Evolutionary Intelligence
https://doi.org/10.1007/s12065-021-00688-6

RESEARCH PAPER

Solving unconstrained, constrained optimization and constrained engineering problems using reconfigured water cycle algorithm

Heba F. Eid[1] · **Ajith Abraham**[2,3]

Received: 29 April 2021 / Revised: 11 November 2021 / Accepted: 18 November 2021

Abstract
Water cycle algorithm (WCA) is a recent meta-heuristic algorithm presented to solve various optimization problems. WCA has received critical intrigued from researchers in different fields. Nevertheless, the search equation provided in WCA is not of adequate explorative behavior. In this study a reconfigured version of the WCA is proposed, the proposed algorithm is named as RWCA. So as to improve the exploration procedure, a new position update strategy is proposed by integrating cauchy operator and a greedy selection procedure. Furthermore, in order to promote the exploration-exploitation balance of the RWCA algorithm, a nonlinear controlling parameter is proposed. The performance of RWCA is exhibited on 19 unconstrained benchmark functions. Statistical analysis proves that, RWCA significantly improves the performance of basic WCA by providing better solution quality, faster convergence and stronger robustness. Moreover, RWCA has been used to solve five constrained numerical and three engineering application problems. Based on the experiments and comparative findings, RWCA illustrates the adequacy and effectiveness to solve various constrained problems; as well as its capability of providing promising and competitive results solving real-world challenging engineering problems.

Keywords Water cycle algorithm · Constrained optimization · Constrained engineering problem

1 Introduction

Global Optimization methods play a vital role in many real-world applications such as operational research, information science, economics management, data reduction and engineering design [17, 18, 31, 36, 66, 67].

In real world most optimization (design) problems are highly nonlinear and containing many variables under different complex constraints. Classical optimization algorithms require enormous computational efforts, which tend to fail as the problem search space increases. This motivates for employing meta-heuristics algorithms which show reliable, robust performance and higher computational efficiency in avoiding local minima [8, 9, 19, 23, 57, 77]. Due to these advantages, meta-heuristics methods have been successfully applied to solve a difference of constrained optimization problems. The most prominent meta-heuristics algorithms proposed in literature are particle swarms [16], ants [15], bees [58], fishes [47] and bats [75].

A general constrained problem is an optimization problem for which an objective function $f(\mathbf{x})$ is to be minimize or maximize subject to nonlinear or linear equality constraints $G_j(\mathbf{x}) = 0$, inequality constraints $G_j(\mathbf{x}) \geq 0$ and the design variables $\mathbf{x} = \left(x_1, x_2, \dots, x_n\right)$; $l_i \leq x_i \leq u_i$; where l_i and u_i are the lower bound and the upper bound of x_i respectively.

Wang et al. presented an adaptive trade-off model (ATM) for solving different constrained problems. The ATM model obtains a trade-off scheme between objective function and constraint violations during the different phases of a search process [72].

Tuba and Bacanin developed an enhanced seeker optimization algorithm (SOA) hybridized with firefly algorithm (FA) for solving constrained optimization problems. In this model, the FA was used to enhance the model exploitation capabilities, and hybridized with SOA to control exploitation-exploration balance [69].

✉ Heba F. Eid
eid.heba@gmail.com; heba.fathy@azhar.edu.eg

[1] Faculty of Science, Al-Azhar University, Cairo, Egypt

[2] Machine Intelligence Research Labs, MIR Labs, Auburn, 98071 Washington, USA

[3] Center for Artificial Intelligence, Innopolis University, Innopolis, Russia

Published online: 06 January 2022

Mirjalili et al. presented a new meta-heuristic named Grey Wolf Optimizer (GWO) motivated by the leadership and social behavior of grey wolves. GWO is benchmarked on 29 different test functions and substantiated its sufficient explorative capability as compared to other well-known meta-heuristic algorithms. In addition, GWO algorithm was considered to solve 3 classical engineering design problems; welded beam, tension-compression spring and pressure vessel [54].

Wen et al. developed a constrained optimization algorithm which combines the improved grey wolf optimization (IGWO) with the modified augmented Lagrangian (MAL) multiplier, the presented algorithm named MAL-IGWO. For which, the MAL method converts a constrained problem into an unconstrained one and the IGWO is applied to find the global optimum the unconstrained problem [43].

Authors in [59] presented a nature-inspired optimization algorithm, called polar bear optimization algorithm (PBO); which mimics the way polar bear hunt to survive in harsh arctic conditions. The PBO simulated a global and local search with an efficient model of the dynamic mechanism of births and deaths of individuals in the population. The PBO algorithm was subjected to 13 test functions and for 4 real design engineering problems. The experimental research and comparisons have shown high potential of the PBO for various applications.

Cuevas et al. proposed an approach which combines the explorative characteristics of the invasive weed optimization algorithm (IWO), the dispersion capacities of a mixed Gaussian–Cauchy distribution, and the probabilistic models of the estimation distribution algorithms (EDAs) to create its own search method. The proposed approach, have been used to solve five engineering problems [11].

Polap and Wozniak proposed a mathematical model based on red fox habits. The model was developed for optimization purposes and named red fox optimization algorithm (RFO). RFO algorithm, simulated a global search model that mimicked how red foxes look for prey over land, as well as a local search model that mimicked how red foxes disguise prey while hunting. The RFO algorithm was tested on 22 benchmark test functions and 7 classic engineering optimization problems. Experimental and comparative results to other meta-heuristic algorithms have shown that the RFO algorithm can precisely find the optima solution of the test functions [60].

Hayyolalam and Kazem propose a meta-heuristic algorithm inspired by the unique mating behavior of black widow spiders (BWO); which is inspired by the special mating and reproducing the new generation behavior of black widow spiders. The BWO algorithm was investigated over 51 benchmark functions and also 3 real engineering design problems in order to illustrate its performance efficiency. The experimental comparison of BWO with other well-known optimization algorithm shows the BWO high performance in finding the real global optima with fast convergence and a high level of accuracy [27].

Water Cycle Algorithm (WCA) is a meta-heuristic population-based algorithm which mimics the water cycle in the nature; how streams and rivers flow downhill towards the sea; to perform optimization. It was proposed by Eskandar et al. for solving constrained optimization problems and engineering design problems [20].

WCA has been progressively applied to various research areas in the literature due to its efficiency of solving complex optimization problems. For instance, solving multi-objective optimization problems [63], detecting optimum reactive power dispatch problems [37], antenna array pattern synthesis [26], sizing optimization of space trusses [64], tune the gains of PID controller [55], distribution network reconfiguration [56] and spam e-mail detection [2].

Nevertheless, population-based algorithms can provide promising solution for the optimization problems, as the extension of the search space dimension, they faces up to some challenging issues. One of the main issue is that, they often get trapped in local optima (LO) when solving complex multi-modal problems [3, 35]. Furthermore, the convergence speed of the population based algorithm can be typically slow. For population-based algorithms; exploration represents the diversification of the search space, while exploitation considers the intensification of the best solution. Consequently, in order to obtain good performance on complex optimization problems, they should be well balanced [1, 13, 25, 29].

In this paper, a reconfigured water cycle algorithm (RWCA) is proposed. For which two effective strategies composed of Cauchy distributions (CD) and nonlinear control parameter are synchronously embedded into the basic WCA to guide the agents throughout the search space. The lingering tails of the CD maintain a higher probability of making long jumps, which helps in preventing the trapping in local optimums and premature convergence [76]. The sea in the RWCA is allowed to update its position using cauchy distributed random number. Consequently under this distribution, the exploration ability of the proposed RWCA is promoted. Moreover, to adjust the balance between the exploitation and exploration capabilities of the RWCA; a nonlinear control strategy is proposed.

In order to assess the coherence and robustness of the RWCA, the performance of the proposed RWCA is investigated on 19 unconstrained benchmark functions. As well as, 5 constrained numerical and 3 engineering design problems are solved using the proposed RWCA. The results obtained by the RWCA of all test problems were compared with various meta-heuristic algorithms reported in the literature.

The major contributions of this paper are as follows:

1. A reconfigured framework for the WCA algorithm is proposed (RWCA), which combining the advantages of the WCA, Cauchy distribution position update strategy and nonlinear control parameter.
2. Several tests are conducted over unconstrained unimodal and multimodal benchmark functions that are adopted for assessing the effectiveness of the proposed RWCA algorithm.
3. Developing a constrained handling RWCA algorithm, which is able to escape from local optima problem and promote the harmony between exploration and exploitation.
4. The constrained handling RWCA algorithm is adapted for solving various constrained numerical and real world engineering problems.The experimental results ensure that the constrained handling RWCA algorithm is efficient enough in solving complex constrained real world optimization problems.

