An Efficient Double Adaptive Random Spare Reinforced Whale Optimization Algorithm

Article *in* Expert Systems with Applications · October 2019

DOI: 10.1016/j.eswa.2019.113018

4 authors, including:

Huiling Chen
Wenzhou University

Ali Asghar Heidari
National University of Singapore

Xuehua Zhao
Shenzhen Institute of information technology

Expert Systems With Applications xxx (xxxx) xxx

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An efficient double adaptive random spare reinforced whale optimization algorithm

Q1 Huiling Chen[a], Chenjun Yang[a], Ali Asghar Heidari[b,c], Xuehua Zhao[d,*]

[a] Department of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China
[b] School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran
[c] Department of Computer Science, School of Computing, National University of Singapore, Singapore
[d] School of Digital Media, Shenzhen Institute of Information Technology, Shenzhen 518172, China

ARTICLE INFO

Article history:
Received 8 June 2019
Revised 28 September 2019
Accepted 10 October 2019
Available online xxx

Keywords:
Whale optimization
Engineering design
Swarm-intelligence
Global optimization
Nature-inspired computing

ABSTRACT

Whale optimization algorithm (WOA) is a newly developed meta-heuristic algorithm, which is mainly based on the predation behavior of humpback whales in the ocean. In this paper, a reinforced variant called RDWOA is proposed to alleviate the central shortcomings of the original method that converges slowly, and it is easy to fall into local optimum when dealing with multi-dimensional problems. Two strategies are introduced into the original WOA. One is the strategy of random spare or random replacement to enhance the convergence speed of this algorithm. The other method is the strategy of double adaptive weight, which is introduced to improve the exploratory searching trends during the early stages and exploitative behaviors in the later stages. The combination of the two strategies significantly improves the convergence speed and the overall search ability of the algorithm. The advantages of the proposed RDWOA are deeply analyzed and studied by using typical benchmark examples such as unimodal, multi-modal, and fixed multi-modal functions, and three famous engineering design problems. The experimental results show that the exploratory and exploitative tendencies of WOA and its convergence mode have been significantly improved. The RDWOA developed in this paper is a promising improved WOA variant, and it has better efficacy compared to other state-of-the-art algorithms.

1. Introduction

Swarm intelligence and evolutionary optimizers mainly try to simulate the evolutionary and swarm-based hunting, foraging, surviving, and persuading processes in nature (Aljarah et al., 2018; Mafarja et al., 2018; Xu, Chen, Heidari, et al., 2019). In recent years, these optimizers have found their significant role in realizing many large-scale and real-world problems (Faris et al., 2018; Faris, Ala'M, et al., 2019, Faris, Heidari, et al., 2019). In terms of efficiency, these algorithms have proven to be more efficient than gradient-based algorithms (Zhang, Wang, Zhou, & Ma, 2019). Some well-known methods are classical particle swarm optimization (PSO) (Kennedy & Eberhart, 1995; Xiaoqin Zhang, Hu, Qu, & Maybank, 2010), simulated annealing (SA) (Kirkpatrick, Gelatt Jr, & Vecchi, 1983), genetic algorithms (GA) (Booker, Goldberg, & Holland, 1989), ant colony al-
Q2 gorithm (ACO) (Deng, Xu, & Zhao, 2019; Dorigo & Blum, 2005), the algorithms proposed in recent years include artificial bee colony algorithm (ABC) (Karaboga & Basturk, 2007), harris hawks optimizer (HHO) (Heidari et al., 2019), multi-verse optimization algorithm (MVO) (Mirjalili, Mirjalili, & Hatamlou, 2016), moth-flame optimization algorithm (MFO) (Mirjalili, 2015; Xu et al., 2019, Xu, Chen, Luo, et al., 2019), fruit fly optimization algorithm (FOA) (Shen et al., 2016; Zhang et al., 2019), sine cosine algorithm (SCA) (Chen et al., 2019; Mirjalili, 2016), grasshopper optimization algorithm (GOA) (Saremi, Mirjalili, & Lewis, 2017), bat-inspired algorithm (BA) (Yang, 2010; Yu, Zhao, Wang, Chen, & Li, 2019), grey wolf optimizer (GWO) (Cai et al., 2019; Heidari & Pahlavani, 2017), chicken swarm optimization algorithm (CSO) (Meng, Liu, Gao, & Zhang, 2014), and fish-swarm algorithm (AFSA) (Li, Shao, & Qian, 2002). WOA is a meta-heuristic algorithm proposed by Mirjalili and Lewis (2016). It works based on some predatory behaviors of whales in the process of foraging to achieve the possible optimal or suboptimal solutions.

In recent years, WOA has been widely applied to feature selection (Mafarja & Mirjalili, 2018; Zheng et al., 2018), retinal vascular recognition (Hassan & Hassanien, 2017), neural network (Aljarah, Faris, & Mirjalili, 2016; Sun & Wang, 2017), image segmentation

* Corresponding author.
E-mail addresses: chenhuiling.jlu@gmail.com (H. Chen), 1396207543@qq.com (C. Yang), as_heidari@ut.ac.ir, comaliasgha@comp.nus.edu.sg, t0917038@u.nus.edu (A.A. Heidari), zhaoxh@sziit.edu.cn (X. Zhao).

https://doi.org/10.1016/j.eswa.2019.113018

(Aziz, Ewees, & Hassanien, 2017), image retrieval (Aziz, Ewees, & Hassanien, 2018), key recognition (Thanga Revathi, Ramaraj, & Chithra, 2018), wind speed prediction (Wang, Du, Niu, & Yang, 2017), emotional analysis (Tubishat, Abushariah, Idris, & Aljarah, 2018), and so on. WOA has the characteristics of simple principle and strong global search ability that assist it in being better compared to some algorithms such as PSO and SCA in terms of solution quality. However, for complex and high-dimensional functions, WOA still has the characteristics of two common problems of swarm intelligence optimizers: slow convergence rate and low quality of solutions in the later stage of the iteration process. To mitigate these shortcomings in convergence rate and stagnation, researchers have improved WOA in various aspects. Zhou, Ling, and Luo (2018) proposed a Lévy flight trajectory-based WOA for engineering optimization called LWOA. In the algorithm, the Lévy flight trajectory is helpful to improve the diversity of population to premature convergence and enhance the ability to jump out of local optimal solution. Mafarja and Mirjalili (2018) proposed the binary version and the integration of several evolutionary operators (selection, crossover, and mutation). Yousri, Allam, and Eteiba (2019) proposed to combine the standard WOA with ten different chaotic maps to optimize some of its parameters. CWOA variants and standard WOA versions are also proposed to estimate the chaotic behavior parameters of PMSM under off-line and on-line operation conditions when there is no noise or noise. It provides a lower error value, a higher convergence rate, and a shorter execution time between the estimation and the performance of the original system. Khashan, Elhosseini, Haikal, and Badawy (2018) proposed to balance local and global searches by using non-linear and random "a" parameter variation and updating parameter "c" with inertial weight strategy. Sun and Wang (2017) proposed the idea of introducing chaos into the initialization process of WOA, and proposed a chaotic method based on the concept of chaos, using chaotic features to improve the diversity of search objects and the self-centeredness of search objects.

In this paper, a new WOA is proposed to improve the performance of WOA by introducing double adaptive weight strategy and random spare strategy. The proposed algorithm is named RDWOA. In RDWOA, double adaptive weights w_1 and w_2 are introduced, w_1 enhances the global search ability in the first half of iteration, w_2 enhances the local search ability in the second half, thus balancing the local and global search ability. Random spare mechanism is embedded to make each individual in the algorithm have the chance to replace the optimal value in each dimension, to break through the local optimum. In this algorithm, the two mechanisms can be well balanced. By combining these strategies, the algorithm has achieved higher search accuracy and convergence speed.

In order to verify the effectiveness of this method, 29 representative benchmark functions were selected from 23 standard functions and CEC 2014 (Liang, Qu, & Suganthan, 2014) to validate the performance of RDWOA and other competitive competitors. These competitors include six recognized meta-heuristic algorithms: original BA, GWO, gravitational Search Algorithm (Rashedi, Nezamabadi-pour, & Saryazdi, 2009), SCA, FA and MFO, and five most advanced evolutionary algorithms: WOA based on elite reverse learning and evolutionary operator (IWOA) (Tubishat et al., 2018), Lévy Flight Trajectory-Based WOA (LWOA) (Ling, Zhou, & Luo, 2017), Opposition-Based Sine Cosine Algorithm (OBSCA) (Abd Elaziz, Oliva, & Xiong, 2017), a LGMS-based improved FOA, (LGMS-FOA) (Shan, Cao, & Dong, 2013), a novel Bat Algorithm based on Collaborative and Dynamic Learning of Opposite Population (CDLOBA) (Yong, He, Li, & Zhou, 2018). The simulation results show that the proposed RDWOA is superior to all the involved competitive rivals, which proves the effectiveness of the developed RDWOA. Moreover, the obtained solutions for pressure vessel design (PVD), welded beam design (WBD) and tension/compression spring design (TCSD) are also superior to other algorithms reported recently, which shows that the proposed RDWOA can also solve the constraint problems.

This paper is organized as follows. Section 2 briefly describes WOA. The proposed RDWOA is described in detail in Section 3. Section 4 introduces and analyses RDWOA in parameter selection and benchmark function testing. The fifth part analyzes the experimental results of engineering problems. The sixth part summarizes the whole paper and looks forward to the future.

