.6300	1E−07	100,100
PSO [25]	684.5289	680.9710	680.6345	5.1E−01	140,100
CPSO−GD [29]	681.3710	680.7810	680.6780	0.1484	NA
SMES [23]	680.7190	680.6430	680.6320	1.55E−02	240,000
DEDS [30]	680.6300	680.6300	680.6300	2.9E−13	225,000
HEAA [32]	680.6300	680.6300	680.6300	5.8E−13	200,000
ISR [31]	680.6300	680.6300	680.6300	3.2E−13	271,200
α Simplex [33]	680.6300	680.6300	680.6300	2.9E−10	323,426
PESO [34]	680.6300	680.6300	680.6310	NA	350,000
CDE [35]	685.1440	681.5030	680.7710	NA	248,000
ABC [36]	680.6380	680.6400	680.6340	4E−03	240,000
WCA	680.6738	680.6443	680.6311	1.14E−02	110,050

As shown in Table 2, in terms of the number of function evaluations, the proposed method shows superiority to other algorithms (except for the CULDE method which requires 100,100 function evaluations). In terms of the optimum solution, the WCA offered better or close to the best value compared with other algorithms.

3.2. Constrained problem 2

This minimization function (see Appendix A.2) was previously solved using HM, ASCHEA, SR, cultural algorithms with evolutionary programming (CAEP) [37], hybrid particle swarm optimization (HPSO) [38], changing range genetic algorithm (CRGA) [39], DE,

Table 3
Comparison of the best solution given by various algorithms for the constrained problem 2.

DV	CULDE [28]	HS [1]	GA1 [22]	WCA	Optimal
X_1	78.000000	78.000000	78.0495	78.000000	78.00000
X_2	33.000000	33.000000	33.007000	33.000000	33.00000
X_3	29.995256	29.995000	27.081000	29.995256	29.99526
X_4	45.000000	45.000000	45.000000	45.000000	45.00000
X_5	36.775813	36.776	44.940000	36.775812	36.77581
$g_1(X)$	1.35E−08	4.34E−05	1.283813	−1.96E−12	−9.71E−04
$g_2(X)$	−92.000000	−92.000043	−93.283813	−91.999999	−92.000000
$g_3(X)$	−11.15945	−11.15949	−9.592143	−11.159499	−1.11E+01
$g_4(X)$	−8.840500	−8.840510	−10.407856	−8.840500	−8.870000
$g_5(X)$	−4.999999	−5.000064	−4.998088	−5.000000	−5.000000
$g_6(X)$	4.12E−09	6.49E−05	1.91E−03	0.000000	9.27E−09
$f(X)$	−30665.5386	−30665.5000	−31020.8590	−30665.5386	−30665.5390

Table 4
Comparison of statistical results for reported algorithms for constrained problem 2.

Methods	Worst	Mean	Best	SD	NFEs
HM [19]	−30645.9000	−30665.3000	−30664.5000	NA	1400,000
ASCHEA [20]	NA	−30665.5000	−30665.5000	NA	1500,000
SR [26]	−30665.5390	−30665.5390	−30665.5390	2E−05	88,200
CAEP [37]	−30662.2000	−30662.5000	−30665.5000	9.300	50,020
PSO [25]	−30252.3258	−30570.9286	−30663.8563	81.000	70,100
HPSO [38]	−30665.5390	−30665.5390	−30665.5390	1.70E−06	81,000
PSO−DE [25]	−30665.5387	−30665.5387	−30665.5387	8.30E−10	70,100
CULDE [28]	−30665.5386	−30665.5386	−30665.5386	1E−07	100,100
DE [27]	−30665.5090	−30665.5360	−30665.5390	5.067E−03	240,000
HS [1]	NA	NA	−30665.5000	NA	65,000
CRGA [39]	−30660.3130	−30664.3980	−30665.5200	1.600	54,400
SAPF [24]	−30656.4710	−30655.9220	−30665.4010	2.043	500,000
SMES [23]	−30665.5390	−30665.5390	−30665.5390	0.000	240,000
DELC [40]	−30665.5390	−30665.5390	−30665.5390	1.0E−11	50,000
DEDS [30]	−30665.5390	−30665.5390	−30665.5390	2.70E−11	225,000
HEAA [32]	−30665.5390	−30665.5390	−30665.5390	7.40E−12	200,000
ISR [31]	−30665.5390	−30665.5390	−30665.5390	1.10E−11	192,000
α Simplex [33]	−30665.5390	−30665.5390	−30665.5390	4.20E−11	305,343
WCA	−30665.4570	−30665.5270	−30665.5386	2.18E−02	18,850

CULDE, particle swarm optimization with differential evolution (PSO–DE) [25], PSO, HS, SMES, SAPF, differential evolution with level comparison (DELC) [40], DEDS, ISR, HEAA, and α Simplex method.

Table 3 compares the reported best solutions for the CULDE, HS, GA1, and WCA. The statistical results of different algorithms are given in Table 4. From Table 4, the PSO method found the best solution (−30663.8563) in 70,100 function evaluations. The proposed WCA reached the best solution (−30665.5386) in 18,850 function evaluations. The WCA offered modest solution quality in less number of function evaluations for this problem.

3.3. Constrained problem 3

This minimization problem (see Appendix A.3) has n decision variables and one equality constraint. The optimal solution of the problem is at $X^* = (\frac{1}{\sqrt{n}}, \cdots, \frac{1}{\sqrt{n}})$ with a corresponding function value of $f(x) = 1$. For this problem n is considered equal to 10. This function was previously solved using HM, ASCHEA, PSO–DE, PSO, CULDE, SR, DE, SAPF, SMES, GA, CRGA, DELC, DEDS, ISR, HEAA, α Simplex method, and PESO.

The comparison of best solutions is shown in Table 5. The statistical results of seventeen algorithms including the WCA are shown in Table 6. From Table 6, the WCA obtained its best solution in 103,900 function evaluations (which is considerably less than other compared algorithms except for the CRGA). However, the statistical results obtained by WCA are more accurate than the CRGA.

Table 5
Comparison of best solution for the constrained problem 3 given by two algorithms.

DV	CULDE [28]	WCA	Optimal solution
X_1	0.304887	0.316011	0.316227
X_2	0.329917	0.316409	0.316227
X_3	0.319260	0.315392	0.316227
X_4	0.328069	0.315872	0.316227
X_5	0.326023	0.316570	0.316227
X_6	0.302707	0.316209	0.316227
X_7	0.305104	0.316137	0.316227
X_8	0.315312	0.316723	0.316227
X_9	0.322047	0.316924	0.316227
X_{10}	0.309009	0.316022	0.316227
$h(X)$	9.91E−04	0.000000	0
$f(X)$	−0.995413	−0.999981	−1

3.4. Constrained problem 4

For this maximization problem (see Appendix A.4) which is converted to the minimization problem, the feasible region of the search space consists of 729 disjoint spheres. A point (x_1, x_2, x_3) is feasible, if and only if, there exist p, q, and r such that the inequality holds, as given in Appendix A [41]. For this problem, the optimum solution is $X^* = (5,5,5)$ with $f(X^*) = -1$.

This problem was previously solved using HM, SR, CULDE, CAEP, HPSO, ABC, PESO, CDE, SMES, and teaching-learning-based optimization (TLBO) [42]. The statistical results of eleven optimizers including the WCA are shown in Table 7. From Table 7, the WCA

Table 6
Comparison of statistical results given by various algorithms for constrained problem 3.