The rest of the paper is organized as follows: In Sect. 2, a brief description of Water cycle algorithm (WCA) and Cauchy distributions is presented. In Sect. 3, the proposed RWCA is elaborated. In Sect. 4, the efficiency of the proposed RWCA algorithm investigated on unconstrained functions, constrained numerical and 3 well-known engineering design problems, in addition the comparative study of RWCA against various state-of-the-art optimization algorithms are presented. Finally, in Sect. 5 the main findings of this study are discussed.

2 Preliminary descriptions concepts

The proposed RWCA algorithm integrates the optimization characteristics of the WCA algorithm and the dispersion capacities of CD to creates its own search strategy. In the following section, the brief description of the WCA algorithm and the CD are presented.

2.1 Water cycle algorithm

WCA is a meta-heuristic algorithm which is derived by the observation of the water cycle process in nature and proposed by Eskandar et al. [20]. WCA simulates the flow of streams and rivers, rainfall, confluence, and evaporation.

For which, an initial population of variables is randomly generated by the rainfall process. Then, the initial population is divided in terms of having the least cost into three grades; sea (best solution), river (near to the current best) and stream.

$$\textit{Total population} = \begin{bmatrix} \text{Sea} \\ \text{River}_{1} \\ \vdots \\ \text{Stream}_{N_{sr+1}} \\ \vdots \\ \text{Stream}_{N_{pop}} \end{bmatrix} \tag{1}$$

where N and N_{pop} are the number of design variables (problem dimension) and the total number of population respectively.

$$N_{sr} = \textit{Number Of Rivers} + 1 \tag{2}$$

$$N_{streams} = N_{pop} - N_{sr} \tag{3}$$

N_{sr} represents the total number of sea and rivers; and $N_{streams}$ indicates the number of streams which indirectly or directly flow to sea and rivers.

The cost of a raindrop is attain by the evaluation of the cost function

$$\text{cost}_i = f\left(x_1^i, x_2^i, \ldots, x_{N_{rs}}^i\right) \quad i = 1, 2, 3, \ldots, N_{pop} \tag{4}$$

In order to simulate the flow of the streams to the rivers, and the streams and the rivers to the sea in nature, WCA uses the following position updating equation:

$$X_{\text{Stream}}^{i+1} = X_{\text{Stream}}^{i} + \text{rand} \times C \times \left(X_{\text{River}}^{i} - X_{\text{Stream}}^{i}\right) \tag{5}$$

$$X_{\text{Stream}}^{i+1} = X_{\text{Stream}}^{i} + \text{rand} \times C \times \left(X_{\text{Sea}}^{i} - X_{\text{Stream}}^{i}\right) \tag{6}$$

$$X_{\text{River}}^{i+1} = X_{\text{River}}^{i} + \text{rand} \times C \times \left(X_{\text{Sea}}^{i} - X_{\text{Rirer}}^{i}\right) \tag{7}$$

where rand is a uniformly distributed random number within the range of [0, 1] and C is a constant value between 1 and 2.

One of the most important characteristics of the meta-heuristic algorithms is randomization. In WCA, to increase randomization, the raining and evaporation process are considered.

Raining and evaporation take place when the distance between a river or any stream and the sea is less than parameter d_{max}

$$\left|X_{\text{Sea}}^{i} - X_{\text{River}}^{i}\right| < d_{max} \quad i = 1, 2, 3, \ldots, N_{sr} - 1 \tag{8}$$

A large value for d_{max} reduces the search and leads to focus more on exploration, while for exploitation a small d_{max} value motivate the search intensity near the sea. Therefore, to make a proper trade-off between exploitation and exploration in the WCA, the value of d_{max} adaptively decreases linearly using the following equation:

$$d_{\max}^{i+1} = d_{\max}^{i} - \frac{d_{\max}^{i}}{\max\ \text{iteration}} \tag{9}$$

After fulfilling the evaporation condition, the raining procedure is performed. In the raining procedure, the new solutions (scattered streams) are generate by the formula:

$$X_{\text{Stream}}^{\text{new}} = LB + \text{ rand } \times (UB - LB) \tag{10}$$

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

2.2 Cauchy distribution

Cauchy distribution (CD) is a continuous probability distribution, with parameters s and t; where , s is a scaling parameter and t is a real positive number named as the location parameter.

The probability density function of the standard Cauchy distribution is given as follows

$$f(x) = \frac{1}{s\pi\left(1 + ((x-t)/s)^2\right)} \tag{11}$$

For the standard cauchy distribution $t = 0$ and$s = 1$; and the cumulative distribution function is given by:

$$F(x) = \frac{1}{\pi}\arctan(x) + 0.5 \tag{12}$$

Therefore if

$$y = \frac{1}{\pi}\arctan(x) + 0.5 \tag{13}$$

A cauchy random variable in the range [0, 1) can be generated by inverting Eq. 13

$$x = \tan(\pi(y - 0.5)) \tag{14}$$

3 Reconfigured water cycle algorithm (RWCA)

3.1 Non-linear control parameter strategy

From the original literature of WCA, parameter C is a predetermined constant (set to 2) that affects the balance among the exploitative and explorative tendencies. For which, a C value greater than ”one” enables the streams to move toward the rivers and sea from disparate directions.

In the proposed RWCA a nonlinear decreasing control parameter strategy is introduce. Whereby, the value of C is decreased nonlinearly over the course of iteration by using the following equation:

$$C = 1 + \cos\left(\frac{t}{t_{max}}\right) \tag{15}$$

For which, t is the current iteration and t_{max} is the total number of iteration. The reasons for proposing such nonlinear increasing strategy is that, at the beginning of the iterations the population has a higher diversity, thus a larger value of C encourages global exploration. On the contrary, the streams are captivate to the global optima (sea) at the latter iterations, thus a smaller value of C encourages local exploitation.

3.2 Modified position update strategy

One of the main operators of the WCA is position updating process. In the updating process, the population individuals (rivers and streams) are attracted towards the sea as the guide solution for other solutions. Therefore, the algorithm may converge prematurely without sufficient search space exploration, which leads to local optima stagnation issues.

To reduce the potential of premature convergence and local optima stagnation of the proposed RWCA, cauchy operator strategy was used to the search agents positions update process.Since the expectation of CD is not defined (has no mean), so the variance of a CD does not exist (the variance is infinite). Accordingly, cauchy operator produce a long jump, which helps in escaping from trapping in local optima.

In the proposed RWCA, the Cauchy operator is used to generate a new agent relative to the current best agent when the i-th agent x(t + 1) position is updated. From Eq. 14 since y has a uniform distribution in the range (0,1]. Thus, we obtain the following equation:

$$x = \tan(\pi(rand - 0.5)) \tag{16}$$

Under a cauchy perturbation, Eq. 16, a new best individual (sea) is produced considering the following equation:

$$X_c(t+1) = X_{sea}(t) + X_{sea}(t) \cdot Cauchy(dim) \tag{17}$$

where (.) is the entry-wise multiplications and *dim* is the number of design variables.

However, according to the position update Eq. 17 the probability that the new best agent is a good agent is the same as the probability that the agent is a bad one. Thus, in order to guarantee the excellence of the best agent (sea), RWCA uses a selection criterion to determine which solution has a better fitness value (minimum cost function). The criterion for selection is described as follows:

$$X_{sea}^{t+1} = \begin{cases} X_c^{t+1} & , \quad f\left(X_c^{t+1}\right) < f(\text{X}_{\text{sea}}{}^{t+1}) \\ X_{sea}^{t+1}, & \quad otherwise \end{cases} \tag{18}$$

According to Eq. 18, the agent with fitter value in each iteration is chosen as the sea for the next iterations and the other one is discarded.

3.3 Constraints handling strategy

When solving constrain problems a main challenge is how the algorithm handles constraints relating to the given problem. The proposed RWCA attempted to find design variables solutions that lie within the LB and UB bounds, while handling constraints (inequality, equality, linear and nonlinear) based on the concept of feasibility-based criterion [12]. At each iteration the solutions; sea and Cauchy-sea; are compared using the following: (1) The feasible solution is preferred to the infeasible solution; (2) If both solutions are feasible, the one with better function cost is selected; (3) If both solutions are infeasible, the solution with the smaller constraint violation degree is preferred. Using the above criterion, the proposed RWCA search is oriented to the feasible region with the better global minimum cost.

Aiming to enhance the exploration near the optimum solution in the feasible region for constrained problems, RWCA adapted the following equation to motivate the generation of new streams.

$$X^{new}_{stream} = X_{sea} + \sqrt{\mu} \times \mathrm{rand}n(1, dim) \tag{19}$$

where μ is a coefficient representing the domain of searching near the best solution (sea). Large value for μ increases the potentiality to exit from feasible domain, while smaller value leads the algorithm to search in smaller domain around the sea. In order to determine proper values for μ, a trial and error experiments has been conducted. According to the experiments, the most suitable values of μ to solve constrained optimization problems has recognized as 0.4. The procedure for the proposed RWCA is shown in Fig. 1.