2. An overview of WOA

WOA is a meta-heuristic algorithm proposed by Mirjalili in 2016 (Mirjalili & Lewis, 2016), which simulates the hunting mode of humpback whales. Humpback whales dive underwater, spiral upward from about 12 m deep, and spit out bubbles of various sizes. These bubbles upswing to the shallow of the water at the same time, forming a spiral bubble net, which tightly surrounds the prey and pushes it towards the center of the net. It opens its mouth almost upright in the bubble circle and swallows the surrounded prey. WOA is to imitate the spiral bubble network strategy of humpback whales and forage through three mechanisms: spiral predation, random predation and encirclement, and contraction. WOA's mathematical model is described in the following sections.

2.1. Encircling prey

In nature, whales can identify the location of their prey and surround them for predation. WOA assumes that the optimal position in the current population is a prey; all other whale individuals surround the prey, while the location is updated by Eqs. (1) and (2):

$$D = |CX^*(t) - X(t)| \tag{1}$$

$$X(t+1) = X^*(t) - A \times D \tag{2}$$

where t is iteration, A and C represent coefficient vectors, $X^*(t)$ is the best position in the current population, A and C are obtained from Eqs. (3) and (4).

$$A = 2ar_1 - a \tag{3}$$

$$C = 2r_2 \tag{4}$$

Among them, r_1 and r_2 are random numbers in the range (0,1). The value of a decreases linearly from 2 to 0. t represents the current iteration number and $T_{\max}$ is the maximum iteration number.

2.2. Spiral bubble-net feeding maneuver

Humpback whales, when hunting, spiral upward toward their prey. In WOA, the whale uses Eq. (5) to update its position when it swims to the optimal individual.

$$X(t+1) = X^*(t) + D_p e^{bl} \cos(2\pi l) \tag{5}$$

Q3

Among them, $D_p = |X^*(t) - X(t)|$ denotes the distance between the optimal individual X before the update and the optimal position X_{best}, b is a constant defining the spiral shape, l is a random number between [−1,1]. Because the spiral predator of whales not only moves around the outer ring but also shrinks the enclosure, half of the probability in the mathematical model will choose the contraction mechanism such as Eq. (2) to update the location of whales.

Please cite this article as: H. Chen, C. Yang and A.A. Heidari et al., An efficient double adaptive random spare reinforced whale optimization algorithm, Expert Systems With Applications, https://doi.org/10.1016/j.eswa.2019.113018

2.3. Searching for prey

The whale searches and preys randomly according to its position. In WOA, the whale updates its position by Eqs. (6) and (7).

$$D = CX_{rand} - X(t) \tag{6}$$

$$\mathrm{X}(t+1) = \mathrm{X}_{rand} - A \times D \tag{7}$$

Among them, X_{rand} is a randomly selected whale position vector.

The pseudocode of WOA is shown as in Algorithm 1.

Algorithm 1 Pseudocode of WOA.

```
Initialize random distributed agents X_i (i=1,2,3....,n)
Calculate the fitness of each search agent
X*=the best search agent
while(FEs<MaxFEs)
  for each search agent
    Update a,A,C,L,and p
      if(p < 0.5)
        if(|A|<1)
          Update the position of search agent use Eq. (2)
        else if(|A| > 1)
          Select a random search agent (X_rand)
          Update the position of search agent use Eq. (7)
        end if
    else if(p > 0.5)
          Update the position with spiral Eq. (5)
    end if
  end for Check if any search agent goes beyond the search space and amend it
  Calculate the fitness of solutions
  Update X* if the method can detect a better solution
  t=t+1
end while
Return x*
```

3. Proposed RDWOA strategy

In this section, RDWOA is described in detail. Compared with the original algorithm, RDWOA adds two strategies. Firstly, in the original algorithm, RDWOA inspired by the characteristics that the adaptive weights of PSO are continually changing with the number of iterations, thus adding the strategy of double weights. In the first half of the algorithm, weight w_1 increases the global search ability, while weight w_2 in the second half of the algorithm increases the local search ability and the overall optimization ability of the algorithm. Then, a random spare strategy is introduced to the algorithm to improve the convergence speed and solution quality of the algorithm.

3.1. Random spare strategy

The random spare strategy is to replace the position vector on the *n*th dimension of the current individual with the value of the position vector on the current dimension of the optimal individual. In the process of searching, the original algorithm may have some suitable position vectors in some individual dimensions, but deficient position vectors in other dimensions. However, the position vectors in the optimal individual dimension are outstanding. So, we propose a random spare strategy to reduce the occurrence of this situation. Because not every position vector in an individual is bad, this strategy should be implemented with a certain probability and in the range from m to the end of the evaluation. Among them, m is the starting value. Through trial and error, we found that the result is the best when m is set to 0. Finally, we choose to compare the Cauchy random number with the ratio of the current number of evaluations to the total number of evaluations to determine whether to implement the random spare strategy in the global scope.

3.2. Double adaptive weight strategy

Weight is a critical parameter in PSO. Many studies change the adaptive weight to improve the PSO algorithm (Ratnaweera, Halgamuge, & Watson, 2004; Shi & Eberhart, 1999, 2001; Zhan, Zhang, Li, & Chung, 2009). Inspired by PSO, RDWOA also tries to change the local and global search ability of the algorithm by adding double adaptive weights. When dealing with multi-peak functions, the original WOA will soon fall into local optimum. Weight w_1 was added to improve the global search ability, and the weight w_2 was added to improve the local search ability in the later stage. w_1, w_2 can be formulated as in Eqs. (8) and (9).

$$w_1 = (1 - FEs/MaxFEs)^{1-\tan(pi\times(rand-0.5)\times s/MaxFEs)} \tag{8}$$

$$w_2 = (2 - 2 \times FEs/MaxFEs)^{1-\tan(pi\times(rand-0.5)\times s/MaxFEs)} \tag{9}$$

where s is changed with the local optimum degree of the algorithm. s is automatically added when the individual position is not updated, and s is divided by two to control the size of s when it is updated. Because of the addition of s and Cauchy random numbers, w_1 and w_2 do not decrease linearly, but fluctuate as the algorithm falls into local optimum. FEs is the current number of evaluations. The value of FEs is automatically increased by one for each evaluation. MaxFEs is the maximum number of evaluations. In the test, its value is 300,000. The range of w_1 is [0,1], and that of w_2 is [0.5,1]. Note that w_1 is added in the first half of the algorithm, such as Eq. (2) transforms into Eqs. (10) and (6) transforms into Eqs. (11) and (7) transforms into Eq. (12).

$$X(t+1) = w_1 \times \mathrm{X}^*(\mathrm{t}) - A \times D \tag{10}$$

$$X(t+1) = w_1 \times \mathrm{X}^*(\mathrm{t}) + \mathrm{D}_p e^{bl}\cos(2\pi l) \tag{11}$$

$$X(t+1) = w_1 \times X_{rand} - A \times D \tag{12}$$

where w_2 is added in the second half of the algorithm, such as Eq. (2) transforms into Eqs. (13) and (6) transforms into Eqs. (14) and (7) transforms into Eq. (15).

$$X(t+1) = X^*(t) - w_2 \times A \times D \tag{13}$$

$$X(t+1) = X^*(t) + w_2 \times D_p e^{bl}\cos(2\pi l) \tag{14}$$

$$X(t+1) = X_{rand} - w_2 \times A \cdot D \tag{15}$$

The pseudocode of RDWOA is shown in Algorithm 2. The flowchart of the proposed method is also shown in Fig. 1.

The computational complexity of RDWOA depends on population size (n), dimension size (d), and a maximum number of evaluations (*Max_Fes*). The number of iterations (t) is determined by the maximum number and total size of the function evaluation: t= *Max_Fes* /2*n*. After analysis, the overall time complexity is O (RDWOA) = O (initialization) +O (calculation of initial fitness and selection) +O (random spare mechanism) +O (introducing double weights to update the location of the whale population). Initialize O $(n \times d)$. The adaptability of all whale populations was assessed to be O $(n \times d)$. The complexity under random spare mechanism is O $(n \times d + 2 \times n \times d)$. Two weights are introduced to update the position O $(n \times d \times 2 \times 2 \times 2)$ of the whale population. Therefore, we get: O (RDWOA) = O $(n \times d)$+O $(n \times d \times 8)$+O $(n \times d + 2 \times n \times d)$.

Algorithm 2 Pseudocode of RDWOA.

```
# the pseudocode of the WOA algorithm
Initialize the whales population Xi(i=1,2,3....,n)
Calculate the fitness of each search agent and update s
X*=the best search agent
for each search agent
    if (tan(pi*(rand-0.5))<(1-FEs/MaxFEs))
        Replace the current search location with the optimal search location
    end if
    Calculate the fitness of each search agent and update s
    X*=the best search agent
end for
while(FEs<MaxFEs)
    for each search agent
      update a,A,C,L,and p
        if(p<0.5)
          if(|A|<1)
            if(FEs/MaxFEs>0.5)
              Update the position of search agent use Eq. (13)
            else
              Update the position of search agent use Eq. (10)
            end if
          else if(|A|>1)
            if(FEs/MaxFEs>0.5)
              Select a random search agent (Xrand)
              Update the position of search agent use Eq. (15)
            else
              Select a random search agent (Xrand)
              Update the position of search agent use Eq. (12)
            end if
          end if
        else if(p > 0.5)
          if(FEs/MaxFEs > 0.5)
              Update the position with spiral Eq. (14)
        else
              Update the position with spiral Eq. (11)
            end if
      end if
    end for
    Check if any search agent goes beyond the search space and amend it
    Calculate the fitness of each search agent
    Update X*if there is a better solution
    t=t+1
end while
Return x*
```

4. Experimental results and discussions

In this section, we first report the obtained comparative results and then discuss the observations in detail. Firstly, the parameters of the algorithm are analyzed. Then, the simulation experiments are carried out on 29 benchmark functions to verify the effectiveness of the proposed RDWOA comprehensively. Finally, the algorithm is applied to three engineering design problems.