Methods	Worst	Mean	Best	SD	NFEs
HM [19]	−0.997800	−0.998900	−0.999700	NA	1400,000
ASCHEA [20]	NA	−0.999890	−1.000000	NA	1500,000
PSO [25]	−1.004269	−1.004879	−1.004986	1.0*E*+0	140,100
PSO−DE [25]	−1.005010	−1.005010	−1.005010	3.8*E*−12	140,100
CULDE [28]	−0.639920	−0.788635	−0.995413	0.115214	100,100
CRGA [39]	−0.993100	−0.997500	−0.999700	1.4*E*−03	67,600
SAPF [24]	−0.887000	−0.964000	−1.000000	3.01*E*−01	500,000
SR [26]	−1.00000	−1.000000	−1.000000	1.9*E*−04	229,000
ISR [31]	−1.001000	−1.001000	−1.001000	8.2*E*−09	349,200
DE [27]	−1.025200	−1.025200	−1.025200	NA	8000,000
SMES [23]	−1.000000	−1.000000	−1.000000	2.09*E*−04	240,000
GA [11]	−0.999790	0.999920	0.999980	5.99*E*−05	320,000
DELC [40]	−1.000000	−1.000000	−1.000000	2.1*E*−06	200,000
DEDS [30]	−1.000500	−1.000500	−1.000500	1.9*E*−08	225,000
HEAA [32]	−1.000000	−1.000000	−1.000000	5.2*E*−15	200,000
α Simples [33]	−1.000500	−1.000500	−1.000500	8.5*E*−14	310,968
PESO [34]	−0.464000	−0.764813	−0.993930	NA	350,000
WCA	−0.999171	−0.999806	−0.999981	−1.91*E*−04	103,900

Table 7
Comparison of statistical results given by various algorithms for constrained problem 4.

Methods	Worst	Mean	Best	SD	NFEs
HM [19]	−0.991950	−0.999135	−0.999999	NA	1400,000
SR [26]	−1	−1	−1	0	350,000
CAEP [37]	−0.996375	−0.996375	−1	9.7*E*−03	50,020
HPSO [38]	−1	−1	−1	1.6*E*−15	81,000
CULDE [28]	−1	−1	−1	0	100,100
SMES [23]	−1	−1	−1	0	240,000
PESO [34]	−0.994	−0.998875	−1	NA	350,000
CDE [35]	−1	−1	−1	0	248,000
ABC [36]	−1	−1	−1	0	240,000
TLBO [42]	−1	−1	−1	0	50,000
WCA	−0.999998	−0.999999	−0.999999	2.51*E*−07	6100

reached its best solution considerably faster than other reported algorithms using 6100 function evaluations.

3.5. Constrained benchmark engineering and mechanical design problems

3.5.1. Three-bar truss design problem

The three-bar truss problem (see Appendix B.1) is one of the engineering test problems for constrained algorithms. The comparison of the best solution among the WCA, DEDS, and PSO–DE is presented in Table 8. The comparison of obtained statistical results for the WCA with previous studies including DEDS, PSO–DE, HEAA, and society and civilization algorithm (SC) [43] is presented in Table 9. The proposed WCA obtained the best solution in 5250 function evaluations which is superior to other considered algorithms.

3.5.2. Speed reducer design problem

In this constrained optimization problem (see Appendix B.2), the weight of speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts, and stresses in the shafts [44]. The variables x_1 to x_7 represent the face width (b), module of teeth (m), number of teeth in the pinion (z), length of the first shaft between bearings (l_1), length of the second shaft between bearings (l_2), and the diameter of first (d_1) and second shafts (d_2), respectively. This is an example of a mixed integer programming problem. The third variable x_3 (number of teeth) is of integer values while all left variables are continuous.

Table 8
Comparison of the best solution obtained from the previous algorithms for three-bar truss problem.

DV	DEDS [30]	PSO-DE [25]	WCA
X_1	0.788675	0.788675	0.788651
X_2	0.408248	0.408248	0.408316
$g_1(X)$	1.77*E*−08	−5.29*E*−11	0.000000
$g_2(X)$	−1.464101	−1.463747	−1.464024
$g_3(X)$	−0.535898	−0.536252	−0.535975
$f(X)$	263.895843	263.895843	263.895843

Table 9
Comparison of statistical results obtained from various algorithms for the three-bar truss problem.

Methods	Worst	Mean	Best	SD	NFEs
SC [43]	263.969756	263.903356	263.895846	1.3*E*−02	17,610
PSO–DE [25]	263.895843	263.895843	263.895843	4.5*E*−10	17,600
DEDS [30]	263.895849	263.895843	263.895843	9.7*E*−07	15,000
HEAA [32]	263.896099	263.895865	263.895843	4.9*E*−05	15,000
WCA	263.896201	263.895903	263.895843	8.71*E*−05	5250

This engineering problem previously was optimized using SC, PSO–DE, DELC, DEDS, HEAA, $(\mu + \lambda) - $ ES [44], modified differential evolution (MDE) [45,46], and ABC. The comparison of the best solution of reported methods is presented in Table 10. The statistical results of six algorithms were compared with the proposed WCA and are given in Table 11. The WCA, DELC and DEDS outperformed other considered optimization engines as shown in Table 11. In terms of the number of function evaluations, the WCA method reached the best solution faster than other reported algorithms using 15,150 function evaluations.

3.5.3. Pressure vessel design problem

In pressure vessel design problem (see Appendix B.3), proposed by Kannan and Kramer [47], the target is to minimize the total cost, including the cost of material, forming, and welding. A cylindrical vessel is capped at both ends by hemispherical heads as shown in Fig. 8. There are four design variables in this problem: T_s (x_1, thickness of the shell), T_h (x_2, thickness of the head), R (x_3, inner radius), and L (x_4, length of the cylindrical section of the vessel). Among the four design variables, T_s and T_h are expected to be integer multiples of 0.0625 *in*, and R and L are continuous design variables.

Table 12 shows the comparisons of the best solutions obtained by the proposed WCA and other compared methods. This problem has been solved previously using GA based co-evolution model (GA2) [48], GA through the use of dominance-based tour tournament selection (GA3) [49], CPSO, HPSO, hybrid nelder-mead simplex search and particle swarm optimization (NM–PSO) [41], gaussian quantum-behaved particle swarm optimization (G-QPSO), quantum-behaved particle swarm optimization (QPSO) [50], PSO, and CDE and compared with the proposed WCA as given in Table 13.

From Table 13, the WCA was executed for obtaining two sets of statistical results for comparative study. The first set of statistical results was obtained for 27,500 function evaluations using and the second set of statistical results was obtained for 8000 function evaluations using WCA. As can be seen from Table 13, in terms of the best solution and number of function evaluations, the proposed method is superior to other optimizer.

Considering the statistical and comparison results in Table 13, it can be concluded that the WCA is more efficient than the other optimization engines for the pressure vessel design problem, in

Table 10
Comparison of the best solution obtained from the previous algorithms for the speed reducer problem.

DV	DEDS [30]	DELC [40]	HEAA [32]	MDE[45,46]	PSO−DE [25]	WCA
X_1	3.5 + 09	3.5 + 09	3.500022	3.500010	3.500000	3.500000
X_2	0.7 + 09	0.7 + 09	0.700000	0.700000	0.700000	0.700000
X_3	17.000000	17.000000	17.000012	17.000000	17.000000	17.000000
X_4	7.3 + 09	7.3 + 09	7.300427	7.300156	7.300000	7.300000
X_5	7.715319	7.715319	7.715377	7.800027	7.800000	7.715319
X_6	3.350214	3.350214	3.350230	3.350221	3.350214	3.350214
X_7	5.286654	5.286654	5.286663	5.286685	5.2866832	5.286654
$f(X)$	2994.471066	2994.471066	2994.499107	2996.356689	2996.348167	2994.471066

Table 11
Comparison of statistical results obtained from various algorithms for the speed reducer problem.

Methods	Worst	Mean	Best	SD	NFEs
SC [43]	3009.964736	3001.758264	2994.744241	4.0	54,456
PSO–DE [25]	2996.348204	2996.348174	2996.348167	6.4E−06	54,350
DELC [40]	2994.471066	2994.471066	2994.471066	1.9E−12	30,000
DEDS [30]	2994.471066	2994.471066	2994.471066	3.6E−12	30,000
HEAA [32]	2994.752311	2994.613368	2994.499107	7.0E−02	40,000
MDE [45,46]	NA	2996.367220	2996.356689	8.2E−03	24,000
$(\mu + \lambda)$ − ES [44]	NA	2996.348000	2996.348000	0.0	30,000
ABC [36]	NA	2997.058000	2997.058000	0.0	30,000
WCA	2994.505578	2994.474392	2994.471066	7.4E−03	15,150

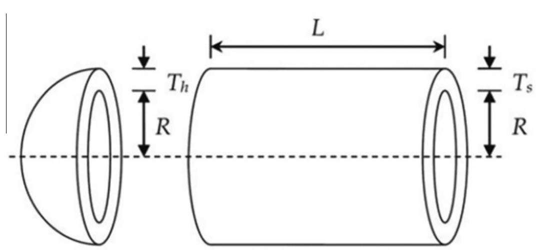

Fig. 8. Schematic view of pressure vessel problem.

this paper. Fig. 9 depicts the function values versus the number of iterations for the pressure vessel design problem.