3.4 Proposed reconfigured water cycle algorithm

According to the above description, the exploration and exploitation modification in the RWCA are hybridized and the search process is being performed until the max number of iterations is reaches or when the best solution is found. The pseudo code of the proposed RWCA algorithm is presented in Algorithm 1.

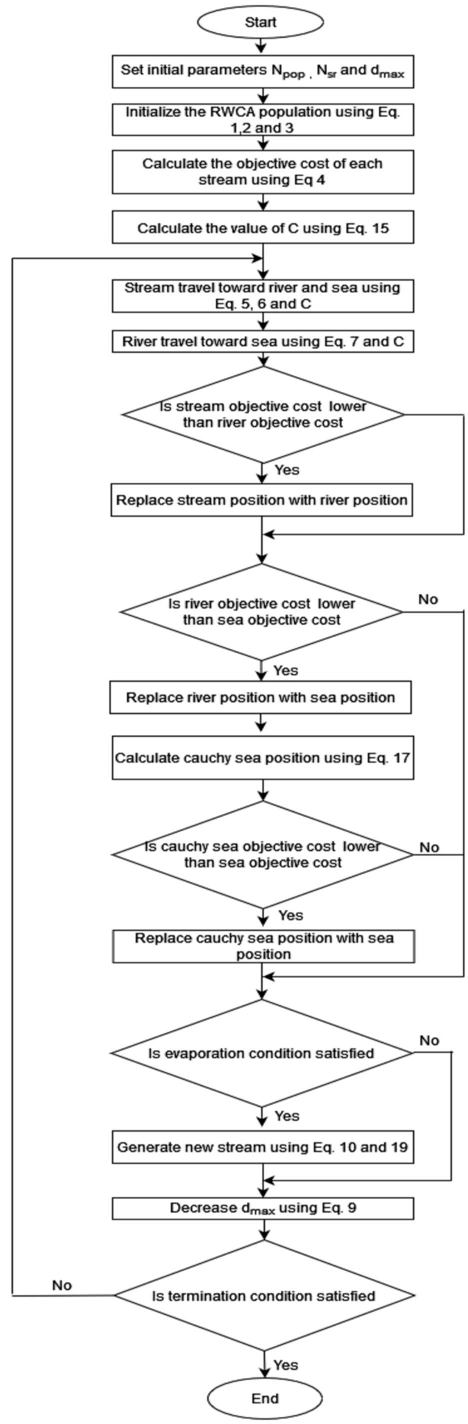

Fig. 1 Flowchart of the proposed RWCA

4 Experimental results and analysis

Algorithm 1 Pseudocode of RWCA Algorithm

Input:
Population total number N_{pop}
total number of sea and rivers N_{sr}
Number of optimization iterations Max_Iter and d_{max}
Output:
Optimal RWCA position X_{sea}

```
#Initialize the RWCA population positions randomly.
for i=1:$N_{pop}$ do
    Create streams $X_{stream}$
    Calculate objective cost $f(X_{stream})$
end for
sort $X_{stream}$ from $f_{best}$ to $f_{worst}$
Sea ← $the first stream$
Rivers ←$N_{sr}$ - 1
Stream ←$N_{pop}$ - $N_{sr}$
while $t \le Max\_Iter$ do
    Calculate the value of $C$ using eq. 15
    for i=1:$N_{pop}$ do
        Update the position of $X_{stream}$ using eq. 5, 6 and C
        Calculate generated stream objective cost $f(X_{stream})$
        if $f(X_{stream}) < f(X_{river})$ then
            River.position= new stream.position
            if $f(X_{stream}) < f(X_{sea})$ then
                Update the position of $X_{stream}$ using eq. 17
                if $f(X_{c.stream}) < f(X_{sea})$ then
                    Sea.position= cauchy Stream.position
                else
                    Sea.position= new stream.position
                end if
            end if
        end if
        if $f(X_{river}) < f(X_{sea})$ then
            Update the position of $X_{river}$ using eq. 17
            if $f(X_{c.river}) < f(X_{sea})$ then
                Sea.position= cauchy River.position
            else
                Sea.position= River.position
            end if
        end if
    end for
    for i=1:$N_{sr} - 1$ do
        Update the position of $X_{river}$ using eq. 7 and C
        Calculate generated river objective cost $f(X_{river})$
        if $f(X_{river}) < f(X_{sea})$ then
            Sea.position= River.position
        end if
    end for
    for i=1:$N_{sr} - 1$ do
        if $|River - Sea| < d_{max} or rand < 0.1$ then
            Generate new $X_{stream}$ using eq. 10
        end if
    end for
    # for solving unconstrained problems
    for i=1: no. of streams do
        if $|Stream - Sea| < d_{max}$ then
            Generate new $X_{stream}$ using eq. 19
        end if
    end for
    Decrease $d_{max}$ using eq. 9
    t=t+1
end while
return $X_{sea}$
```

With the aim of investigating the capabilities of the proposed RWCA algorithms; Matlab R2015b was used for implementation purposes. All of the experiments were carried out on Intel(R), Core i7- 4910MQ CPU @ 2.90 GHz and 16 GB RAM.

4.1 Unconstrained optimization problems statistical analysis

By virtue of stochastic characteristic of the meta-heuristic algorithms, the performance of a specific algorithm may vary on a certain test problem within different runs.

In consequence, in order to assess the performance of the RWCA, several test problems solved within various runs to conduct an accurate conclusion. In this paper, the proposed RWCA is evaluated on 19 well-regarded benchmark functions extracted from the CEC2015 competition [39].

According to their characteristics, the benchmark functions are divided into two distinct groups : unimodal (*f*1–*f*7) and multimodal functions (*f*8–*f*19). Whereby, The benchmark objective functions could be characterized as continuous, discontinuous, scalable, non-scalable, differentiable, non-differentiable, separable and non-separable. Moreover, the test functions have different dimensionality, through which the search space increases exponentially as the function dimension increases. This dimensionality produces a critical barrier for optimization algorithms solving highly nonlinear problems [74].

The unimodal functions have a single global optimum and no local optima. Thus, these functions are considered to evaluate the exploitative tendencies of meta-heuristic algorithms. Conversely, multimodal and fixed-dimension multimodal functions (*f*14–*f*19) are helpful to estimate the exploration ability and local optima escaping of the meta-heuristic algorithms. These functions have a unique global optima as well as several number of local optima.

The internal parameters of the WCA and RWCA are chosen as: population size (Npop) was 30 and the total number of iterations (max_it) was 500, Nsr = 4 and dmax = 1.0E–16 on all of the simulations. To make an unbiased comparison, the results are obtained over 30 independent runs on each test function; with entirely random initial conditions.

For comparing the WCA and RWCA algorithms four distinct performance criteria are considered: the minimum solution (best), the average best solution (mean), the maximum solution (worst) and the standard deviation (St.dev). The best, mean and worst indicators assess the solution accuracy, while the St.dev evaluates the robustness of the obtained solution.

The experimental results of the unimodal test functions are reported in Table 1. From the results in Table 1, RWCA outperformed the standard WCA algorithm in term of best, mean and worst results for *f*1–*f*5 functions, and provided better results for *f*7. Both RWCA and WCA could constantly obtain the global optima for *f*6. Moreover, the RWCA was able to find the global optima with less St.dev than WCA for all unimodel functions, these small values shows the RWCA robustness in finding global optima.

Table 2 recorded the statistical results of RWCA and WCA in solving the multimodal and fixed-modal functions. From Table 2, the RWCA obtain better best, Mean and worst results for *f*8. For *f*12–*f*17 both RWCA and WCA were able to obtain the global optimum. Furthermore, RWCA provide better mean, worst and St.dev for *f*10, *f*18, *f*19 functions and a stable global optima in term of best, mean and worst for *f*9 and *f*11.

A graphical demonstration of the convergence of the RWCA and WCA on some of the unimodal and multimodal functions is shown in Fig 2. Figure 2 shows that RWCA obtains the global optima faster than WCA without trapping in local optima. This is due to applying a nonlinear convergence parameter and Cauchy distribution, which helps the RWCA to effectively explore the solution space and find the global optima as fast as conceivable.

The RWCA was compared with six state-of-the-art met-heuristic algorithms: Whale Optimization Algorithm (WOA) [52], Particle Swarm Optimization (PSO) [16], Dragonfly Algorithm (DA) [50], Gravitational Search Algorithm (GSA) [61], Moth-Flame Optimization (MFO) [49], and Differential Evolution (DE) [68]; as reported in Table 3. From Table 3, the proposed RWCA algorithm significantly outperformed the other compared algorithm for six test functions *f*1–*f*3, *f*10, *f*12 and *f*13 in term of mean and St.dev measure. Moreover, the RWCA could consistently obtain the theoretical global optima (0) for *f*9 and *f*11 similar to WOA and DE algorithms respectively. For functions *f*14–*f*16 RWCA could find the global optimum similar to all other algorithms, however with a best St.dev for function *f*16. Compared to the DE algorithm, RWCA finds the second best results for function *f*18.