4.1. Test functions

F1–F23 benchmark functions are some single modal, multi-modal, mixed and composite functions. They are all taken from 23 classical functions (Digalakis & Margaritis, 2001; Molga, 2005; Yang, 2010; Yao, Liu, & Lin, 1999), and F24–F29 from CEC2014 (Liang et al., 2014). The detailed description of 29 benchmark functions is shown in Table 1.

To obtain unbiased results, all algorithms are tested under the same settings: population size and maximum evaluation times are set to 30 and 300,000, respectively. For each benchmark function, all algorithms were tested 30 times independently. Friedman test is used to evaluate the results of all algorithms on the benchmark functions and give their rankings. Friedman test is a comparative test of non-parametric statistical test, which is used to find the difference between multiple test results. The Friedman test will rank the average performance of all selected methods for further statistical comparison and report ARV (average ranking value) in the comparison results. An algorithm with a smaller ARV will have a better performance.

4.2. Parameter values

Because the new random spare strategy uses a specific range of random spare, which affects the accuracy and convergence speed of the algorithm, the parameters are analyzed and selected in this section.

Table 2 gives the test results of 29 functions from F1 to F29 with starting values m of 0–0.9, respectively. In order to obtain better convergence, the optimal value of starting values in the random spare strategy is studied. In these experiments, the number of evaluations and group size were set to 300,000 and 30 respectively, and the dimension of search space was set to 30. Also, F1–F29 is used as a benchmark for 29 different functions. In this experiment, RDWOA with different starting values of random spare was performed on 29 functions. The results of each parameter value are executed 30 times independently. According to the results in Table 2, we can see that different starting values correspond to different ARVs. According to the results of ARVs, the starting value can be set to 0 as the best choice.

4.3. Impacts of components

As can be seen from Section 3, two strategies are introduced in the original WOA, namely, double weights and random spare. As shown in Table 3, three different WOAs are developed to study the effects of each strategy and any combination. Among them, “R” and “D” represent “random spare” and “double weight” respectively. In Table 3, “1” means that WOA uses this operation, while “0” means that the corresponding operation should not be used.

The 29 benchmark functions described in Table 1 are used to test the performance of various RDWOAs. We conducted non-parametric statistical tests, such as the Wilcoxon Symbol Level Test and Friedman Test. Table 4 lists the results of various RDWOA comparisons. Firstly, using 5% non-parametric statistical Wilcoxon symbol level test, the statistical difference between pairing algorithms is studied. As shown in Table 4, the results of P values between RDWOA and other algorithms on 29 functions (F1–F29) are listed in rows 2–30. The symbols “+”, “−” and “=” in the table indicate that RDWOA is superior to, inferior to, and equal to other algorithms, respectively. In the case of “+/−/=”, RDWOA is significantly different from these comparison methods. As we can see, RDWOA is better than RWOA, DWO, and WOA in 29 test functions. Moreover, 29 outcomes were better than them. Therefore, RDWOA is the best performing variant. Also, the Friedman test is used to test the average ranking values of these algorithms, and the differences among these methods are further studied. Obviously, according to these ranking results, we can find that RDWOA converges faster and has higher precision than RWOA in single peak functions, hybrid functions, and composite functions, and RDWOA has higher precision than DWOA in multi-module benchmark functions. Therefore, it verifies that RDWOA has the best performance in resolving these test functions compared with other combinations. Finally, according to the above analysis, RDWOA is chosen as the best improvement method of WOA.

4.4. Comparison with other well-established methods

In order to verify the effectiveness of RDWOA, it is firstly compared with several well established original algorithms: BA (Yang,

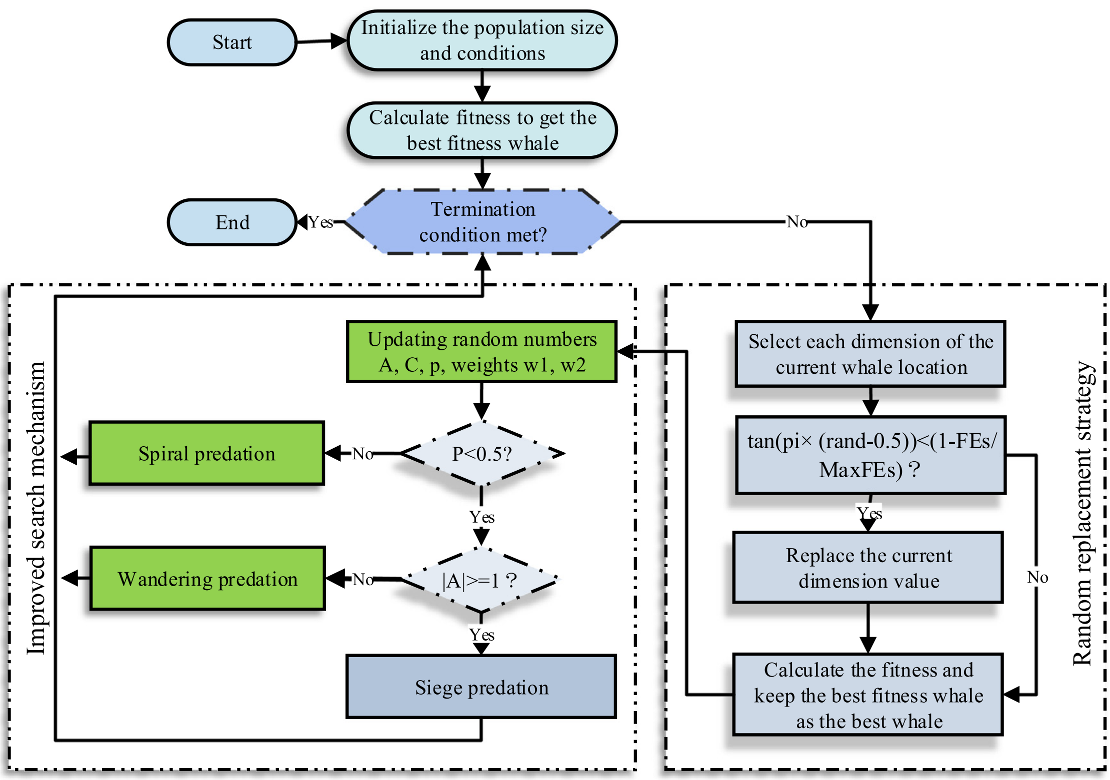

Fig. 1. flowchart of RDWOA.

2010), GWO (Heidari & Pahlavani, 2017), GSA (Rashedi et al., 2009), SCA (Mirjalili, 2016), FA (Yang, 2009) and MFO (Mirjalili et al., 2016).

In the experiment, the number of function evaluations and population size were set to 300,000 and 30 respectively, and the dimension of search space was set to 30. The comparison results are illustrated in Table 5. The average and standard deviation of each algorithm performed 30 times independently on 29 functions are listed. Wilcoxon sign rank test (García, Fernández, Luengo, & Herrera, 2010) is also used to evaluate the merits and demerits of RDWOA over other algorithms on 29 functions. The rows in Table 6 "+/=/−" show that RDWOA algorithm is significantly superior to other algorithms in most functions, and then Friedman test (Derrac, García, Molina, & Herrera, 2011) is also used to rank all the involved algorithms. The ARV of RDWOA is 1.172413, which is much lower than that of other algorithms. It suggests that the proposed RDWOA has performed the best among all the involved algorithms.

The convergence rate is an essential criterion for studying the effectiveness of evolutionary algorithms and exploring their development capabilities. To quickly and clearly understand the effectiveness and search trend of RDWOA, F3, F4, F6, F9, F10, F11, F13, F19, and F24 are selected as the convergence curves of the algorithm and recorded in Fig. 2. It can be seen that in all cases, the convergence speed of RDWOA is faster than that of other competitors, and the convergence accuracy is also higher than that of other competitors.

Fig. 2 shows the convergence curve of RDWOA and other original methods for different 30-dimensional benchmark problems. According to the convergence curve shown in Fig. 2, we can see that RDWOA has reasonable convergence rate in F3, F4, F9, and F10 test functions, while GWO is the second-best optimization algorithm, and other algorithms fall into local optimum early. In the test functions, RDWOA has better exploratory and exploitative abilities than other original optimization algorithms. It can make a more stable efficacy by balancing these core trends during iterations. The main reason lies in that random spare strategy, and double adaptive weight strategy has aided the WOA to find the global optimum quickly.

4.5. Comparison with state-of-the-art algorithms

In this section, RDWOA is further compared with some advanced algorithms, including LWOA (Ling et al., 2017), IWOA (Tubishat et al., 2018), OBSCA (Abd Elaziz et al., 2017), LGMSFOA (Shan et al., 2013), and CDLOBA (Yong et al., 2018). All the parameters are set as the same as those in the original study.

Table 7 lists the comparison results between RDWOA and advanced algorithms. As can be seen from Table 7, RDWOA is superior to its most advanced competitors in these benchmark functions.

Table 7 shows the mean and standard deviation (STD) values of different algorithms tested 30 times independently. In order to further derive the advantages of each algorithm intuitively, Friedman test (Derrac et al., 2011) is used to rank all the involved algorithms, as shown in Table 8.