One of the advantages of the proposed method is that the function values are reduced to near optimum point in the early iterations (see Fig. 9). This may be due to the searching criteria and constraint handling approach of WCA where it initially searches a wide region of problem domain and rapidly focuses on the optimum solution.

3.5.4. Tension/compression spring design problem

The tension/compression spring design problem (see Appendix B.4) is described in Arora [51] for which the objective is to minimize the weight ($f(x)$) of a tension/compression spring (as shown in Fig. 10) subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The independent variables are the wire diameter $d(x_1)$, the mean coil diameter $D(x_2)$, and the number of active coils $P(x_3)$.

The comparisons of the best solutions among several reported algorithms are given in Table 14. This problem has been used as a benchmark problem for testing the efficiency of numerous optimization methods, such as GA2, GA3, CAEP, CPSO, HPSO, NM–PSO, G-QPSO, QPSO, PSO–DE, PSO, DELC, DEDS, HEAA, SC, DE, ABC, and $(\mu + \lambda)$ − ES. The obtained statistical results using the reported optimizers and the proposed WCA are given in Table 15.

For comparison of statistical results obtained by WCA, two sets of statistical results are presented as shown in Table 15. The first set of results corresponds to 11,750 function evaluations and the second set corresponds to 2000 function evaluations using WCA. The best function value is 0.012630 with 80,000 function evaluations obtained by NM–PSO. From Table 15, the proposed WCA produced equally best result in the same number of function evaluations as G-QPSO method. In terms of solution quality, only NM–PSO was superior to all other considered algorithms (including WCA). However, WCA and G-QPSO offer competitive best solutions in much less number of function evaluations than offered by the NM–PSO method.

Fig. 11 demonstrates the function values with respect to the number of iterations for the tension/compression spring design problem. From Fig. 11, the initial population of the algorithm

Table 12
Comparison of the best solution obtained from various studies for the pressure vessel problem.

DV	CDE [35]	GA3 [49]	CPSO [29]	HPSO [38]	NM–PSO [41]	G-QPSO [50]	WCA
X_1	0.8125	0.8125	0.8125	0.8125	0.8036	0.8125	0.7781
X_2	0.4375	0.4375	0.4375	0.4375	0.3972	0.4375	0.3846
X_3	42.0984	42.0974	42.0913	42.0984	41.6392	42.0984	40.3196
X_4	176.6376	176.6540	176.7465	176.6366	182.4120	176.6372	−200.0000
$g_1(X)$	−6.67E−07	−2.01E−03	−1.37E−06	−8.80E−07	3.65E−05	−8.79E−07	−2.95E−11
$g_2(X)$	−3.58E−02	−3.58E−02	−3.59E−04	−3.58E−02	3.79E−05	−3.58E−02	−7.15E−11
$g_3(X)$	−3.705123	−24.7593	−118.7687	3.1226	−1.5914	−0.2179	−1.35E−06
$g_4(X)$	−63.3623	−63.3460	−63.2535	−63.3634	−57.5879	−63.3628	−40.0000
$f(X)$	6059.7340	6059.9463	6061.0777	6059.7143	5930.3137	6059.7208	5885.3327

Table 13
Comparison of statistical results given by different optimizers for the pressure vessel problem.

Methods	Worst	Mean	Best	SD	NFEs
GA2 [48]	6308.4970	6293.8432	6288.7445	7.4133	900,000
GA3 [49]	6469.3220	6177.2533	6059.9463	130.9297	80,000
CPSO [29]	6363.8041	6147.1332	6061.0777	86.4500	240,000
HPSO [38]	6288.6770	6099.9323	6059.7143	86.2000	81,000
NM–PSO [41]	5960.0557	5946.7901	5930.3137	9.1610	80,000
G-QPSO [50]	7544.4925	6440.3786	6059.7208	448.4711	8000
QPSO [50]	8017.2816	6440.3786	6059.7209	479.2671	8000
PSO [25]	14076.3240	8756.6803	6693.7212	1492.5670	8000
CDE [35]	6371.0455	6085.2303	6059.7340	43.0130	204,800
WCA	6590.2129	6198.6172	5885.3327	213.0490	27,500
	7319.0197	6230.4247	5885.3711	338.7300	8000

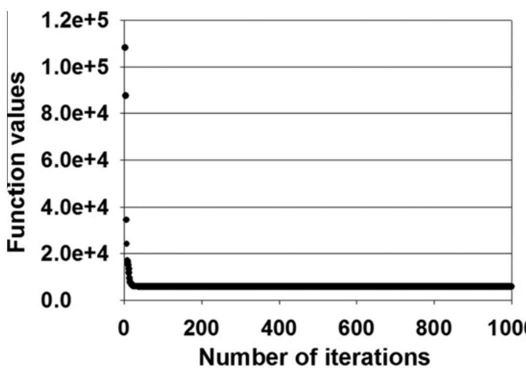

Fig. 9. Function values versus number of iterations for the pressure vessel problem.

Fig. 10. Schematic view of tension/compression spring problem.

Table 15
Comparisons of statistical results obtained from various algorithms for the tension/compression spring problem.

Methods	Worst	Mean	Best	SD	NFEs
GA2 [48]	0.012822	0.012769	0.012704	3.94E–05	900,000
GA3 [49]	0.012973	0.012742	0.012681	5.90E–05	80,000
CAEP [37]	0.015116	0.013568	0.012721	8.42E–04	50,020
CPSO [29]	0.012924	0.012730	0.012674	5.20E–04	240,000
HPSO [38]	0.012719	0.012707	0.012665	1.58E–05	81,000
NM–PSO [41]	0.012633	0.012631	0.012630	8.47E–07	80,000
G-QPSO [50]	0.017759	0.013524	0.012665	0.001268	2000
QPSO [50]	0.018127	0.013854	0.012669	0.001341	2000
PSO [25]	0.071802	0.019555	0.012857	0.011662	2000
DE [27]	0.012790	0.012703	0.012670	2.7E–05	204,800
DELC [40]	0.012665	0.012665	0.012665	1.3E–07	20,000
DEDS [30]	0.012738	0.012669	0.012665	1.3E–05	24,000
HEAA [32]	0.012665	0.012665	0.012665	1.4E–09	24,000
PSO–DE [25]	0.012665	0.012665	0.012665	1.2E–08	24,950
SC [43]	0.016717	0.012922	0.012669	5.9E–04	25,167
$(\mu+\lambda)$–ES [44]	NA	0.013165	0.012689	3.9E–04	30,000
ABC [36]	NA	0.012709	0.012665	0.012813	30,000
WCA	0.012952	0.012746	0.012665	8.06E–05	11,750
	0.015021	0.013013	0.012665	6.16E–04	2000

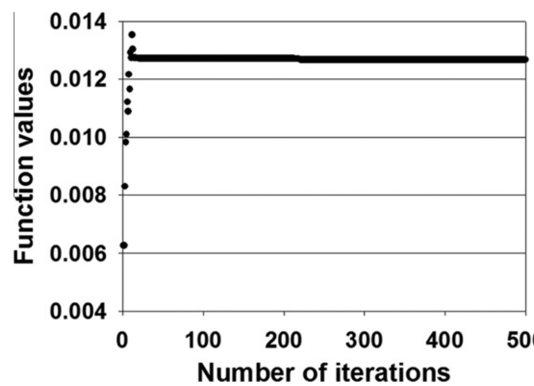

Fig. 11. Function values with respect to the number of iterations for tension/compression spring problem.

was in the infeasible region in the early iterations of WCA. After further iterations, the population was adjusted to the feasible region and the function values were reduced at each iteration.