Table 1 Statistical results of WCA and RWCA algorithm on Unimodal functions

Function	WCA				RWCA			
	Best	Mean	Worst	St.dev	Best	Mean	Worst	St.dev
*f*1	7.6015e−39	1.9486e−36	2.8938e−35	5.2767e−36	3.2352e−97	5.9567e−62	1.787e−60	3.2626e−61
*f*2	1.3334e−19	5.1713e−18	2.2823e−17	5.3268e−18	6.3674e−54	5.9814e−39	1.7913e−37	3.2702e−38
*f*3	5.8724e−38	2.0684e−35	1.0404e−34	2.5528e−35	1.2026e−83	4.5493e−57	1.3643e−55	2.4908e−56
*f*4	1.3096e−19	2.2317e−18	1.0032e−17	2.5978e−18	2.1755e−42	4.4966e−27	1.2812e−25	2.3382e−26
*f*5	0	5.4174e−4	0.010836	2.1808e−7	0	1.9625e−07	3.9215e−6	9.6741e−8
*f*6	0	0	0	0	0	0	0	0
*f*7	0.0040279	0.50879	0.97407	0.33477	0.0013685	0.42772	0.96423	0.26622

Table 2 Statistical results of WCA and RWCA algorithm on multimodal functions

Function	WCA				RWCA			
	Best	Mean	Worst	St.dev	Best	Mean	Worst	St.dev
$f8$	− 3175.9	− 3425.9	− 1320.6	98.075	− 5965.6	− 4545.6	− 1988.9	921.5
$f9$	0	1.3604	2.9849	0.95989	0	0	0	0
$f10$	8.8818e–16	3.8488e–15	4.4409e–15	1.3467e–15	8.8818e–16	8.8818e–16	8.8818e–16	0
$f11$	0	0.10827	0.41874	0.099326	0	0	0	0
$f12$	1.1779e–31	1.1779e–31	1.1779e–31	2.227e–47	1.1779e–31	1.1779e–31	1.1779e–31	2.227e–47
$f13$	1.3498e–32	1.3498e–32	1.3498e–32	5.5674e–48	1.3498e–32	1.0498e–32	1.3498e–32	5.5674e–48
$f14$	0.398	0.398	0.398	0	0.398	0.398	0.398	0
$f15$	3	3	3	3.433e–15	3	3	3	3.1272e–15
$f16$	− 3.8628	− 3.8628	− 3.8628	3.1618e–15	− 3.8628	− 3.8628	− 3.8628	2.2749e–15
$f17$	− 3.29	− 3.26	− 2.26	6.146e2	− 3.29	− 3.28	− 2.26	6.046e2
$f18$	− 10.403	− 8.3149	− 1.8376	3.314	− 10.403	− 9.9028	− 2.7519	1.5309
$f19$	− 10.536	− 8.8419	− 2.4217	3.1802	− 10.536	− 9.8442	− 2.4217	1.4433

4.2 Constrained optimization problems

In this section, the efficiency of the proposed RWCA is evaluated by solving several numerical constrained optimization problems with distinct properties extracted from the CEC2006 competition [38]. The main properties of the constrained numerical problems are listed in Table 4. From Table 4, n is "the number of decision variables", ρ "the ratio between the size of the feasible search space and that the entire search space" , LI "the number of linear inequality constraints", NI "the number of nonlinear inequality constraints", NE "the number of nonlinear equality constraints", LE "the number of linear equality constraints" and a "the number of active constraints".

For all test problems the initial parameters for RWCA, Npop, Nsr , dmax and μ are chosen as 50, 4, 1E–03 and 0.4 respectively. Whereby, the maximization problems are converted into minimization using $-f(x)$.

The objective values, Best, Mean, Worst and St.dev for G1–G5 are obtained by performing 30 independent runs of WCA and RWCA algorithms and reported in Table 5. As described in Table 5, it can be seen that the proposed RWCA algorithm is able to find resulting "Mean" solutions very close to the global optima solutions for all problems G1–G5. With respect to the test problem G1, RWCA is able to find a typical "best" result to the global optimal solution $f(x) = -30665.53867$; and the "best" and "mean" typical of the the global optimal solution for problem G5. In addition to, it can be observed that the st.dev over 30 runs for the G1–G3 test problems are relatively small. In particular, for problems G5, st.dev of the objective function are equal to 0, which reflects that RWCA is robust and stable for solving constrained numeric problems. Compared with WCA, the proposed RWCA algorithm can obtain better results for the five test problems. RWCA algorithm found better "best","mean", "worst", and St.dev values for G1–G4, and similar "best","mean", "worst" Results for G5 and a better st.dev.

To further validate the performances of the proposed RWCA algorithm, RWCA is compared against seven innovatory algorithms from the literature: improved grey wolf optimization with modified augmented Lagrangian (MAL-IGWO) [43], Genetic algorithm (GA) [40], particle swarm optimization with differential evolution (PSO-DE) [41], hybrid differential evolution algorithm (HDE), modified artificial bee colony algorithm (MABC) [46], constrained optimization by artificial bee colony (COABC) [24] and modified global best artificial bee colony (MGABC) [6].

Table 6 lists the "best" , "mean" and "worst" obtained function values, to ensure the comparison fair; the experimental results of MAL-IGWO, GA, PSO-DE, HDE, MABC, COABC and MGABC reported in Table 6 were straightly taken from their literatures. For the test problem G1, as seen from Table 6, the proposed RWCA algorithm is able to find the global optimal solution in term of "best" similar to the seven compared algorithms. However, as seen MAL-IGWO, PSO-DE and COABC obtained the better "best" and HDE obtained the better"mean" results for problem G2. For the test problems G3, the proposed RWCA algorithm found better "best", "mean" and "worst" results than MABC; and found better "best" than GA , MABC and MGABC for problem G4. The above observation and analysis validate that the proposed RWCA is an effective algorithm for solving constrained numerical problems.

4.3 Analysis of parameter "'μ"

The main objective of this section is to analysis the effect of the μ parameter setting on the performance of RWCA. with the aim of investigating the sensitivity of the μ parameter,

Fig. 2 Convergence curves of WCA and RWCA

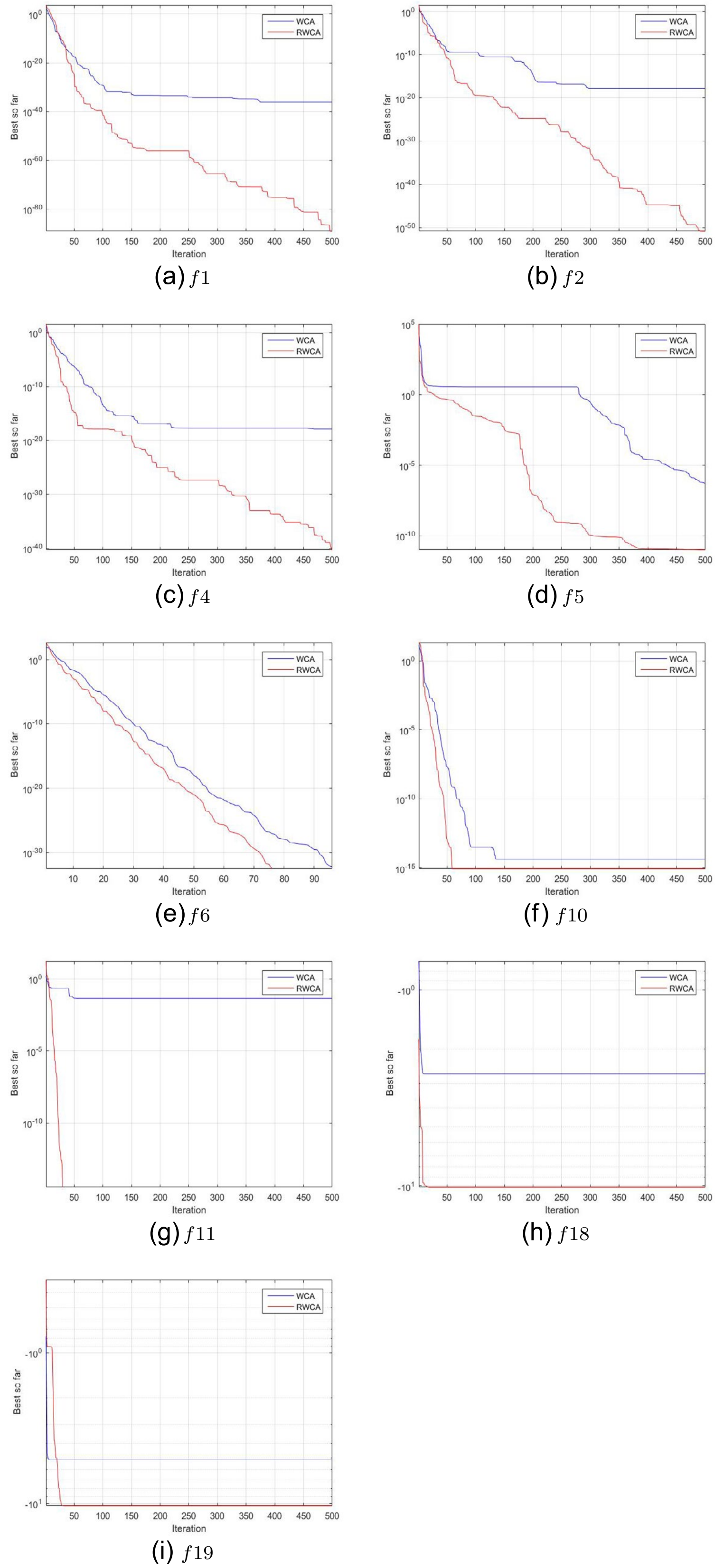

Table 3 Comparison results obtained for RWCA and different optimization algorithms