As can be seen from Table 7, RDWOA has a minimum mean and STD value for 29 test functions. Therefore, the overall performance of RDWOA is the best among all the competitors. According to the statistics of the Wilcoxon test in Table 7, RDWOA can significantly outperform other competitors. According to "+/−/=", RDWOA is superior to IWOA, LWOA, OBSCA, LGMFOA, and CDLOBA for all 29 test functions. According to the average ranking value of Friedman test (Derrac et al., 2011) described in Table 8, the algorithms used are sorted in the following order: RDWOA, IWOA, LWOA, OBSCA, LGMFOA, and CDLOBA, which also shows that RDWOA is better than other advanced al-

Please cite this article as: H. Chen, C. Yang and A.A. Heidari et al., An efficient double adaptive random spare reinforced whale optimization algorithm, Expert Systems With Applications, https://doi.org/10.1016/j.eswa.2019.113018

Table 1
Description of the 29 benchmark functions.

ID	Function equation	Search range	Optimum value
Unimodal functions			
F1	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[−100,100]	$f_1\{X_{min}\} = 0$
F2	$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	[−10,10]	$f_2\{X_{min}\} = 0$
F3	$f_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	[−100,100]	$f_3\{X_{min}\} = 0$
F4	$f_4(x) = max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	[−100,100]	$f_4\{X_{min}\} = 0$
F5	$f_5(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	[−1.28,1.28]	$f_5\{X_{min}\} = 0$
Multimodal functions			
F6	$f_6(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert})$	[−500,500]	$f_6\{X_{min}\} = -418.9829 \times 5$
F7	$f_7(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	[−5.12,5.12]	$f_7\{X_{min}\} = 0$
F8	$f_8(x) = -20\exp\{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i}\} - exp\{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\} + 20 + e$	[−32,32]	$f_8\{X_{min}\} = 0$
F9	$f_9(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	[−600,600]	$f_9\{X_{min}\} = 0$
F10	$f_{10}(x) = \frac{\pi}{n}\{10\sin(ay_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2 + \sum_{i=1}^{n}\mu(x_i, 10, 100, 4)\}$	[−50,50]	$f_{10}\{X_{min}\} = 0$
F11	$f_{11}(x) = 0.1\{sin^2(3\pi x_i) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)] + \sum_{i=1}^{n}\mu(x_i, 5, 100, 4)$	[−50,50]	$f_{11}\{X_{min}\} = 0$
Simple multimodal functions			
F12	$f_{12}(x) = (\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i-1}^{2}(x_i - a_{ij})^6})$	[−65,65]	$f_{12}\{X_{min}\} = 1$
F13	$f_{13}(x) = \sum_{i=11}^{11}[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]$	[−5,5]	$f_{13}\{X_{min}\} = 0.00030$
F14	$f_{14}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[−2,2]	$f_{14}\{X_{min}\} = 3$
F15	$f_{15}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	$f_{15}\{X_{min}\} = -10.1532$
F16	$f_{16}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	$f_{16}\{X_{min}\} = -10.4028$
F17	$f_{17}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0,10]	$f_{17}\{X_{min}\} = -10.5363$
CEC 2014 simple multimodal functions			
F18	Shifted and Rotated Ackley's Function	[−100,100]	$f_{18}\{X_{min}\} = 500$
F19	Shifted Schwefel's Function	[−100,100]	$f_{19}\{X_{min}\} = 1000$
F20	Shifted and Rotated Schwefel's Function	[−100,100]	$f_{20}\{X_{min}\} = 1100$
F21	Shifted and Rotated Katsuura Function	[−100,100]	$f_{21}\{X_{min}\} = 1200$
F22	Shifted and Rotated HGBat Function	[−100,100]	$f_{22}\{X_{min}\} = 1400$
F23	Shifted and Rotated Expanded Scaffer's F6 Function	[−100,100]	$f_{23}\{X_{min}\} = 1600$
CEC 2014 composition functions			
F24	Composition Function 1 ($N = 5$)	[−100,100]	$f_{24}\{X_{min}\} = 2300$
F25	Composition Function 2 ($N = 3$)	[−100,100]	$f_{25}\{X_{min}\} = 2400$
F26	Composition Function 3 ($N = 3$)	[−100,100]	$f_{26}\{X_{min}\} = 2500$
F27	Composition Function 4 ($N = 5$)	[−100,100]	$f_{27}\{X_{min}\} = 2600$
F28	Composition Function 5 ($N = 5$)	[−100,100]	$f_{28}\{X_{min}\} = 2700$
F29	Composition Function 6 ($N = 5$)	[−100,100]	$f_{29}\{X_{min}\} = 2800$

Table 2
Outcomes of RDWOA with altered m values.

Function	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ARV	4.0301	4.1925	4.5047	4.87174	5.4264	5.3453	6.2491	6.3889	6.7642	7.2274
Rank	1	2	3	4	6	5	7	8	9	10

Table 3
Various WOAs with three strategies.

	RWOA	DWOA
WOA	0	0
RWOA	1	0
DWOA	0	1
RDWOA	1	1

gorithms. Fig. 3 shows the convergence curve of these advanced algorithms.

As shown in Fig. 3, RDWOA has the fastest convergence rate on all benchmark functions. For F11, although IWOA converges to the result faster than other algorithms in the early stage, the improved double adaptive weight strategy makes RDWOA perform more precise in local search, so the performance of the reinforced RDWOA is still the best on the whole. As shown in the test function plot, we can see that RDWOA is far superior to other competitors in these selected test functions, and other algorithms fall into local optimum prematurely.

In short, from the above experimental results and analysis, RDOWA is the best choice compared with other comparison algorithms, regardless of whether the terrain of the area is single modal, multimodal. The main reason is that the proposed RDOWA adopts random spare strategy and double adaptive weight strategy, which effectively improves the overall exploration ability and exploitation capability of the original WOA.

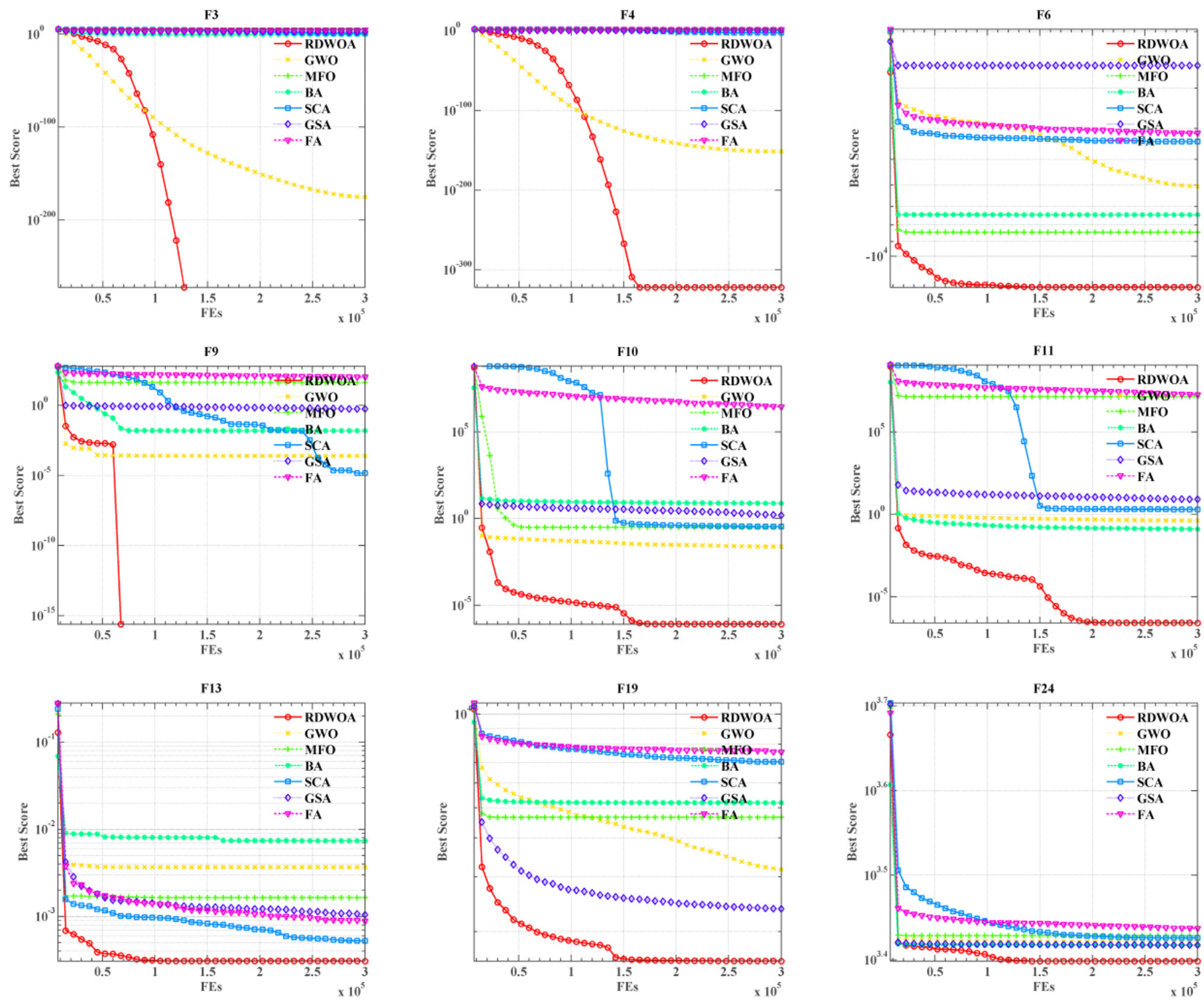

Fig. 2. Convergence trends of RDWOA and original algorithms for some selected problems.