The constraint violation values with respect to the number of iterations for the tension/compression spring problem are shown in Fig. 12. From Fig. 12, the obtained solutions did not satisfy the constraints in the early iterations. As the algorithm continued, the obtained results satisfied the constraints, while the value of constraint violation decreased.

3.5.5. Welded beam design problem

This design problem (see Appendix B.5), which has been often used as a benchmark problem, was proposed by Coello [48]. In this problem, a welded beam is designed for the minimum cost subject to constraints on shear stress (τ), bending stress (σ) in the beam, buckling load on the bar (P_b), end deflection of the beam (δ), and side constraints. There are four design variables as shown in Fig. 13: $h(x_1)$, $l(x_2)$, $t(x_3)$ and $b(x_4)$.

The optimization engines previously applied to this problem include GA2, GA3, CAEP, CPSO, HPSO, NM–PSO, hybrid genetic algorithm (HGA) [52], MGA [53], SC, and DE. The comparisons for the best solutions given by different algorithms are presented in Table 16. The comparisons of the statistical results are given in Table 17.

Among those previously reported studies, the best solution was obtained using NM–PSO with an objective function value of

Table 14
Comparison of the best solution obtained from various algorithms for the tension/compression spring problem.

DV	DEDS [30]	GA3 [49]	CPSO [29]	HEAA [32]	NM-PSO [41]	DELC [40]	WCA
X_1	0.051689	0.051989	0.051728	0.051689	0.051620	0.051689	0.051680
X_2	0.356717	0.363965	0.357644	0.356729	0.355498	0.356717	0.356522
X_3	11.288965	10.890522	11.244543	11.288293	11.333272	11.288965	11.300410
$g_1(X)$	1.45E–09	−1.26E–03	−8.25E–04	3.96E–10	1.01E–03	−3.40E–09	−1.65E–13
$g_2(X)$	−1.19E–09	−2.54E–05	−2.52E–05	−3.59E–10	9.94E–04	2.44E–09	−7.9E–14
$g_3(X)$	−4.053785	−4.061337	−4.051306	−4.053808	−4.061859	−4.053785	−4.053399
$g_4(X)$	−0.727728	−0.722697	−0.727085	−0.727720	−0.728588	−0.727728	−0.727864
$f(X)$	0.012665	0.012681	0.012674	0.012665	0.012630	0.012665	0.012665

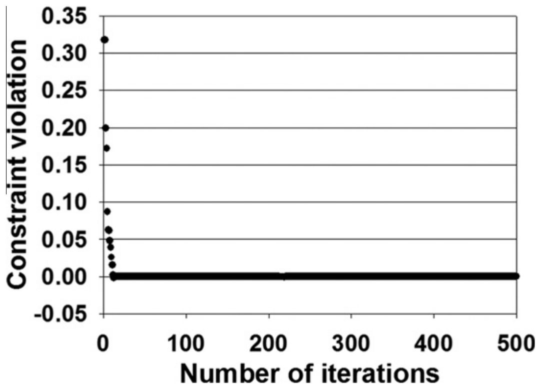

Fig. 12. Constraint violation values with respect to the number of iterations for tension/compression spring problem.

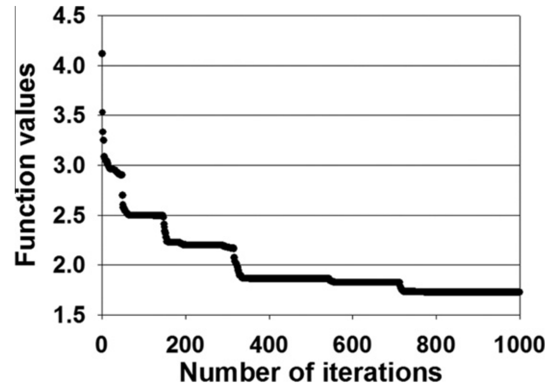

Fig. 14. Function values versus number of iterations for the welded beam problem.

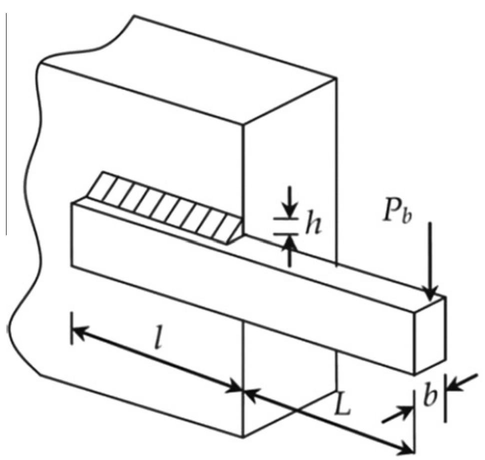

Fig. 13. Schematic view of welded beam problem.

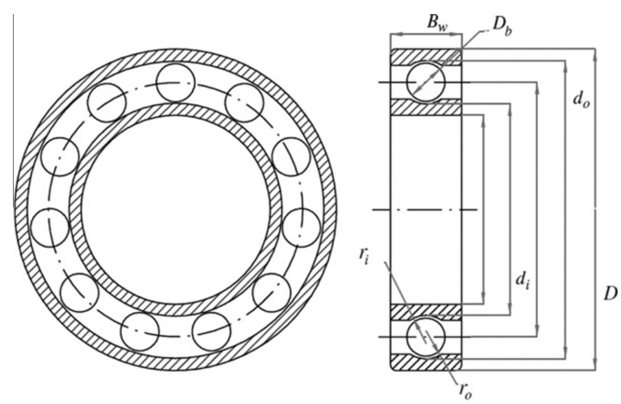

Fig. 15. Schematic view of rolling element bearing problem.

Table 17
Comparison of the statistical results obtained from different optimization engines for the welded beam problem.

Methods	Worst	Mean	Best	SD	NFEs
GA2 [48]	1.785835	1.771973	1.748309	1.12E−02	900,000
GA3 [49]	1.993408	1.792654	1.728226	7.47E−02	80,000
CAEP [37]	3.179709	1.971809	1.724852	4.43E−01	50,020
CPSO [29]	1.782143	1.748831	1.728024	1.29E−02	240,000
HPSO [38]	1.814295	1.749040	1.724852	4.01E−02	81,000
PSO–DE [25]	1.724852	1.724852	1.724852	6.7E−16	66,600
NM–PSO [41]	1.733393	1.726373	1.724717	3.50E−03	80,000
MGA [53]	1.995000	1.919000	1.824500	5.37E−02	NA
SC [43]	6.399678	3.002588	2.385434	9.6E−01	33,095
DE [27]	1.824105	1.768158	1.733461	2.21E−02	204,800
WCA	1.744697	1.726427	1.724856	4.29E−03	46,450
	1.801127	1.735940	1.724857	1.89E−02	30,000

$f(x)$ = 1.724717 after 80,000 function evaluations. Using the proposed WCA, the best solution of $f(x)$ = 1.724856 was obtained using 46,450 function evaluations. The proposed WCA was also performed for 30,000 function evaluations offering a best solution of $f(x)$ = 1.724857.

The optimization statistical results obtained by the WCA outperformed the obtained results by other considered algorithms, except the NM–PSO, PSO–DE, HPSO and CAEP in terms of cost value. However, WCA could offer a competitive set of statistical results in less number of function evaluations as shown in Table 17. Fig. 14 illustrates the function values with respect to the number of iterations for the welded beam design problem.

3.5.6. Rolling element bearing design problem

The objective of this problem (see Appendix B.6) is to maximize the dynamic load carrying capacity of a rolling element bearing, as demonstrated in Fig. 15. This problem has 10 decision variables

Table 16
Comparison of the best solution obtained from various algorithms for the welded beam problem.