Function		RWCA	WOA	PSO	DA	GSA	MFO	DE
*f*1	Mean	**5.9567e–62**	1.41e–30	1.36	5.303e–1	2.53e–16	1.65e–31	8.2e–14
	St.dev	**3.2626e–61**	4.91e–30	2.02e–7	1.3180	9.67e–17	4.91e–31	5.9e–14
*f*2	Mean	**5.9814e–39**	1.06e–21	0.042144	2.392	0.055655	2.69e–19	1.5e–09
	St.dev	**3.2702e–38**	2.39e–21	0.045421	3.912	0.194074	6.22e–19	9.9e–10
*f*3	Mean	**4.5493e–57**	5.3901e–07	70.12562	215.45	896.5347	2.05e–11	6.8e–11
	St.dev	**2.4908e–56**	2.9310e–06	22.11924	935.17	318.9559	4.21e–11	7.4e–11
*f*4	Mean	4.4966e–27	0.072581	1.086481	1.153	7.35487	5.79e–06	**0**
	St.dev	2.3382e–26	0.39747	0.317039	2.702	1.741452	3.17e–05	**0**
*f*5	Mean	1.9625e–07	27.86558	96.71832	6784.5	67.54309	133.11	**0**
	St.dev	9.6741e–8	0.763626	60.11559	21974.5	62.22534	555.57	**0**
*f*6	Mean	**0**	3.116266	0.000102	2.2023	2.5e–16	4.78e–32	**0**
	St.dev	**0**	0.532429	8.28e–05	5.528	1.74e–16	1.27e–31	**0**
*f*7	Mean	4.2772e–6	**1.425e–7**	0.122854	6.9e–3	0.089441	1.2e–3	70.00463
	St.dev	0.26622	**1.149e–7**	0.044957	7.6e3	0.04339	7.2e–4	0.0012
*f*8	Mean	– 4545.6	– 5080.76	– 4841.29	– 3213.66	– 2821.07	– 3329.13	**– 11080.1**
	St.dev	921.5	695.7968	1152.814	431.748	493.0375	**288.317**	574.7
*f*9	Mean	**0**	**0**	46.70423	11.561	25.96841	12.8372	69.2
	St.dev	**0**	**0**	11.62938	10.177	7.470068	7.352	38.8
*f*10	Mean	**8.8818e–16**	7.4043	0.276015	3.14e–5	0.062087	8.88e–16	9.7e–08
	St.dev	**0**	9.897572	0.50901	1.7e–4	0.23628	1.00e–31	4.2e–08
*f*11	Mean	**0**	0.000289	0.009215	0.3846	27.70154	1.78e–01	**0**
	St.dev	**0**	0.00158	0.007724	0.3826	5.040343	8.43e–2	**0**
*f*12	Mean	**1.1779e–31**	0.339676	0.006917	0.5296	1.799617	3.11e–02	7.9e–15
	St.dev	**2.227e–47**	0.214864	0.026301	0.6912	0.95114	9.487e–2	8e–15
*f*13	Mean	**1.0498e–32**	1.889015	0.006675	0.5292	8.899084	1.10e–03	5.1e–14
	St.dev	**5.5674e–48**	0.266088	0.008907	0.7173	7.126241	3.33e–3	4.8e–14
*f*14	Mean	**0.398**	**0.398**	**0.398**	**0.3.98**	**0.398**	**0.398**	**0.398**
	St.dev	**0**	2.7e–05	**0**	7.60e13	**0**	1.13e–16	9.9e–09
*f*15	Mean	**3**	**3**	**3**	**3**	**3**	**3**	**3**
	St.dev	3.1272e–15	4.22e–15	**1.33e–15**	1.38e–06	4.17e–15	1.95e–15	2e–15
*f*16	Mean	**– 3.8628**	– 3.85616	**– 3.8628**	**– 3.86**	**– 3.8628**	**3.86**	N/A
	St.dev	**2.2749e–15**	0.002706	2.58e–15	1.587e–3	2.29e–15	2.71e–15	N/A
*f*17	Mean	– 3.28	– 3.2202	– 3.26634	– 3.25	**– 3.31778**	– 3.22	N/A
	St.dev	6.046e–2	0.098696	0.060516	6.720e–2	**0.023081**	4.5066e–2	N/A
*f*18	Mean	– 9.9028	– 8.18178	– 8.45653	– 10.4	– 9.68447	– 9.35	**– 10.403**
	St.dev	1.5309	3.829202	3.087094	0.192434	2.014088	2.423664	**3.9e–07**
*f*19	Mean	– 9.8442	– 9.34238	– 9.95291	– 10.3	**– 10.536**	– 10.3	**– 10.536**
	St.dev	1.4433	2.414737	1.782786	1.060781	**2.6e–15**	1.39948	1.9e–07

The best attained statistical results (mean and standard deviation) are highlighted in bold

Table 4 Description of constrained numerical problems

Function ID	$f(x)$ type	Problem type	n	$\rho(\%)$	LI	NE	NI	LE	a	$f_{optimal}$
G1	Quadratic	Min	5	52.1230	0	0	6	0	2	– 30665.53867
G2	Quadratic	Min	10	0.0003	3	0	5	0	6	24.306209
G3	Polynomial	Min	7	0.5251	0	0	4	0	2	680.630057
G4	Linear	Min	8	0.0010	3	0	3	0	6	7049.24802
G5	Quadratic	Max	3	4.7713	0	0	1	0	0	1.00000

several experiments for solving five constrained numerical problems (G1–G5) with 30 independent runs have been performed.

As mentioned previously, the μ parameter has the proficiency of adjusting the range of searching region near the best solution. With the aim of the μ parameter study, all other parameters settings of RWCA were kept unchanged; and RWCA is tested with different values of μ: 0.1, 0.2, 0.3, 0.4 and 0.5.

The best and mean values of the constrained numerical problem objective function are summarized in Table 7 .Based on the results recorded in Table 7, a recommended value for the μ parameter is set to 0.4.

Table 5 Statistical results of WCA and RWCA algorithm on Constrained numerical problems

Function	RWCA				WCA			
	Best	Mean	Worst	St.dev	Best	Mean	Worst	St.dev
G1	− 30665.53867	− 30665.5379547	− 30665.5233568	0.00352299731	− 30666.75111	− 30666.74930	− 30666.7374	0.00355035
G2	24.40061143543	25.43319101453	26.4237858525	1.5431248221	24.47779949	28.105921756	32.79760011	2.38927372
G3	680.630478509	680.637660382	680.684003898	0.0391426637583	680.6517860	680.7722062	680.9591332	0.07991533
G4	7049.32961241	8317.15836709	13120.516179989	1240.694873032	7109.138173	9531.775360	18342.00464	2443.65135
G5	− 1	− 1	− 1	0	− 1	− 1	− 1	1.2711E–16

Table 6 Comparison of RWCA results with regard to different algorithms on constrained numerical problems