4.6. High-dimensional experiment

Eleven test functions (F1–F11) are used to study the performance of RDWOA in the high dimensional environment and compare RDWOA and original WOA with different dimensions, including 100, 200, 500, 1000, and 2000. The comparison results are shown in Table 9, which lists the average, and standard deviation of each algorithm performed 30 times independently in five dimensions on 11 functions. Wilcoxon sign rank test (García et al., 2010) is also used to evaluate the advantages and disadvantages of RDWOA on 11 functions. The rows in Table 10 "+/=/−" show that RDWOA is superior to the original algorithm in most dimensions, and then through Friedman test (Derrac et al., 2011) shows that the ARV of RDWOA is 1.545455 when the dimension is 100. Comparisons show that RDWOA is still better than the original WOA in the case of higher dimensions. It means the proposed RDWOA has good scalability for tackling the optimization tasks.

From Table 9, we can see that RDWOA not only significantly improves the performance of the original WOA, but also outperforms the original WOA in dealing with F1–F11 test functions in both low and high dimensions. Also, all STD values in Table 9 show that RDWOA deviates the least when the dimension is 100. That is to say, RDWOA is more stable when the dimension is 100 than other test dimensions and can find the best solution in a smaller range. Friedman test was used to explore the significant differences of RDWOA under different dimensions further. Based on the ranking results in Table 10, it is revealed that the proposed RDWOA with dimension=100 achieves the lowest average ranking for both low-dimensional and high-dimensional problems, indicating that RDWOA has the best performance with dimension=100 and is significantly superior to RDWOA in other dimensions.

From Fig. 4, we can see that the reinforced algorithm is better than the original algorithm in the high dimensions. Moreover, we find that in this high dimension test, most of the features appear as follows: the higher the dimension, the worse the search ability. However, in the test function of F6, we can find that the phenomenon is just the opposite. This function is a multi-modal function, and we conclude that the higher the dimension of the algorithm, the stronger the performance of the algorithm in dealing with some multi-modal functions.

As we know, it is particularly noteworthy that providing solutions becomes more complex and challenging as the dimensions of test functionality continue to grow. Through the above analysis, we can conclude that RDWOA is a better choice to find the optimal value of high-dimensional functions than original WOA.

Please cite this article as: H. Chen, C. Yang and A.A. Heidari et al., An efficient double adaptive random spare reinforced whale optimization algorithm, Expert Systems With Applications, https://doi.org/10.1016/j.eswa.2019.113018

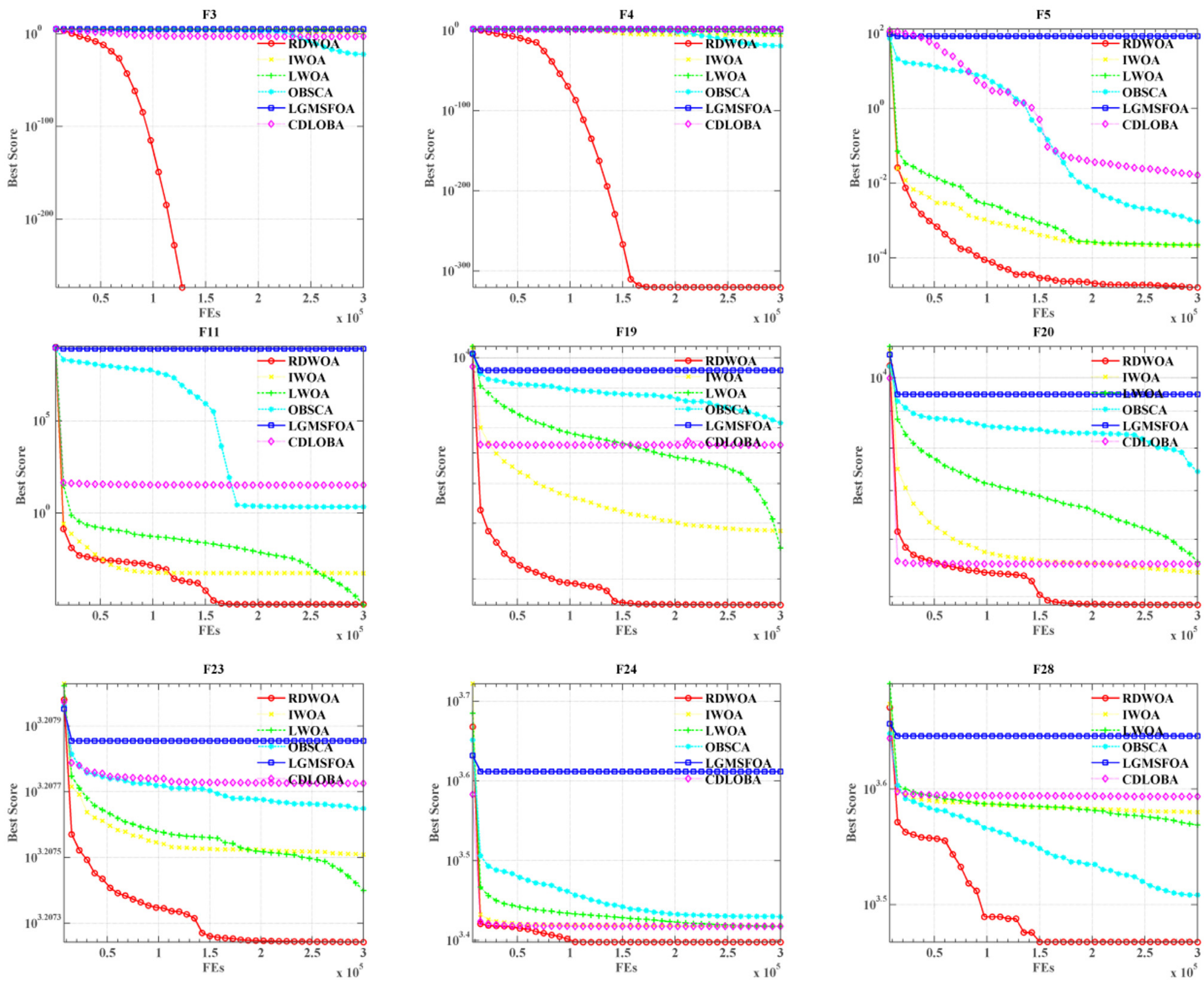

Fig. 3. Convergence trends of RDWOA and advanced algorithms for some selected problems .

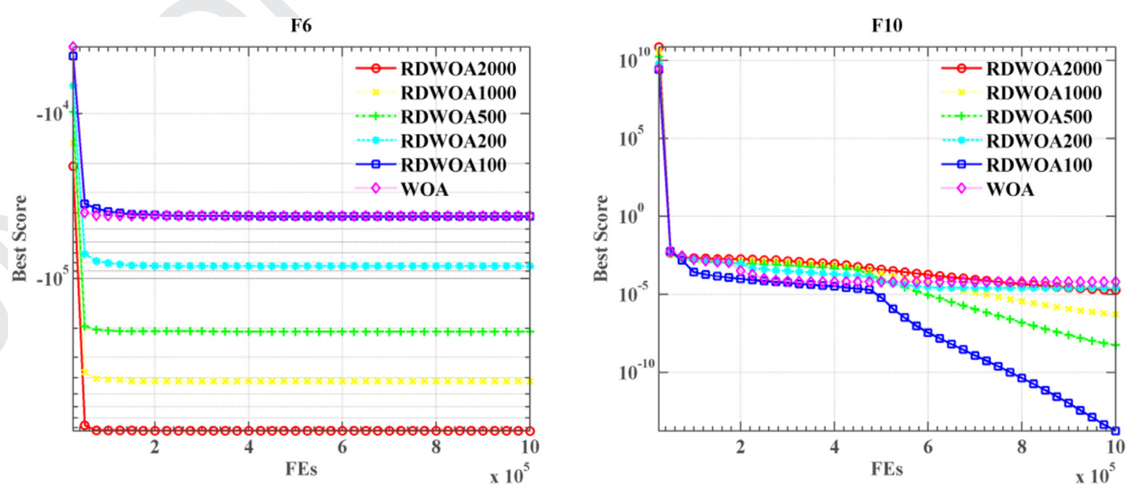

Fig. 4. Convergence trends of existing problems in different dimensions.

5. RDWOA for the engineering benchmarks

Many cases in industry and engineering deal with algorithms and mathematical models (Gao, Guirao, Basavanagoud, & Wu, 2018, Gao, Wang, Dimitrov, & Wang, 2018; Gao, Wu, Siddiqui, & Baig, 2018, 2019; Wei, Darko, & Hosam, 2018). In this section, RDWOA is applied to three engineering benchmarks: pressure vessel design (PVD), welded beam design (WBD), and tension/compression spring design (TCSD). Also, the obtained optimal solution has many limits that should not be violated.

Please cite this article as: H. Chen, C. Yang and A.A. Heidari et al., An efficient double adaptive random spare reinforced whale optimization algorithm, Expert Systems With Applications, https://doi.org/10.1016/j.eswa.2019.113018

Table 4
P-values of Wilcoxon test between RDWOA and other WOA variants.