DV	GA3 [49]	CPSO [29]	CAEP [37]	HGA [52]	NM–PSO [41]	WCA
$X_1(h)$	0.205986	0.202369	0.205700	0.205700	0.205830	0.205728
$X_2(l)$	3.471328	3.544214	3.470500	3.470500	3.468338	3.470522
$X_3(t)$	9.020224	9.048210	9.036600	9.036600	9.036624	9.036620
$X_4(b)$	0.206480	0.205723	0.205700	0.205700	0.20573	0.205729
$g_1(X)$	−0.103049	−13.655547	1.988676	1.988676	−0.02525	−0.034128
$g_2(X)$	−0.231747	−78.814077	4.481548	4.481548	−0.053122	−3.49E−05
$g_3(X)$	−5E−04	−3.35E−03	0.000000	0.000000	0.000100	−1.19E−06
$g_4(X)$	−3.430044	−3.424572	−3.433213	−3.433213	−3.433169	−3.432980
$g_5(X)$	−0.080986	−0.077369	−0.080700	−0.080700	−0.080830	−0.080728
$g_6(X)$	−0.235514	−0.235595	−0.235538	−0.235538	−0.235540	−0.235540
$g_7(X)$	−58.646888	−4.472858	2.603347	2.603347	−0.031555	−0.013503
$f(X)$	1.728226	1.728024	1.724852	1.724852	1.724717	1.724856

Table 18
Comparison of the best solution obtained using three algorithms for the rolling element bearing problem.

DV	GA4 [54]	TLBO [42]	WCA
X_1	125.717100	125.7191	125.721167
X_2	21.423000	21.42559	21.423300
X_3	11.000000	11.000000	11.001030
X_4	0.515000	0.515000	0.515000
X_5	0.515000	0.515000	0.515000
X_6	0.415900	0.424266	0.401514
X_7	0.651000	0.633948	0.659047
X_8	0.300043	0.300000	0.300032
X_9	0.022300	0.068858	0.040045
X_{10}	0.751000	0.799498	0.600000
$g(X_1)$	0.000821	0.000000	0.000040
$g(X_2)$	13.732999	13.15257	14.740597
$g(X_3)$	2.724000	1.525200	3.286749
$g(X_4)$	3.606000	0.719056	3.423300
$g(X_5)$	0.717000	16.49544	0.721167
$g(X_6)$	4.857899	0.000000	9.290112
$g(X_7)$	0.003050	0.000000	0.000087
$g(X_8)$	0.000007	2.559363	0.000000
$g(X_9)$	0.000007	0.000000	0.000000
$g(X_{10})$	0.000005	0	0.000000
$f(X)$	81843.30	81859.74	85538.48

Table 19
Comparison of statistical results using four optimizers for the rolling element bearing problem.

Methods	Worst	Mean	Best	SD	NFEs
GA4 [54]	NA	NA	81843.30	NA	225,000
ABC [36]	78897.81	81496.00	81859.74	0.69	10,000
TLBO [42]	80807.85	81438.98	81859.74	0.66	10,000
WCA	83942.71	83847.16	85538.48	488.30	3950

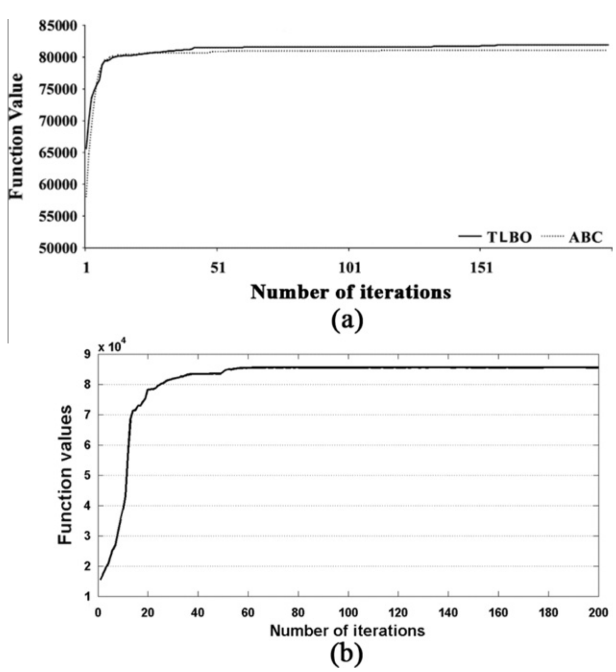

Fig. 16. Comparison of convergence rate for the rolling element bearing problem using: (a) TLBO and ABC, (b) WCA.

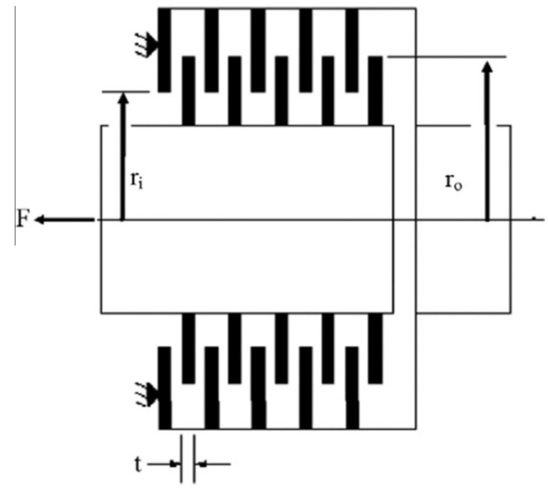

Fig. 17. Schematic view of multiple disc clutch brake design problem.

Table 20
Comparison of the best solution obtained using three algorithms for the multiple disc clutch brake problem.

DV	NSGA-II [56]	TLBO [42]	WCA
X_1	70.0000	70.000000	70.000000
X_2	90.0000	90.000000	90.000000
X_3	1.5.0000	1.000000	1.000000
X_4	1000.0000	810.000000	910.000000
X_5	3.0000	3.000000	3.000000
$g(X_1)$	0.0000	0.000000	0.000000
$g(X_2)$	22.0000	24.000000	24.000000
$g(X_3)$	0.9005	0.919427	0.909480
$g(X_4)$	9.7906	9830.371000	9.809429
$g(X_5)$	7.8947	7894.696500	7.894696
$g(X_6)$	3.3527	0.702013	2.231421
$g(X_7)$	60.6250	37706.250000	49.768749
$g(X_8)$	11.6473	14.297986	12.768578
$f(X)$	0.4704	0.313656	0.313656

which are pitch diameter (D_m), ball diameter (D_b), number of balls (Z), inner and outer raceway curvature coefficients (f_i and f_o), $K_{D_{\min}}$, $K_{D_{\max}}$, ε, e, and ζ, as shown in Fig. 15. The five latter appear in constraints and indirectly affect the internal geometry. Z is the discrete design variable and the remainder are continuous design variables. Constraints are imposed based on kinematic and manufacturing considerations.

The problem of the rolling element bearing was optimized previously using GA4 [54], ABC, and TLBO. Table 18 shows the comparison of the best solutions for the three optimizers and the proposed method in terms of design variables and function values, and constraints accuracy. The statistical optimization results for reported algorithms are given in Table 19.

From Table 19, the proposed method detected the best solution (the maximum dynamic load carrying capacity of a rolling element bearing) with considerable improvement compared with other optimizers. In terms of statistical results, WCA offered better results using less number of function evaluations (NFEs) compared with other considered algorithms. Fig. 16 compares the convergence rates.

From Fig. 16a, the convergence rates for ABC and TLBO are close (with a slightly higher mean searching capability for TLBO). However, WCA reached the best solution at 79 iterations (iteration number for obtained best solution = NFEs/N_s). Meanwhile, the WCA also found the best solution, as shown in Table 19.

3.5.7. Multiple disk clutch brake design problem

This minimization problem (see Appendix B.7), which is categorized as discrete optimization problem, is taken from [55]. Fig. 17 illustrates a schematic view of multiple disc clutch brake. The objective is to minimize the mass of the multiple disc clutch brake using five discrete design variables: inner radius (r_i = 60, 61, 62, ..., 80), outer radius (r_o = 90, 91, 92, ..., 110), thickness of the disc (t = 1, 1.5, 2, 2.5, 3), actuating force (F = 600, 610, 620, ..., 1000), and number of friction surfaces (Z = 2, 3, 4, 5, 6, 7, 8, 9).