Function		RWCA	MAL-IGWO	GA	PSO-DE	HDE	MABC	COABC	MGABC
G1	Best	− 30665.5386713	− 30665.53868	− 30665.539	− 30665.5387	− 30665.54	− 30665.539	− 30665.5387	− 30665.54
	Mean	− 30665.5376881	− 30665.53867	− 30665.539	− 30665.5387	− 30665.54	− 30665.539	− 30665.5387	− 30665.54
	Worst	− 30665.5233568	− 30665.53867	− 30665.539	− 30665.5387	− 30665.54	− 30665.539	− 30665.5387	− 30665.54
G2	Best	24.40061143543	24.306209	24.333	24.3062091	24.31	24.315	24.3062	24.32653
	Mean	25.43319101454	24.31	24.387	24.3062100	24.306209	24.415	24.3062	24.78064
	Worst	26.4237858526	24.306209	24.427	24.3062172	24.31	24.854	24.3062	25.09927
G3	Best	680.630478509	680.630037	680.631	680.6300574	680.63	680.632	680.63005	680.6302
	Mean	680.637660382	680.630052	680.634	680.6300574	680.63	680.647	680.63005	680.6309
	Worst	680.684003898	680.630057	680.637	680.6300574	680.63	680.691	680.63005	680.6322
G4	Best	7049.32961241	7049.23670	7049.861	7049.248021	7049.25	7051.706	7049.248	7104.006
	Mean	8317.15836709	7049.23678	7131.084	7049.248038	7049.34	7233.882	7049.248	7357.461
	Worst	13120.5161799	7049.23693	7263.461	7049.248233	7050.23	7473.109	7049.248	7504.944
G5	Best	− 1	− 1	− 1	− 1	− 1	− 1	− 1	− 1
	Mean	− 1	− 1	− 1	− 1	− 1	− 1	− 1	− 1
	Worst	− 1	− 1	NA	− 1	− 1	− 1	− 1	− 1

Table 7 Experimental results of RWCA with varying μ values

Function		$\mu = 0.1$	$\mu = 0.2$	$\mu = 0.3$	$\mu = 0.4$	$\mu = 0.5$
G1	Best	− 30665.53867086600	− 30665.53867130919	− 30665.53867146241	− 3.06655386712672	− 30665.53867091266
	Mean	− 30665.53456178970	− 30665.53768806154	− 30665.53738850630	− 3.066553795477157	− 30665.53805278325
G2	Best	24.686819946426752	24.713979261467394	24.672498912860164	24.400611435439497	24.654308616123654
	Mean	27.169967121020655	27.189462664782805	27.157627411534797	26.433191014537027	27.087088854785282
G3	Best	680.6360629694065	680.6444462398060	680.6351110564727	680.6304785096923	680.6369084149803
	Mean	680.7077412763562	680.6958662080972	680.6963826195898	680.6376603822925	680.6946003870487
G4	Best	7.132043920970878	7098.566179492316	7065.331807218106	7049.329612417167	7.119826056831297
	Mean	8.476611113404624	8619.069093291610	8316.651175871899	8317.158367097893	8.924311885842019
G5	Best	− 1	− 1	− 1	− 1	− 1
	Mean	− 1	− 1	− 1	− 1	− 1

4.4 Constrained engineering problems

In this section, the proposed RWCA performance is evaluated on real-world constrained problems, 3 well-studied engineering design problems have been solved: (1) Tension/compression spring design, (2) pressure vessel design and (3) welded beam design problem. RWCA algorithm parameters values for solving the constrained engineering problems were kept the same set as follows: Npop=50, Nsr=4 , dmax = 1E–03 and μ =0.4.

4.4.1 Tension/compression spring design problem

The tension/compression spring design problem includes three continuous variables and four non-linear inequality constraints. The problem was presented by Arora [4] to minimize the weight *f*(*x*) of a tension/compression spring subject to the following constraints: “minimum deflection” (g1), “shear stress” (g2), “surge frequency” (g3), and “limits on outside diameters” (g4). The three design variables are the wire diameter d, the mean coil diameter D, and the number of active coils P, as shown in fig 3; the variable vector is given by:

$$X = (d, D, P) = (x_1, x_2, x_3)$$

Tables 8 and 9 compare the statistical optimization results obtained by RWCA, WCA and various optimization algorithms reported in the literature on tension/compression spring design problem.

From the comparison results listed in Table 9, the proposed RWCA obtain the second best in term of ”best”, ”mean” and ”worst”. Moreover, RWCA outperforms all other algorithms, in terms of standard deviation results; which indicates the RWCA robustness in finding the global optima solution.

4.4.2 Pressure vessel design problem

The pressure vessel design (PVD) problem is a mixed-integer optimization problem introduced in [22]; where a cylindrical pressure vessel with two hemispherical heads at both ends is deliberate for minimum fabrication cost, considering the material cost, forming and welding. The problem mathematical model has one nonlinear (g3) and three linear inequality (g1, g2 and g4) constraints. The design variables associated with this problem includes two continuous variables : internal radius ”R” and length of the vessel without heads ”L”; and two discrete design variables: “the depth of the shell” T_s and “thickness of the head” T_h, as shown in Fig. 4. Whereby, the variable vector of the PVD problem to be optimized is:

$$X = (T_s, T_h, R, L) = (x_1, x_2, x_3, x_4)$$

The pressure vessel design problem has been solved with regard to the following design variables bounds:

Model I : $1 \le x_{1,2} \le 99$ *and* $10 \le x_{3,4} \le 200$

Moreover, intending to examine the entire constrained region, the upper limit range of the design variable x_4 has extended:

Model II:$1 \le x_{1,2} \le 99, \quad 10 \le x_3 \le 200$ *and* $10 \le x_4 \le 240$

Tables 10 and 11 compare the optimization results obtained by the proposed RWCA ,basic WCA and other several algorithms reported in the literature to solve PV problem. The results of compared algorithms are taken directly from their original papers, while ”NA” denotes the results are not available. From Table 11, it can be realized that for model I: compared to BWO, RWA reported second best results. While, RWCA got better results in term of “best, mean and worst” compared to WCA and all other algorithms. Meanwhile for model II: RWCA was able to find the best results in term of best, and the second best in term of mean measure.

4.4.3 Welded beam design problem

Welded beam design problem was proposed by [65], the objective of this problem is to obtain the fabrication cost minimum value of the welded beam subject to inequality constraints. The optimization constraints are: “shear stress” τ, “bucking load on the bar” P_b, “bending stress in the beam” σ, “deflection rate of the beam” δ and side constraints. The problem includes four design variables for optimizing the fabrication cost: “ thickness of the weld” h, “length of the

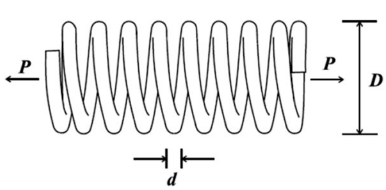

Fig. 3 Tension/compression spring design problem Schema

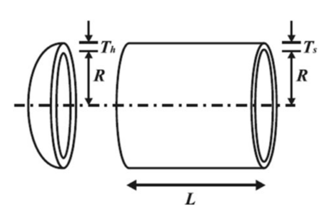

Fig. 4 Pressure vessel design problem Schema

bar" l, "thickness of the bar" b and "height of the bar" t, Fig. 5.

$X = (h, l, t, b) = (x_1, x_2, x_3, x_4)$

The performance of the RWCA for solving the Welded beam design problem is assessed and compared with WCA and other algorithms from the literature that were proposed to solve the problem, the comparisons results are given in Tables 12 and 13.

Referring to Table 12, the solutions realized by the proposed RWCA on the basis of "best, mean, worst" objective function values and std.dev are better than those obtained by the WCA. From the results in Table 13, RWCA outperformed other reported results in term of best, mean and worst except for those obtained by PSO-DE and BWO algorithm.

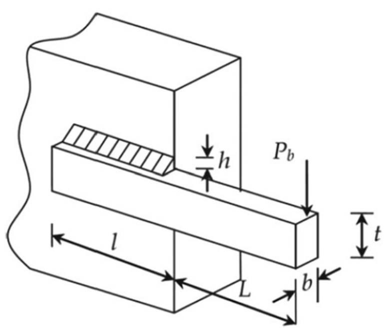

Fig. 5 Welded beam design problem Schema

Table 9 Comparison results of RWCA with different algorithms for the tension/compression spring problem

Method	Best	Mean	Worst	St.dev
SCPSO [45]	0.0126652	0.0127576	0.0146117	2.70e–04
GSA [61]	0.0128739	0.0134389	0.0142117	1.34e–02
GDA [7]	0.012665	0.012875	0.014079	2.97e–04
GWO [54]	0.0126723	0.0126971	0.0127208	2.10e–05
MVO [48]	0.0128169	0.0144644	0.0178397	1.62e–03
TEO [34]	0.012665	0.012685	0.012715	4.41e–06
EEGWO [42]	0.012665	0.012685	0.012720	2.22e–05
CWCA III [30]	0.012672	0.013401	0.016811	0.4578e–03
BWO [27]	0.0126029	0.012613028	0.0126421	2.6e–5
WCA [20]	0.012665	0.012746	0.012952	8.06e–05
RWCA	0.0126652	0.0126828	0.012773	1.00e–05

5 Conclusion

This paper proposed a reconfigured water cycle algorithm (RWCA) for solving unconstrained, constrained numerical and real world engineering problems. Firstly, a modified position-updated strategy was proposed to improve the exploration ability of the RWCA algorithm. The updating strategy of the proposed RWCA algorithm allows the sea to update their position using Cauchy distribution, then a greedy procedure between the current sea and the cauchy sea is applied. In this greedy procedure, the position which is better in term of the fitness cost is selected. As a results of the cauchy distribution strong disrupting, it can promote