Function	RDWOA	RWOA	DWOA	WOA
F1	N/A	1.00E+00	1.00E+00	1.00E+00
F2	N/A	1.00E+00	1.00E+00	1.00E+00
F3	N/A	1.73E−06	1.73E−06	1.73E−06
F4	N/A	1.73E−06	1.73E−06	1.73E−06
F5	N/A	4.29E−06	1.73E−06	1.13E−05
F6	N/A	6.34E−06	4.73E−06	1.73E−06
F7	N/A	1.00E+00	1.00E+00	1.00E+00
F8	N/A	5.79E−05	5.79E−05	9.77E−04
F9	N/A	2.50E−01	2.50E−01	1.00E+00
F10	N/A	1.73E−06	1.73E−06	1.92E−06
F11	N/A	3.11E−05	1.73E−06	2.84E−05
F12	N/A	1.00E+00	1.73E−06	3.11E−05
F13	N/A	1.20E−03	1.48E−02	7.73E−03
F14	N/A	9.23E−02	2.56E−06	1.97E−05
F15	N/A	5.04E−04	1.73E−06	2.13E−06
F16	N/A	5.10E−04	1.73E−06	1.73E−06
F17	N/A	2.55E−06	3.11E−05	1.73E−06
F18	N/A	1.04E−03	1.73E−06	1.73E−06
F19	N/A	1.32E−02	1.73E−06	6.04E−03
F20	N/A	7.50E−01	1.73E−06	9.32E−06
F21	N/A	7.66E−01	1.73E−06	2.21E−01
F22	N/A	1.89E−04	1.15E−04	4.45E−05
F23	N/A	7.73E−03	2.41E−04	6.88E−01
F24	N/A	1.73E−06	1.92E−06	1.92E−06
F25	N/A	1.73E−06	1.73E−06	1.73E−06
F26	N/A	1.73E−06	1.73E−06	2.93E−04
F27	N/A	6.29E−01	9.92E−01	3.60E−01
F28	N/A	1.73E−06	2.16E−05	8.47E−06
F29	N/A	1.73E−06	1.80E−05	2.88E−06
+/−/=	N/A	16/4/9	13/3/13	21/2/6
ARV	**1.586207**	2.586207	2.37931	2.758621

5.1. PVD problem

The purpose of this mathematical model is to minimize the total cost of cylindrical PV, which is closely related to material, structure, and welding. The end of the PV is covered, and the front has a hemispherical figure. In PVD problem, the thickness of the shell (T_s) and head (T_h), inner radius (r) and the range of cross-section minus head (l) are all variables to be optimized. The method can be described as follows.

$$
\begin{aligned}
&\text{Consider} && \vec{x} = [x_1\ x_2\ x_3\ x_4] = [T_s\ T_h\ R\ L] \\
&\text{Objective:} && f(\vec{x})_{\min} = 0.6224x_1x_3x_4 + 1.7781x_3x_1^2 + \\
& && 3.1661x_4x_1^2 + 19.84x_3x_1^2 \\
&\text{Subject to} && g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0 \\
& && g_2(\vec{x}) = -x_3 + 0.00954x_3 \le 0, \\
& && g_3(\vec{x}) = -\pi x_4 x_3^2 - \tfrac{4}{3}\pi x_3^3 + 1,296,000 \le 0, \\
& && g_4(\vec{x}) = x_4 - 240 \le 0, \\
&\text{Variable ranges:} && \\
& && 0 \le x_1 \le 99, \\
& && 0 \le x_2 \le 99, \\
& && 10 \le x_3 \le 200, \\
& && 10 \le x_4 \le 200.
\end{aligned}
$$

Many meta-heuristic methods have tried this mathematical model. He et al. [45] proposed PSO to solve this problem. The optimal cost was 6061.0777. Deb et al. [46] use GA to try this optimization problem. The optimal cost of 6410.3811 is realized. Also, ES [47], IHS [48], and mathematical methods [44, 49] are all used to handle this task.

When calculating pressure design problems, the optimization results of the algorithms in the literature are shown in Table 11. As can be seen from Table 11, the minimum value of RDWOA is 5912.53868, which means that when T_S, T_h, R and L are set to 0.793769, 0.39236, 41.127973, and 189.045124, the total cost of cylindrical PV is the smallest. Among all these algorithms, RDWOA

Table 5
Comparison of results for different algorithms.

	F1		F2		F3	
	mean	STD	mean	STD	mean	STD
RDWOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
GWO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.00E−176	0.00E+00
MFO	1.00E+03	3.05E+03	3.47E+01	2.16E+01	1.56E+04	1.16E+04
BA	6.49E−01	3.16E−01	3.64E+00	1.85E+00	2.54E−01	1.45E−01
SCA	2.27E−53	1.24E−52	1.28E−57	7.01E−57	7.61E+00	1.68E+01
GSA	1.50E+01	1.62E+00	1.58E+01	9.81E−01	3.15E+02	7.15E+01
FA	1.12E+04	1.39E+03	4.84E+01	3.71E+00	2.01E+04	2.32E+03
	F4		F5		F6	
	mean	STD	mean	STD	mean	STD
RDWOA	0.00E+00	0.00E+00	1.30E−05	1.23E−05	−1.25E+04	1.80E+02
GWO	2.60E−152	1.32E−151	7.32E−05	5.54E−05	−6.07E+03	6.71E+02
MFO	6.48E+01	1.03E+01	4.88E+00	7.96E+00	−8.42E+03	9.49E+02
BA	5.07E+00	4.41E+00	1.13E+01	7.36E+00	−7.43E+03	8.42E+02
SCA	3.63E−03	1.50E−02	2.68E−03	3.60E−03	−4.40E+03	1.96E+02
GSA	1.78E+00	1.18E−01	3.08E+01	6.02E+00	−2.55E+03	3.94E+02
FA	4.03E+01	2.67E+00	3.80E+00	8.40E−01	−4.14E+03	1.94E+02
	F7		F8		F9	
	mean	STD	mean	STD	mean	STD
RDWOA	0.00E+00	0.00E+00	8.88E−16	0.00E+00	0.00E+00	0.00E+00
GWO	0.00E+00	0.00E+00	7.64E−15	1.08E−15	2.51E−04	1.37E−03
MFO	1.58E+02	3.47E+01	1.43E+01	8.09E+00	4.22E+01	6.15E+01
BA	2.47E+02	1.94E+01	2.88E+00	3.30E+00	1.58E−02	1.69E−02
SCA	4.54E−03	2.49E−02	1.13E+01	9.35E+00	1.48E−05	8.10E−05
GSA	1.99E+02	7.56E+00	4.18E+00	1.60E−01	5.49E−01	4.12E−02
FA	2.33E+02	1.16E+01	1.60E+01	3.66E−01	1.03E+02	1.35E+01
	F10		F11		F12	
	mean	STD	mean	STD	mean	STD
RDWOA	8.16E−07	3.15E−06	2.60E−07	1.42E−06	9.98E−01	2.08E−16
GWO	2.39E−02	1.20E−02	4.24E−01	2.17E−01	5.23E+00	4.64E+00
MFO	3.12E−01	4.58E−01	1.37E+07	7.49E+07	2.12E+00	1.69E+00
BA	7.55E+00	3.36E+00	1.27E−01	7.08E−02	3.16E+00	2.85E+00
SCA	3.46E−01	6.80E−02	1.99E+00	9.43E−02	9.98E−01	4.36E−07
GSA	1.55E+00	3.19E−01	8.07E+00	1.34E+00	9.98E−01	2.56E−05
FA	2.72E+06	1.00E+06	1.77E+07	4.10E+06	9.98E−01	2.13E−05
	F13		F14		F15	
	mean	STD	mean	STD	mean	STD
RDWOA	3.08E−04	1.24E−07	3.00E+00	1.01E−07	−1.02E+01	1.17E−04
GWO	3.68E−03	7.59E−03	3.00E+00	1.38E−07	−9.81E+00	1.29E+00
MFO	1.64E−03	3.57E−03	3.00E+00	1.85E−15	−6.98E+00	3.53E+00
BA	7.36E−03	9.35E−03	3.01E+00	1.01E−02	−7.54E+00	2.22E+00
SCA	5.26E−04	3.85E−04	3.00E+00	3.04E−07	−2.40E+00	2.49E+00
GSA	1.04E−03	2.27E−04	3.00E+00	3.33E−03	−6.19E+00	1.36E+00
FA	8.82E−04	1.75E−04	3.00E+00	9.41E−04	−7.93E+00	9.39E−01
	F16		F17		F18	
	mean	STD	mean	STD	mean	STD
RDWOA	−1.04E+01	2.30E−05	−1.05E+01	3.13E−15	5.20E+02	1.78E−01
GWO	−1.02E+01	9.70E−01	−1.04E+01	9.87E−01	5.21E+02	5.82E−02
MFO	−6.77E+00	3.50E+00	−8.76E+00	3.03E+00	5.20E+02	1.83E−01
BA	−8.82E+00	1.86E+00	−9.08E+00	1.73E+00	5.21E+02	3.25E−02
SCA	−4.93E+00	2.93E+00	−4.74E+00	2.46E+00	5.21E+02	6.16E−02
GSA	−6.61E+00	1.38E+00	−6.58E+00	1.45E+00	5.21E+02	5.32E−02
FA	−8.45E+00	6.92E−01	−8.67E+00	8.12E−01	5.21E+02	5.78E−02
	F19		F20		F21	
	mean	STD	mean	STD	mean	STD
RDWOA	1.61E+03	2.64E+02	4.51E+03	6.05E+02	1.20E+03	1.83E−01
GWO	3.17E+03	5.87E+02	3.71E+03	6.06E+02	1.20E+03	1.13E+00
MFO	4.66E+03	9.56E+02	5.14E+03	7.32E+02	1.20E+03	2.84E−01
BA	5.19E+03	7.29E+02	5.26E+03	7.93E+02	1.20E+03	3.75E−01
SCA	7.02E+03	5.68E+02	8.16E+03	2.53E+02	1.20E+03	3.15E−01
GSA	2.37E+03	2.45E+02	2.88E+03	2.95E+02	1.20E+03	1.02E−01
FA	7.55E+03	2.73E+02	7.97E+03	3.05E+02	1.20E+03	2.36E−01
	F22		F23		F24	
	mean	STD	mean	STD	mean	STD
RDWOA	1.40E+03	1.21E−01	1.61E+03	5.54E−01	2.50E+03	2.13E+01
GWO	1.40E+03	5.23E+00	1.61E+03	6.14E−01	2.64E+03	1.04E+01
MFO	1.43E+03	1.67E+01	1.61E+03	4.81E−01	2.68E+03	4.82E+01
BA	1.40E+03	1.14E−01	1.61E+03	3.94E−01	2.62E+03	2.21E−03
SCA	1.45E+03	9.08E+00	1.61E+03	2.13E−01	2.67E+03	1.29E+01
GSA	1.40E+03	4.48E−02	1.61E+03	3.50E−01	2.62E+03	4.28E+00
FA	1.44E+03	4.61E+00	1.61E+03	1.78E−01	2.74E+03	1.47E+01