The problem of the multiple disc clutch brake was also attempted by Deb and Srinivasan [56] using NSGA-II, TLBO, and ABC. The comparisons of best solutions and the statistical optimization results are given in Tables 20 and 21, respectively.

All the reported optimizers give the same optimal solution for the multiple disc clutch brake problem as shown in Table 21. However, in terms of statistical results including worst, mean, and standard deviation, the proposed WCA shows superiority compared with other optimization engines. Fig. 18 shows the comparison of convergence rate for three optimizers.

From Fig. 18a, it can be seen that the convergence rate for the TLBO method is faster than ABC in earlier generations. However, as the number of iterations increase, the convergence rate for both algorithms becomes nearly the same. As shown in Fig. 18b, the proposed method detected the best solution faster than other considered optimizers at 25 iterations (500 function evaluations), while TLBO and ABC methods found the best solution at over 45 iterations (more than 900 function evaluations).

These overall optimization results indicate that the WCA has the capability in handling various combinatorial optimization problems (COPs) and can offer optimum solutions (near or better than to the best-known results) under lower computational efforts (measured as the number of function evaluations). Therefore, it can be concluded that the WCA is an attractive alternative optimizer for constrained and engineering optimization challenging other metaheuristic methods especially in terms of computational efficiency

Table 21
Comparison of statistical results obtained using three optimizers for the multiple disc clutch brake problem.

Methods	Worst	Mean	Best	SD	NFEs
ABC [36]	0.352864	0.324751	0.313657	0.54	>900
TLBO [42]	0.392071	0.327166	0.313657	0.67	>900
WCA	0.313656	0.313656	0.313656	1.69E−16	500

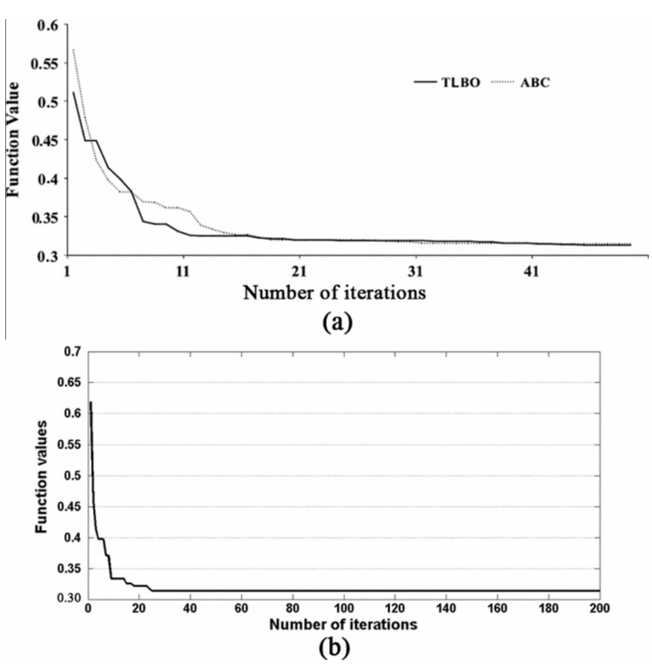

Fig. 18. Comparison of convergence rate for the multiple disc clutch brake problem using: (a) TLBO and ABC, (b) WCA.

4. Conclusions

This paper presented a new optimization technique called the water cycle algorithm (WCA). The fundamental concepts and ideas which underlie the method are inspired from nature and based on the water cycle process in real world. In this paper, the WCA with embedded constraint handling approach is proposed for solving eleven constrained benchmark optimization and engineering design problems. The statistical results, based on the comparisons of the efficiency of the proposed WCA against numerous other optimization methods, illustrate the attractiveness of the proposed method for handling numerous types of constraints. The obtained results show that the proposed algorithm, (generally), offers better solutions than other optimizers considered in this research in addition to its efficiency in terms of the number of function evaluations (computational cost) for almost every problem. In general, the WCA offers competitive solutions compared with other metaheuristic optimizers based on the reported and experimental results in this research. However, the computational efficiency and quality of solutions given by the WCA depends on the nature and complexity of the underlined problem. This applies to the efficiency and performance of numerous metaheuristic methods. The proposed method may be used for solving the real world optimization problems which require significant computational efforts efficiently with acceptable degree of accuracy for the solutions. However, further research is required to examine the efficiencies of the proposed WCA on large scale optimization problems.

Acknowledgments

The authors would like to acknowledge the Ministry of Higher Education of Malaysia and the University of Malaya, Kuala Lumpur, Malaysia for the financial support under grant number UM.TNC2/IPPP/UPGP/628/6/ER013/2011A.

Appendix A

A.1. Constrained problem 1

$$f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

subject to:

$$g_1(x) = 127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geqslant 0$$
$$g_2(x) = 282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geqslant 0$$
$$g_3(x) = 196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geqslant 0$$
$$g_4(x) = -4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \geqslant 0$$
$$-10 \leqslant x_i \leqslant 10 \quad i = 1,2,3,4,5,6,7$$

A.2. Constrained problem 2

$$f(x) = 5.3578547x_3^3 + 0.8356891x_1x_5 + 37.293239x_1 + 40729.141$$

subject to:

$$
\begin{aligned}
&g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 \\
&\qquad -0.0022053x_3x_5 - 92 \leqslant 0 \\
&g_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 \\
&\qquad -0.0022053x_3x_5 \leqslant 0 \\
&g_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 \\
&\qquad +0.0021813x_3^2 - 110 \leqslant 0 \\
&g_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 \\
&\qquad -0.0021813x_3^2 + 90 \leqslant 0 \\
&g_5(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 \\
&\qquad +0.0019085x_3x_4 - 25 \leqslant 0 \\
&g_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 \\
&\qquad -0.0019085x_3x_4 + 20 \leqslant 0 \\
&78 \leqslant x_1 \leqslant 102 \\
&33 \leqslant x_2 \leqslant 45 \\
&27 \leqslant x_i \leqslant 45 \quad i = 3,4,5
\end{aligned}
$$

A.3. Constrained problem 3

$$f(x) = -(\sqrt{n})^n.\prod_{i=1}^{n} x_i$$

subject to:

$$
\begin{aligned}
&h(x) = \sum_{i=1}^{n} x_i^2 = 1 \\
&0 \leqslant x_i \leqslant 1 \quad i = 1,\ldots,n
\end{aligned}
$$

A.4. Constrained problem 4

$$f(x) = -\frac{100 - (x_1-5)^2 - (x_2-5)^2 - (x_3-5)^2}{100}$$

subject to:

$$
\begin{aligned}
&g(X) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leqslant 0 \\
&0 \leqslant x_i \leqslant 10 \quad i = 1,2,3 \quad p,q,r = 1,2,3,\ldots,9
\end{aligned}
$$

Appendix B

B.1. Three-bar truss design problem

$$f(x) = (2\sqrt{2}x_1 + x_2) \times l$$

subject to:

$$
\begin{aligned}
&g_1(x) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leqslant 0 \\
&g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leqslant 0 \\
&g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leqslant 0 \\
&0 \leqslant x_i \leqslant 1 \quad i = 1,2 \\
&l = 100\ \text{cm}, P = 2\ \text{kN/cm}^2, \sigma = 2\ \text{kN/cm}^2
\end{aligned}
$$