Table 8 Comparison of the best solution for various algorithms on the tension/compression spring problem

Method	x_1	x_2	x_3	f_{min}
SCPSO [45]	0.051688	0.356705	11.289687	0.0126652
GSA [61]	0.317312	0.05000	14.22867	0.0128739
GDA [7]	0.0516925	0.3568108	11.2835059	0.012665
GWO [54]	0.051178	0.344541	12.04249	0.0126723
MVO [48]	0.05000	0.315956	14.22623	0.0128169
TEO [34]	0.051775	0.358792	11.168390	0.012665
EEGWO [42]	0.051673	0.35634	11.3113	0.012665
WOA [53]	0.051207	0.345215	12.0043032	0.0126763
BWOA [10]	0.051602	0.357488	11.2441198	0.0126654
MAL-IGWO [43]	0.0517030133	0.3570705348	11.269309420	0.01266523
CWCA III [30]	0.05170910	0.3571073	11.27082577	0.01267157
NRO [73]	0.05168896	0.356715313	11.289108 018	0.01266523
BWO [27]	0.051066	0.342967	12.091428	0.0126029
RFO [60]	0.05189	0.36142	11.58436	0.01321
PBO [59]	0.05102	0.35756	11.6994	0.01275
WCA [20]	0.051680	0.356522	11.300410	0.012665
RWCA	0.05000	0.3174254002225493	13.95995278	0.01266523

Table 10 Comparison of the best solution for various algorithms on the PVD problem

Model	Method	x_1	x_2	x_3	x_4	f_{min}
I	G-QPSO [65]	0.8125	0.4375	42.0984	176.6372	6059.7208
	GDA [7]	0.8125	0.4375	42.0975	176.6484	6059.8391
	MBA [62]	0.7802	0.3856	40.4292	198.4964	5889.3216
	CSA [5]	0.8125	0.4375	42.0984	176.6366	6059.7144
	SCA [51]	0.817577	0.417932	41.74939	183.5727	6137.3724
	TEO [34]	0.8125	0.4325	42.0984	173.6366	6059.71
	EEGWO [42]	13.09291	6.792196	42.09758	176.6495	6059.8704
	Hybrid IWO [11]	1.24000	0.5061	54.20	97.3	5980.000
	BWO [27]	0.777821	0.373174	39.9973587	199.93614	5796.0389
	RFO [60]	0.81425	0.44521	42.20231	176.62145	6113.3195
	PBO [59]	0.81327	0.43702	42.04601	176.75597	6057.54657
	WCA [20]	0.7781	0.3846	40.3196	200.0000	5885.3327
	RWCA	0.7781	0.3846	40.3195	200.0000	5885.3318
II	FSA [28]	0.7683257	0.3797837	39.8096222	207.2255595	5868.764836
	PSOStr [14]	0.75	0.375	38.86010	221.36549	5850.38306
	Mixed-FA [21]	0.75	0.375	38.86010	221.36547	5850.38306
	IHS [44]	0.75	0.375	38.86010	221.36553	5849.76169
	WCA	0.7275	0.359648	37.699014	240.000	5804.37697
	RWCA	0.9020	0.445885	46.738534	126.5267	5804.37675

Table 11 Comparison results of RWCA with different algorithms for the PV problem

Model	Method	Best	Mean	Worst	St.dev
I	G-QPSO [65]	6059.7208	6440.3786	7544.4925	448.4711
	GDA [7]	6059.8391	6149.7276	6823.6024	210.77
	MBA [62]	5889.3216	6200.6476	6392.5062	160.34
	CSA [5]	6059.7144	6342.4991	7332.8416	384.9454
	SCA [51]	6137.3724	6326.7606	6512.3541	126.609
	TEO [34]	6059.71	6138.61	6410.19	129.9033
	EEGWO [42]	6059.8704	6066.7220	6091.0922	10.64121
	Hybrid IWO [11]	5980.000	6500.00	7300.00	352.000
	BWO [27]	5796.0389	5799.0214	5801.2113	002.073
	WCA [20]	5885.3327	6198.6172	6590.2129	213.0490
	RWCA	5885.3318	6059.1913	6319.0024	201.86895
II	FSA [28]	5868.764836	6164.585867	6804.328100	257.473670
	PSOStr [14]	5850.38306	NA	NA	NA
	Mixed-FA [21]	5850.38306	5937.33790	6258.96825	164.54747
	IHS [44]	5849.76169	NA	NA	NA
	WCA	5804.3769794	6354.02071796	7319.00187	596.42664
	RWCA	5804.3767571	6113.78039566	7005.107262963	383.0020

the randomness of RWCA and improve the algorithm global search ability.

Secondly, a non-linear control parameter strategy was adopted to balance the harmony between exploitation and exploration, which leads to improve the convergence precision and speed. Moreover, for solving constrained problems the impact of parameter μ was manipulated and a recommended value for the μ parameter was set. 19 benchmark test functions, five constrained numerical and three engineering applications were laborious in order to validate and verify the performance of the proposed RWCA algorithm. Experimental results showed that, the proposed RWCA algorithm provides highly competitive results compared to WCA and other well known optimization algorithms. The main reason is that, the proposed RWCA uses hybridization of major advantages of the WCA, Cauchy distribution, non-linear control parameter and sensitivity adjustment of the μ parameter. Consequently, it can be designated that the proposed

Table 12 Comparison of the best solution for various algorithms on the Welded beam design problem

Method	x_1	x_2	x_3	x_4	f_{min}
FSA [28]	0.2443	6.2158	8.2939	0.2443	2.3811
HEAA [71]	0.2444	6.2175	8.2915	0.2444	2.3810
EEGWO [42]	0.2444	6.2170	8.2928	0.2444	2.3813
WOA [53]	0.205396	3.484293	9.037426	0.206276	1.730499
Mixed-FA [21]	0.2015	3.562	9.0414	0.2057	1.73121
GWO [54]	0.205676	3.478377	9.03681	0.205778	1.72624
MPSO [32]	0.20573	3.47049	9.03662	0.20573	1.72485084
MVO [48]	0.205463	3.473193	9.044502	0.205695	1.72645
NRO [73]	0.205729	3.4704887	9.036624	0.2057296	1.72485231
AATM [70]	0.2441	6.2209	8.2982	0.2444	2.3823
BWO [27]	0.198694	3.421708	9.028637	0.200138	1.663761
RFO [60]	0.21846	3.51024	8.87254	0.22491	1.86612
PBO [59]	0.23258	3.49706	8.91026	0.21194	1.79863
WCA [20]	0.205728	3.470522	9.036620	0.205729	1.724856
RWCA	0.205855719	3.4688437	9.0338522	0.2058559	1.72485424

Table 13 Comparison results of RWCA with other algorithms for the Welded beam design problem

Method	Best	Mean	Worst	St.dev
FSA [28]	2.3811	2.4042	2.4890	NA
HEAA [71]	2.3810	2.3810	2.3810	1.30E–05
EEGWO [42]	2.3813	2.3817	2.3824	4.18E–04
PSO-DE [41]	1.7248531	1.7248579	1.7248811	4.1E–06
CDE [33]	1.733461	1.768158	1.824105	2.2E–02
AATM [70]	2.3823	2.3870	2.3916	2.20E–03
GWO [54]	1.725571	1.726558	1.730932998	1.056486E–03
BWO [27]	1.663761	1.665621	1.664165	2.79E–04
WCA [20]	1.724856	1.726427	1.744697	4.29E–03
RWCA	1.724854247763787	1.7249417	1.744356418	1.68562E–03