(*continued on next page*)

Table 5 (*continued*)

	F25		F26		F27	
	mean	STD	mean	STD	mean	STD
RDWOA	2.60E+03	1.49E−04	2.70E+03	0.00E+00	2.70E+03	1.26E−01
GWO	2.60E+03	4.43E−04	2.71E+03	5.27E+00	2.75E+03	5.07E+01
MFO	2.68E+03	3.41E+01	2.72E+03	8.79E+00	2.70E+03	1.36E+00
BA	2.67E+03	3.09E+01	2.73E+03	1.81E+01	2.70E+03	1.32E−01
SCA	2.60E+03	4.45E−02	2.72E+03	1.03E+01	2.70E+03	6.65E−01
GSA	2.61E+03	4.42E−01	2.70E+03	1.10E−01	2.77E+03	4.58E+01
FA	2.71E+03	4.76E+00	2.73E+03	3.93E+00	2.70E+03	3.84E−01

	F28		F29	
	mean	STD	mean	STD
RDWOA	2.90E+03	0.00E+00	3.03E+03	1.46E+02
GWO	3.38E+03	9.17E+01	3.94E+03	3.13E+02
MFO	3.68E+03	1.80E+02	3.96E+03	2.71E+02
BA	3.82E+03	4.55E+02	5.38E+03	8.57E+02
SCA	3.48E+03	3.30E+02	4.84E+03	2.92E+02
GSA	3.19E+03	1.22E+02	4.68E+03	3.06E+02
FA	3.80E+03	2.89E+01	4.19E+03	4.46E+01

Table 6
Comparison of results for different algorithms.

	RDWO A	GWO	MFO	BA	SCA	GSA	FA
+/=/−		23/2/4	27/0/2	28/0/1	27/0/2	28/1/0	29/0/0
ARV	1.1724	2.9310	4.4828	4.7241	4.4138	4.5862	5.5862

can find the best feasible optimization design. Therefore, RDWOA can provide powerful help for PVD problem.

5.2. WBD problem

The objective of the WBD problem (Coello, 2000) is to obtain the minimum manufacturing cost of WBD. In this case, takes shear stress (τ), bending stress (θ), buckling load (P_c), deflection (δ) as the constraints (Heidari, Abbaspour, & Jordehi, 2017). There are four variables and seven constraints for WBD problem: x_1 is the thickness (h); x_2 represents the length of the clamped bar (l); x_3 is the height t (t); x_4 is the thickness (b). The optimization model of WBD case is expressed as follows:

Consider $\vec{x} = [x_1\ x_2\ x_3\ x_4] = [h\ l\ t\ b]$

Minimize $f(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_4)$

Subject to
$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \le 0$$
$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \le 0$$
$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \le 0$$
$$g_4(\vec{x}) = x_1 - x_4 \le 0$$
$$g_5(\vec{x}) = P - P_C(\vec{x}) \le 0$$
$$g_6(\vec{x}) = 0.125 - x_1 \le 0$$
$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0$$

Variable range: $0.1 \le x_1 \le 2, 0.1 \le x_2 \le 10, 0.1 \le x_3 \le 10, 0.1 \le x_4 \le 2$

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

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$$
$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$$
$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3{}^2}, \delta(\vec{x}) = \frac{6PL^3}{Ex_3^2x_4}$$
$$P_C(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$
$$P = 6000lb, L = 14in..\delta_{\max} = 0.25in..$$
$$E = 30 \times 1^6 psi, G = 12 \times 10^6 psi$$
$$\tau_{\max} = 13600psi, \sigma_{\max} = 30000psi$$

Table 7
Comparison of results for different algorithms.

Fun	F1		F2		F3	
Method	mean	STD	mean	STD	mean	STD
RDWOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
IWOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.53E+01	1.56E+02
LWOA	3.72E−64	1.66E−63	1.89E−195	0.00E+00	2.30E+02	2.83E+02
OBSCA	2.16E−108	9.57E−108	1.31E−88	5.81E−88	8.23E−23	2.35E−22
LGMSFO	5.48E+04	6.68E+03	1.18E+02	8.93E+00	9.04E+04	2.14E+04
A CDLOBA	2.03E−04	5.99E−05	6.36E+00	2.59E+01	8.50E−04	3.90E−04
Fun	**F4**		**F5**		**F6**	
Method	mean	STD	mean	STD	mean	STD
RDWOA	0.00E+00	0.00E+00	1.61E−05	1.06E−05	−1.25E+04	1.39E+02
IWOA	2.48E−06	6.73E−06	2.16E−04	3.25E−04	−1.24E+04	6.97E+02
LWOA	1.78E−04	3.58E−04	2.20E−04	3.03E−04	−1.15E+04	1.07E+03
OBSCA	9.47E−20	4.23E−19	9.20E−04	5.95E−04	−4.11E+03	2.22E+02
LGMSFO	7.89E+01	4.48E+00	8.53E+01	2.64E+01	−3.54E+03	7.59E+02
A CDLOBA	3.98E+01	8.73E+00	1.63E−02	6.76E−03	−7.37E+03	5.99E+02
Fun	**F7**		**F8**		**F9**	
Method	mean	STD	mean	STD	mean	STD
RDWOA	0.00E+00	0.00E+00	8.88E−16	0.00E+00	0.00E+00	0.00E+00
IWOA	0.00E+00	0.00E+00	1.95E−15	1.67E−15	0.00E+00	0.00E+00
LWOA	0.00E+00	0.00E+00	8.17E−15	2.70E−15	0.00E+00	0.00E+00
OBSCA	0.00E+00	0.00E+00	4.44E−15	0.00E+00	0.00E+00	0.00E+00
LGMSFO	3.69E+02	2.97E+01	2.01E+01	4.26E−01	5.05E+02	5.97E+01
A CDLOBA	1.84E+02	6.82E+01	1.97E+01	6.19E−01	1.09E+02	6.63E+01
Fun	**F10**		**F11**		**F12**	
Method	mean	STD	mean	STD	mean	STD
RDWOA	6.09E−07	2.70E−06	1.15E−05	5.13E−05	9.98E−01	1.96E−12
IWOA	5.37E−07	1.85E−07	5.57E−04	2.46E−03	9.98E−01	2.89E−15
LWOA	9.67E−07	3.02E−07	1.08E−05	3.48E−06	9.98E−01	3.87E−14
OBSCA	3.64E−01	4.57E−02	2.16E+00	1.21E−01	1.10E+00	4.44E−01
LGMSFO	3.75E+08	1.15E+08	8.07E+08	1.65E+08	1.23E+01	7.96E+00
A CDLOBA	1.76E+01	7.90E+00	3.29E+01	1.20E+01	1.79E+00	1.09E+00
Fun	**F13**		**F14**		**F15**	
Method	mean	STD	mean	STD	mean	STD
RDWOA	3.07E−04	1.09E−08	3.00E+00	3.10E−07	−1.02E+01	2.90E−05
IWOA	3.54E−04	2.05E−04	3.00E+00	1.87E−09	−1.02E+01	7.92E−07
LWOA	4.45E−04	3.35E−04	3.00E+00	6.64E−08	−7.65E+00	2.98E+00
OBSCA	5.57E−04	1.43E−04	3.00E+00	2.54E−06	−9.69E+00	2.57E−01
LGMSFO	3.71E−02	4.15E−02	1.85E+01	2.94E+01	−1.05E+00	7.75E−01
A CDLOBA	4.68E−03	8.05E−03	3.00E+00	1.71E−03	−5.73E+00	3.39E+00
Fun	**F16**		**F17**		**F18**	
Method	mean	STD	mean	STD	mean	STD
RDWOA	−1.04E+01	3.26E−15	−1.05E+01	3.35E−05	5.20E+02	1.24E−01
IWOA	−1.04E+01	2.85E−07	−1.05E+01	1.29E−07	5.20E+02	8.27E−02
LWOA	−8.60E+00	2.96E+00	−9.73E+00	1.98E+00	5.20E+02	1.36E−01
OBSCA	−1.00E+01	2.23E−01	−9.94E+00	3.35E−01	5.21E+02	5.62E−02
LGMSFO	−1.84E+00	2.52E+00	−1.74E+00	9.83E−01	5.21E+02	8.92E−02
A CDLOBA	−6.36E+00	3.78E+00	−7.43E+00	3.56E+00	5.21E+02	1.55E−01
Fun	**F19**		**F20**		**F21**	
Method	mean	STD	mean	STD	mean	STD
RDWOA	1.66E+03	2.88E+02	4.88E+03	5.58E+02	1.20E+03	1.20E+03
IWOA	2.83E+03	5.22E+02	5.40E+03	4.76E+02	1.20E+03	1.20E+03
LWOA	2.51E+03	5.58E+02	5.58E+03	6.70E+02	1.20E+03	1.20E+03
OBSCA	6.22E+03	3.87E+02	7.43E+03	4.23E+02	1.20E+03	1.20E+03
LGMSFO	9.12E+03	7.45E+02	9.49E+03	5.82E+02	1.20E+03	1.20E+03
A CDLOBA	5.29E+03	6.89E+02	5.55E+03	7.81E+02	1.20E+03	1.20E+03