B.2. Speed reducer design problem

$$
\begin{aligned}
f(x) = {}& 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) \\
& - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)
\end{aligned}
$$

subject to:

$$
\begin{aligned}
&g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leqslant 0 \\
&g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leqslant 0 \\
&g_3(x) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leqslant 0 \\
&g_4(x) = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leqslant 0 \\
&g_5(x) = \frac{\left[(745x_4/x_2x_3)^2 + 16.9\times10^6\right]^{1/2}}{110x_6^3} - 1 \leqslant 0 \\
&g_6(x) = \frac{\left[(745x_5/x_2x_3)^2 + 157.5\times10^6\right]^{1/2}}{85x_7^3} - 1 \leqslant 0 \\
&g_7(x) = \frac{x_2x_3}{40} - 1 \leqslant 0 \\
&g_8(x) = \frac{5x_2}{x_1} - 1 \leqslant 0 \\
&g_9(x) = \frac{x_1}{12x_2} - 1 \leqslant 0 \\
&g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leqslant 0 \\
&g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leqslant 0
\end{aligned}
$$

where

$$2.6 \leqslant x_1 \leqslant 3.6, 0.7 \leqslant x_2 \leqslant 0.8, 17 \leqslant x_3 \leqslant 28, 7.3 \leqslant x_4 \leqslant 8.3,$$
$$7.3 \leqslant x_5 \leqslant 8.3, 2.9 \leqslant x_6 \leqslant 3.9, 5.0 \leqslant x_7 \leqslant 5.5$$

B.3. Pressure vessel design problem

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

subject to:

$$
\begin{aligned}
&g_1(x) = -x_1 + 0.0193x \\
&g_2(x) = -x_2 + 0.00954x_3 \leqslant 0 \\
&g_3(x) = -\pi x_3^2x_4 - 4/3\pi x_3^3 + 1296,000 \leqslant 0 \\
&g_4(x) = x_4 - 240 \leqslant 0 \\
&0 \leqslant x_i \leqslant 100 \quad i = 1,2 \\
&10 \leqslant x_i \leqslant 200 \quad i = 3,4
\end{aligned}
$$

B.4. Tension/compression spring design problem

$$f(x) = (x_3 + 2)x_2x_1^2$$

subject to:

$$
\begin{aligned}
&g_1(x) = 1 - x_2^3x_3/71,785x_1^4x_1^4 \leqslant 0 \\
&g_2(x) = 4x_2^2 - x_1x_2/12,566(x_2x_1^3 - x_1^4) + 1/5108x_1^2 - 1 \leqslant 0 \\
&g_3(x) = 1 - 140.45x_1/x_2^2x_3 \leqslant 0 \\
&g_4(x) = x_2 + x_1/1.5 - 1 \leqslant 0 \\
&0.05 \leqslant x_1 \leqslant 2.00 \\
&0.25 \leqslant x_2 \leqslant 1.30 \\
&2.00 \leqslant x_3 \leqslant 15.00
\end{aligned}
$$

B.5. Welded beam design problem

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

subject to:

$$\begin{aligned}
&g_1(x) = \tau(x) - \tau_{\max} \leqslant 0\\
&g_2(x) = \sigma(x) - \sigma_{\max} \leqslant 0\\
&g_3(x) = x_1 - x_4 \leqslant 0\\
&g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leqslant 0\\
&g_5(x) = 0.125 - x_1 \leqslant 0\\
&g_6(x) = \delta(x) - \delta_{\max} \leqslant 0\\
&g_7(x) = P - P_c(x) \leqslant 0\\
&0.1 \leqslant x_i \leqslant 2 \quad i = 1, 4\\
&0.1 \leqslant x_i \leqslant 10 \quad i = 2, 3
\end{aligned}$$

where,

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \quad \tau' = \frac{P}{\sqrt{2}x_1x_2} \quad \tau'' = \frac{MR}{J}$$

$$M = P\left(L + \frac{x_2}{2}\right), \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma(x) = \frac{6PL}{x_4x_3^2}, \quad \delta(x) = \frac{4PL^3}{Ex_3^3x_4}, \quad P_c(x) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 6000lb, \quad L = 14in, \quad E = 30 \times 10^6 \text{ psi}, \quad G = 12 \times 10^6 \text{ psi}$$

$$\tau_{\max} = 13,600 \text{ psi}, \quad \sigma_{\max} = 30,000 \text{ psi}, \quad \delta_{\max} = 0.25in$$

B.6. Rolling element bearing design problem

$$\begin{aligned}
\max \quad & C_d = f_cZ^{2/3}D_b^{1.8} \quad \text{if } D \leqslant 25.4 \text{ mm}\\
& C_d = 3.647f_cZ^{2/3}D_b^{1.4} \quad \text{if } D \succ 25.4 \text{ mm}
\end{aligned}$$

subject to:

$$\begin{aligned}
&g_1(x) = \frac{\phi_0}{2\sin^{-1}(D_b/D_m)} - Z + 1 \geqslant 0\\
&g_2(x) = 2D_b - K_{D\min}(D - d) \geqslant 0\\
&g_3(x) = K_{D\max}(D - d) - 2D_b \geqslant 0\\
&g_4(x) = \zeta B_w - D_b \leqslant 0\\
&g_5(x) = D_m - 0.5(D + d) \geqslant 0\\
&g_6(x) = (0.5 + e)(D + d) - D_m \geqslant 0\\
&g_7(x) = 0.5(D - D_m - D_b) - \varepsilon D_b \geqslant 0\\
&g_8(x) = f_i \geqslant 0.515\\
&g_9(x) = f_o \geqslant 0.515
\end{aligned}$$

where,

$$f_c = 37.91\left[1 + \left\{1.04\left(\frac{1-\gamma}{1+\gamma}\right)^{1.72}\left(\frac{f_i(2f_o - 1)}{f_o(2f_i - 1)}\right)^{0.41}\right\}^{10/3}\right]^{-0.3} \times \left[\frac{\gamma^{0.3}(1-\gamma)^{1.39}}{(1+\gamma)^{1/3}}\right]\left[\frac{2f_i}{2f_i - 1}\right]^{0.41}$$

$$\phi_o = 2\pi - 2 \times \cos^{-1}\left(\frac{\left[\{(D-d)/2 - 3(T/4)\}^2 + \{D/2 - T/4 - D_b\}^2 - \{d/2 + T/4\}^2\right]}{2\{(D-d)/2 - 3(T/4)\}\{D/2 - T/4 - D_b\}}\right)$$

$$\gamma = \frac{D_b}{D_m}, \quad f_i = \frac{r_i}{D_b}, \quad f_0 = \frac{r_0}{D_b} \quad T = D - d - 2D_b$$

$$D = 160, \quad d = 90, \quad B_w = 30, \quad r_i = r_o = 11.033$$

$$0.5(D + d) \leqslant Dm \leqslant 0.6(D + d), 0.15(D - d) \leqslant Db \leqslant 0.45(D - d),$$

$$4 \leqslant Z \leqslant 50, 0.515 \leqslant f_i \text{ and } f_o \leqslant 0.6,$$

$$0.4 \leqslant K_{Dmin} \leqslant 0.5, 0.6 \leqslant K_{D_{\max}} \leqslant 0.7, 0.3 \leqslant \varepsilon \leqslant 0.4,$$

$$0.02 \leqslant e \leqslant 0.1, 0.6 \leqslant \zeta \leqslant 0.85.$$

B.7. Multiple disk clutch brake design problem

$$f(x) = \pi(r_o^2 - r_i^2)t(Z + 1)\rho$$

subject to:

$$\begin{aligned}
&g_1(x) = r_o - r_i - \Delta r \geqslant 0\\
&g_2(x) = l_{\max} - (Z + 1)(t + \delta) \geqslant 0\\
&g_3(x) = p_{\max} - p_{rz} \geqslant 0\\
&g_4(x) = p_{\max}v_{sr\max} - p_{rz}v_{sr} \geqslant 0\\
&g_5(x) = v_{sr\max} - v_{sr} \geqslant 0\\
&g_6(x) = T_{\max} - T \geqslant 0\\
&g_7(x) = M_h - sM_s \geqslant 0\\
&g_8(x) = T \geqslant 0
\end{aligned}$$

where,

$$M_h = \frac{2}{3}\mu FZ\frac{r_o^3 - r_i^3}{r_0^2 - r_i^2}, \quad p_{rz} = \frac{F}{\pi(r_0^2 - r_i^2)},$$

$$v_{rz} = \frac{2\pi n(r_o^3 - r_i^3)}{90(r_0^2 - r_i^2)}, \quad T = \frac{I_z\pi n}{30(M_h + M_f)}$$

$\Delta r = 20$ mm, I_z = 55 kgmm^2, $p_{\max}$ = 1 MPa, $F_{\max}$ = 1000 N, $T_{\max}$ = 15 s, μ = 0.5, s = 1.5, M_s = 40 Nm, M_f = 3 Nm, n = 250 rpm, v_{srmax} = 10 m/s, $l_{\max}$ = 30 mm, r_{imin} = 60, r_{imax} = 80, r_{omin} = 90, $r_{o\text{-max}}$ = 110, $t_{\min}$ = 1.5, $t_{\max}$ = 3, $F_{\min}$ = 600, $F_{\max}$ = 1000, $Z_{\min}$ = 2, $Z_{\max}$ = 9.