RWCA sustains competitiveness on solving unconstrained, constrained and engineering practical problems with complex search spaces.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abbassi R, Abbassi A, Heidari AA, Mirjalili S (2019) An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models. Energy Convers Manag 179:362–372
2. Al-Rawashdeh G, Mamat R, Rahim NHBA (2019) Hybrid water cycle optimization algorithm with simulated annealing for spam e-mail detection. IEEE Access 7:143721–143734
3. Ali MZ, Awad NH, Suganthan PN, Duwairi RM, Reynolds RG (2016) A novel hybrid cultural algorithms framework with trajectory-based search for global numerical optimization. Inf Sci 334:219–249
4. Arora JS (2004) Introduction to optimum design. Academic Press, London
5. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12
6. Bansal JC, Joshi SK, Sharma H (2018) Modified global best artificial bee colony for constrained optimization problems. Comput Electr Eng 67(2):365–382
7. Baykasoğlu A (2012) Design optimization with chaos embedded great deluge algorithm. Appl Soft Comput 12:1055–1067
8. Blum C, Roli A (2003) A survey on optimization metaheuristics. Metaheuristics in combinatorial optimization: overview and conceptual comparison 35:268–308
9. BoussaD I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117
10. Chen H, Xu Y, Wang M, Zhao X, Balanced A (2019) Whale optimization algorithm for constrained engineering design problems. Appl Math Modell. https://doi.org/10.1016/j.apm.2019.02.004
11. Cuevas E, Rodríguez A, Valdivia A, Zaldîvar D, Pérez M (2019) A hybrid evolutionary approach based on the invasive weed

optimization and estimation distribution algorithms. Soft Comput 3:13627–13668
12. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186:311–338
13. Deng W, Yao R, Zhao H, Yang X, Li G (2017) A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm. Soft Comput 23:2445–2462
14. Dimopoulos GG (2007) Mixed-variable engineering optimization based on evolutionary and social metaphors. Comput Methods Appl Mech Eng 196:803–817
15. Dorigo M, Birattari M (2010) Ant colony optimization. Springer, Berlin
16. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceeding of the sixth international symposium on micro machine and human science. pp. 39–43. Nagoya, Japan
17. Eid HF (2018) Grey wolf optimizer with v-shaped transfer function for attribute selection and classification. Al Azhar Bull Sci 29(2):11–17
18. Eid HF, Garcia-Hernandez L, Abraham A (2021) Spiral water cycle algorithm for solving multi-objective optimization and truss optimization problems. Eng Comput. https://doi.org/10.1007/s00366-020-01237-y
19. Eid HF, Muda AK (2019) Adjustive reciprocal whale optimization algorithm for wrapper attribute selection and classification. Int J Image Gr Signal Process 3:18–26
20. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 111:151–166
21. Gandomi AH, Yang XS, Alavi AH (2011) Mixed variable structural optimization using firefly algorithm. Comput Struct 89:2325–2336
22. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35
23. Gang X (2013) An adaptive parameter tuning of particle swarm optimization algorithm. Appl Math Comput 219:4560–4569
24. Gao W, Huang L, Luo Y, Wei Z, Liu S (2018) Constrained optimization by artificial bee colony framework. IEEE Access 6:73829–73845
25. Gao W, Liu S, Huang L (2013) A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. IEEE Trans Cybern 43(3):1011–1024
26. Guney K, Basbug S (2015) A quantized water cycle optimization algorithm for antenna array synthesis by using digital phase shifters. Int J RF Microw Comput-Aided Eng 25(1):21–29
27. Hayyolalam V, Pourhaji Kazem AA (2020) Black widow optimization algorithm: a novel meta-heuristic approach for solving engineering optimization problems. Eng Appl Artif Intell 87:103249
28. Hedar AR, Fukushima M (2006) Derivative-free filter simulated annealing method for constrained continuous global optimization. J Global Optim 35:521–549
29. Heidari AA, Faris H, Aljarah I, Mirjalili S (2018) An efficient hybrid multilayer perceptron neural network with grasshopper optimization. Soft Comput 23:7941–7958
30. Heidari A, Abbaspour RA, Jordehi AR (2017) An efficient chaotic water cycle algorithm for optimization tasks. Neural Comput Appl 28:57–85
31. Horst R, Pardalos P (2013) Handbook of global optimization. Springer, Berli
32. Hu X, Eberhart RC, Shi Y (2003) Engineering optimization with particle swarm. In: Proceedings of the 2003 IEEE Swarm intelligence symposium. p. 53–57. SIS'03
33. Huang FZ, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186(1):340–356
34. Kaveh A, Dadras A (2017) A novel meta-heuristic optimization algorithm: thermal exchange optimization. Adv Eng Softw 110:69–84
35. Khalilpourazari S, Shr P (2016) Bi-objective optimization of multi-product EPG model with backorders, rework process and random defective rate. In: IEEE 2nd international conference on industrial engineering, pp 36–40
36. Koupaei JA, Hosseini SMM, Ghaini FMM (2016) A new optimization algorithm based on chaotic maps and golden section search method. Eng Appl Artif Intell 50:201–214
37. Lenin K, Reddy BR, Kalavathi MS (2014) Water cycle algorithm for solving optimal reactive power dispatch problem. J Eng Technol Res 2(2):1–11
38. Liang J, Runarsson TP, Mezura-Montes E, Clerc M, Suganthan P, Coello CC, Deb K (2006) Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real parameter optimization. J Appl Mech 41:1–8
39. Liang JJ, B-y Q, Suganthan PN (2015) Problem definitions and evaluation criteria for the CEC 2015 special session and competition on single objective real parameter numerical optimization. Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Nanyang Technological University, Singapore
40. Lin CH (2013) A rough penalty genetic algorithm for constrained optimization. Inf Sci 241:119–137
41. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 10:629–640
42. Long W, Jiao J, Liang X, Tang M (2018) An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. Eng Appl Artif Intell 68:63–80
43. Long W, Liang X, Cai S, Jiao J, Zhang W (2017) A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems. Neural Comput Appl 28:421–438
44. Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188:1567–1579
45. Maruta I, Kim TH, Sugie T (2009) Fixed-structure h controller synthesis: a metaheuristic approach using simple constrained particle swarm optimization. Automatics 45:553–559
46. Mezura-Montes E, Cetina-Dominguez O (2012) Empirical analysis of a modified artificial bee colony for constrained numerical optimization. Appl Math Comput 218(22):10943–10973
47. Miller P(2010) The Smart Swarm: How understanding flocks schools, and colonies can make us better at communicating, decision making, and getting things done. Avery Publishing Group, Inc
48. Mirjalili S, Mirjalili SM, Hatamlou A (2016) A multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27:495–513
49. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature inspired heuristic paradigm. Knowl Based Syst 89:228–249
50. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete and multi-objective problems. Neural Comput Appl 27:1053–1073
51. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133
52. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
53. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

54. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
55. Mohamed TH, Abubakr H, Alamin MAM, Hassan AM (2020) Modified WCA-based adaptive control approach using balloon effect: electrical systems applications. IEEE Access 8:60877–60889
56. Muhammad MA, Mokhlis H, Naidu K, Amin A, Franco JF, Othman M (2020) Distribution network planning enhancement via network reconfiguration and dg integration using dataset approach and water cycle algorithm. J Mod Power Syst Clean Energy 8(1):86–93
57. Nesmachnow S (2014) An overview of metaheuristics: accurate and efficient methods for optimisation. Int J Metaheuristics 3:320–347
58. Pham D, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M (2005) The bees algorithm. Manufacturing Engineering Centre, Cardiff University, Cardiff, pp 1–57
59. Polap D, Wozniak M (2017) Polar bear optimization algorithm: meta-heuristic with fast population movement and dynamic birth and death mechanism. Symmetry 9(10):203. https://doi.org/10.3390/sym9100203
60. Polap D, Wozniak M (2021) Red fox optimization algorithm. Expert Syst Appl 166:114107
61. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
62. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13(5):2592–2612
63. Sadollah A, Eskandar H, Bahreininejad A, Kim J (2015) Water cycle algorithm for solving multi-objective optimization problems. Soft Comput 19:2587–2603
64. Salar M, Dizangian B, Mir M (2019) Optimum design of space trusses using water cycle algorithm. Adv Res Civ Eng 1(3):36–48
65. dos Santos Coelho L (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 37:1676–1683
66. Seif Z, Ahmadi MB (2015) An opposition-based algorithm for function optimization. Eng Appl Artif Intell 37:293–306
67. Sreeranjini KS, Murugan S (2017) Memory based hybrid dragonfly algorithm for numerical optimization problems. Expert Syst Appl 83:63–78
68. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359
69. Tuba M, Bacanin N (2014) Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems. Neurocomputing 143:197–207
70. Wang Y, Cai Z, Zhou Y (2009) Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization. Int J Numer Methods Eng. 77(11):1501–1534
71. Wang Y, Cai Z, Zhou Y, Fan Z (2009) Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct Multidiscip Optim 37:395–413
72. Wang Y, Cai Z, Zhou Y, Zeng W (2008) An adaptive tradeoff model for constrained evolutionary optimization. IEEE Trans Evol Comput 12(1):80–92
73. Wei Z, Huang C, Wang X, Han T, Li Y (2019) Nuclear reaction optimization: a novel and powerful physics-based algorithm for global optimization. IEEE Access 7:66084–66109
74. Winston PH (1992) Artificial intelligence, 3rd edn. Addison-Wesley, Boston
75. Yang XS (2011) Bat algorithm for multi-objective optimisation. Int J Bio-Inspir Comput 3:267–274
76. Yao X, Liu Y (1996) Fast evolutionary programming. MIT Press, Cambridge, pp 451–460
77. Zhang Y, Wang S, Ji G (2015) comprehensive survey on particle swarm optimization algorithm and its applications. Math Probl Eng. https://doi.org/10.1155/2015/931256