References

[1] Lee KS, Geem ZW. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Meth Appl Mech Eng 2005;194:3902–33.
[2] Holland J. Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press; 1975.
[3] Goldberg D. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley; 1989.
[4] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. Perth, Australia: 1995. p. 1942–8.
[5] Kirkpatrick S, Gelatt C, Vecchi M. Optimization by simulated annealing. Science 1983;220:671–80.
[6] Coello CAC. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Meth Appl Mech Eng 2002;191:1245–87.
[7] Areibi S, Moussa M, Abdullah H. A comparison of genetic/memetic algorithms and other heuristic search techniques. Las VeGAs, Nevada: ICAI; 2001.
[8] Elbeltagi E, Hegazy T, Grierson D. Comparison among five evolutionary-based optimization algorithms. Adv Eng Inf 2005;19:43–53.
[9] Youssef H, Sait SM, Adiche H. Evolutionary algorithms, simulated annealing and tabu search: a comparative study. Eng Appl Artif Intell 2001;14:167–81.
[10] Giraud-Moreau L, Lafon P. Comparison of evolutionary algorithms for mechanical design components. Eng Optim 2002;34:307–22.
[11] Chootinan P, Chen A. Constraint handling in genetic algorithms using a gradient-based repair method. Comput Oper Res 2006;33:2263–81.
[12] Trelea IC. The particle swarm optimization algorithm: convergence analysis and parameter selection. Inform Process Lett 2003;85:317–25.
[13] He Q, Wang L. An effective co-evolutionary particle swarm optimization for engineering optimization problems. Eng Appl Artif Intell 2006;20:89–99.
[14] Gomes HM. Truss optimization with dynamic constraints using a particle swarm algorithm. Expert Syst Appl 2011;38:957–68.

[15] David S. The water cycle, illustrations by John Yates. New York: Thomson Learning; 1993.
[16] Strahler AN. Dynamic basis of geomorphology. Geol Soc Am Bull 1952;63:923–38.
[17] Montes EM, Coello CAC. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 2008;37:443–73.
[18] Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables. J Const Steel Res 2009;65:1558–68.
[19] Koziel S, Michalewicz Z. Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. IEEE Trans Evol Comput 1999;7:19–44.
[20] Ben Hamida S, Schoenauer M. ASCHEA: new results using adaptive segregational constraint handling. IEEE Trans Evol Comput 2002:884–9.
[21] Tang KZ, Sun TK, Yang JY. An improved genetic algorithm based on a novel selection strategy for nonlinear programming problems. Comput Chem Eng 2011;35:615–21.
[22] Michalewicz Z. Genetic algorithms, numerical optimization, and constraints. In: Esheman L, editor. Proceedings of the sixth international conference on genetic algorithms. San Mateo: Morgan Kauffman; 1995. p. 151–8.
[23] Mezura-Montes E, Coello CAC. A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Trans Evol Comput 2005;9:1–17.
[24] Tessema B, Yen GG. A self adaptive penalty function based algorithm for constrained optimization. IEEE Trans Evol Comput 2006:246–53.
[25] Liu H, Cai Z, Wang Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 2010;10:629–40.
[26] Runarsson TP, Xin Y. Stochastic ranking for constrained evolutionary optimization. IEEE Trans Evol Comput 2000;4:284–94.
[27] Lampinen J. A constraint handling approach for the differential evolution algorithm. IEEE Trans Evol Comput 2002:1468–73.
[28] Becerra R, Coello CAC. Cultured differential evolution for constrained optimization. Comput Meth Appl Mech Eng 2006;195:4303–22.
[29] Renato AK, Leandro Dos Santos C. Coevolutionary particle swarm optimization using gaussian distribution for solving constrained optimization problems. IEEE Trans Syst Man Cybern Part B Cybern 2006;36:1407–16.
[30] Zhang M, Luo W, Wang X. Differential evolution with dynamic stochastic selection for constrained optimization. Inform Sci 2008;178:3043–74.
[31] Runarsson TP, Xin Y. Search biases in constrained evolutionary optimization. IEEE Trans Syst Man Cybern Part C Appl Rev 2005;35:233–43.
[32] Wang Y, Cai Z, Zhou Y, Fan Z. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint handling technique. Struct Multidisc Optim 2009;37:395–413.
[33] Takahama T, Sakai S. Constrained optimization by applying the α; constrained method to the nonlinear simplex method with mutations. IEEE Trans Evol Comput 2005;9:437–51.
[34] A.E.M. Zavala, A.H. Aguirre, E.R.V. Diharce, Constrained optimization via evolutionary swarm optimization algorithm (PESO). In: Proceedings of the 2005 conference on genetic and evolutionary computation. New York, USA: 2005. p. 209–16.
[35] Huang FZ, Wang L, He Q. An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 2007;186(1):340–56.
[36] Karaboga D, Basturk B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. LNAI 2007;4529:789–98 [Berlin: Springer-Verlag].
[37] Coello CAC, Becerra RL. Efficient evolutionary optimization through the use of a cultural algorithm. Eng Optim 2004;36:219–36.
[38] He Q, Wang L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 2007;186:1407–22.
[39] Amirjanov A. The development of a changing range genetic algorithm. Comput Meth Appl Mech Eng 2006;195:2495–508.
[40] Wang L, Li LP. An effective differential evolution with level comparison for constrained engineering design. Struct Multidisc Optim 2010;41:947–63.
[41] Zahara E, Kao YT. Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems. Expert Syst Appl 2009;36:3880–6.
[42] Rao RV, Savsani VJ, Vakharia DP. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 2011;43:303–15.
[43] Ray T, Liew KM. Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 2003;7:386–96.
[44] Mezura-Montes E, Coello CAC. Useful infeasible solutions in engineering optimization with evolutionary algorithms, MICAI 2005. Lect Notes Artif Int 2005;3789(2005):652–62.
[45] Montes E, Reyes JV, Coello CAC. Modified differential evolution for constrained optimization. In: IEEE congress on evolutionary computation. CEC; 2006a. p. 25–32.
[46] Montes E, Coello CAC, Reyes JV. Increasing successful offspring and diversity in differential evolution for engineering design. In: Proceedings of the seventh international conference on adaptive computing in design and manufacture. 2006b. p. 131–9.
[47] Kannan BK, Kramer SN. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 1994;116:405–11.
[48] Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 2000;41:113–27.
[49] C Coello CA, Mezura Montes E. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inf 2002;16:193–203.
[50] Coelho LDS. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 2010;37:1676–83.
[51] Arora JS. Introduction to optimum design. New York: McGraw-Hill; 1989.
[52] Yuan Q, Qian F. A hybrid genetic algorithm for twice continuously differentiable NLP problems. Comput Chem Eng 2010;34:36–41.
[53] Coello CAC. Constraint-handling using an evolutionary multiobjective optimization technique. Civ Eng Environ Syst 2000;17:319–46.
[54] Gupta S, Tiwari R, Shivashankar BN. Multi-objective design optimization of rolling bearings using genetic algorithm. Mech Mach Theory 2007;42:1418–43.
[55] Osyczka A. Evolutionary algorithms for single and multicriteria design optimization: studies in fuzzyness and soft computing. Heidelberg: Physica-Verlag; 2002.
[56] Deb K, Srinivasan A. Innovization: innovative design principles through optimization, Kanpur genetic algorithms laboratory (KanGAL). Indian Institute of Technology Kanpur, KanGAL report number: 2005007; 2005.