(*continued on next page*)

Please cite this article as: H. Chen, C. Yang and A.A. Heidari et al., An efficient double adaptive random spare reinforced whale optimization algorithm, Expert Systems With Applications, https://doi.org/10.1016/j.eswa.2019.113018

Table 7 (*continued*)

Fun	F22		F23		F24	
Method	mean	STD	mean	STD	mean	STD
RDWOA	1.40E+03	5.21E−02	1.61E+03	6.64E−01	2.50E+03	0.00E+00
IWOA	1.40E+03	6.00E−02	1.61E+03	4.85E−01	2.62E+03	3.98E+00
LWOA	1.40E+03	4.86E−02	1.61E+03	5.56E−01	2.62E+03	1.02E+00
OBSCA	1.47E+03	1.18E+01	1.61E+03	1.66E−01	2.69E+03	1.70E+01
LGMSFO	1.81E+03	5.24E+01	1.61E+03	3.61E−01	4.09E+03	5.00E+02
A	1.40E+03	1.40E−01	1.61E+03	2.73E−01	2.62E+03	6.20E−01
CDLOBA						
Fun	F25		F26		F27	
Method	mean	STD	mean	STD	mean	STD
RDWOA	2.60E+03	8.50E−05	2.70E+03	0.00E+00	2.71E+03	3.06E+01
IWOA	2.60E+03	1.26E+00	2.72E+03	1.63E+01	2.71E+03	2.23E+01
LWOA	2.60E+03	7.12E−01	2.71E+03	1.14E+01	2.70E+03	1.34E−01
OBSCA	2.60E+03	1.91E−04	2.70E+03	7.39E−05	2.70E+03	5.01E−01
LGMSFO	2.93E+03	5.14E+01	2.86E+03	5.27E+01	2.83E+03	1.20E+02
A	2.71E+03	4.20E+01	2.73E+03	1.45E+01	2.71E+03	3.06E+01
CDLOBA						
Fun	F28		F29			
Method	mean	STD	mean	STD		
RDWOA	2.93E+03	1.56E+02	3.11E+03	4.71E+02		
IWOA	3.80E+03	1.81E+02	4.70E+03	4.72E+02		
LWOA	3.70E+03	2.71E+02	4.13E+03	2.52E+02		
OBSCA	3.22E+03	2.95E+01	5.42E+03	4.75E+02		
LGMSFO	4.43E+03	2.37E+02	9.78E+03	1.23E+03		
A	3.92E+03	3.60E+02	5.49E+03	7.47E+02		
CDLOBA						

Table 8
Comparison of results for different algorithms.

	RDWOA	IWOA	LWOA	OBSCA	LGMSFOA	CDLOBA
+/=/−		17/2//10	22/2/5	25/1/3	29/0/0	27/0/2
ARV	1.403448	2.578448	3.035345	3.774138	5.910345	4.298276

Table 9
Results of RDWOA and WOA on F1-F11 with different dimensions.

Fun	F1		F2		F3	
Method	mean	STD	mean	STD	mean	STD
RDWOA200	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
RDWOA100	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
RDWOA500	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
RDWOA200	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.29E+03	5.31E+03
RDWOA100						
WOA						
Fun	F4		F5		F6	
Method	mean	STD	mean	STD	mean	STD
RDWOA200	0.00E+00	0.00E+00	5.12E−05	2.38E−04	−8.38E+05	1.22E−01
0	0.00E+00	0.00E+00	1.04E−05	1.60E−05	−4.19E+05	1.42E−01
RDWOA100	0.00E+00	0.00E+00	6.94E−06	4.48E−06	−2.09E+05	1.41E+02
0	0.00E+00	0.00E+00	1.13E−05	3.11E−05	−8.38E+04	2.50E−03
RDWOA500	0.00E+00	0.00E+00	1.00E−05	1.39E−05	−4.19E+04	1.29E−01
RDWOA200	5.27E+01	3.58E+01	5.06E−05	4.97E−05	−4.16E+04	7.50E+02
RDWOA100						
WOA						
Fun	F7		F8		F9	
Method	mean	STD	mean	STD	mean	STD
RDWOA200	0.00E+00	0.00E+00	8.88E−16	0.00E+00	0.00E+00	0.00E+00
0	0.00E+00	0.00E+00	8.88E−16	0.00E+00	0.00E+00	0.00E+00
RDWOA100	3.03E−14	1.66E−13	8.88E−16	0.00E+00	0.00E+00	0.00E+00
0	0.00E+00	0.00E+00	8.88E−16	0.00E+00	0.00E+00	0.00E+00
RDWOA500	0.00E+00	0.00E+00	8.88E−16	0.00E+00	0.00E+00	0.00E+00
RDWOA200	0.00E+00	0.00E+00	3.02E−15	1.77E−15	5.57E−04	2.12E−03
RDWOA100						
WOA						
Fun	F10		F11			
Method	mean	STD	mean	STD		
RDWOA200	1.94E−05	2.18E−05	8.20E−02	2.93E−01		
0	5.32E−07	3.54E−07	2.83E−02	1.26E−01		
RDWOA100	5.66E−09	2.04E−09	5.13E−03	5.58E−03		
0	2.64E−05	1.00E−04	6.07E−03	6.49E−03		
RDWOA500	1.78E−14	6.05E−15	4.79E−03	6.12E−03		
RDWOA200	6.22E−05	3.36E−04	2.99E−03	4.94E−03		
RDWOA100						
WOA						

This issue has recently attracted a widespread attention. Similarly, Kaveh et al. use RO (A Kaveh & M Khayatazad, 2012) to solve this problem. Lee et al. Kang and Zong (2005) use HS to optimize the problem. The experimental results show that HS can obtain the optimal cost of 2.3807. The improved HS (IHS) (Mahdavi, Fesanghary, & Damangir, 2007b) has also been tested. Mathematical methods Davidon–Fletcher–Powell, Richardson's random method and Simplex method have been adopted by Ragsdell and Phillips (1976).

The results of RDWOA are compared with other solutions previously worked in Table 12. As shown, RDWOA obtains the optimal cost of 1.6961. This observation shows that when the four parameters are set to 0.205618, 3.252958, 9.04447, and 0.20569, respectively, the manufacturing cost of WB can reach 1.6961. As can be seen from the table, RDWOA is superior to all other methods. In short, compared with other algorithms, the proposed RDWOA solves this problem effectively and obtains the minimum manufacturing cost.

5.3. TCSD problem

Another classic engineering problem is TCSD. In this case, the decision-maker wants to minimize the heaviness of a spring. The problem consists of three variables: diameter (d), mean coil diameter (D), and the number of dynamic coils (N) (Arora, 2004). The mathematical expression for TCSD is as follows.

Consider $\vec{x} = [x_1\ x_2\ x_3] = [d\ D\ N]$
Minimize $f(\vec{x}) = (x_3 + 2)x_2 x_1^2$

Subject to
$$g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{7{,}178{,}5x_1^4} \le 0$$
$$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{1{,}256{,}6(x_2 x_1^3 - x_1^4)} + \frac{1}{5180x_1^2} \le 0$$
$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^3 x_3} \le 0$$
$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

Variable range $0.05 \le x_1 \le 2.00$, $0.25 \le x_2 \le 1.30$, $2.00 \le x_3 \le 15.0$

We can use mathematical and meta-heuristic methods to solve the TCSD problem. Meta-heuristic methods include SSA, GSA, PSO, ES, GA, RO, improved HS, and DE. Numerical optimization techniques (fixed cost constraints correction) and mathematical optimization techniques are mathematical methods for solving TCSD. Table 13 shows the experimental results between the above techniques and RDWOA. Note that for a fair comparison, we apply a similar penalty function to RDWOA. Table 13 shows that RDWOA performs better than all other algorithms in dealing with TCSD problems and provides the most effective design scheme.

In this section, three classical engineering design problems are experimentally studied, and the results show that the RDWOA method can solve practical engineering design problems in a much better way. It also verified that proposed RDWOA could deal with the constraint optimization problems effectively.

Table 10
The Friedman test results of RDWOA and WOA on F1-F11 with different dimensions.

	RDWOA2000	RDWOA1000	RDWOA500	RDWOA200	RDWOA100	WOA
ARV	2.181818	1.818182	1.909091	2.181818	**1.545455**	4.090909
Rank 4	4	2	3	4	**1**	6

Table 11
Results of RDWOA versus peers in literature for PVD case.

Algorithm	Optimum variables				Optimum cost
	T_s	T_h	R	L	
RDWOA	0.793769	0.39236	41.127973	189.045124	5912.53868
IHS (Mahdavi et al., 2007b)	1.125000	0.625000	58.29015	43.69268	7197.7300
PSO (He & Wang, 2007)	0.812500	0.437500	42.091266	176.746500	6061.0777
GA (Deb, 1997)	0.937500	0.500000	48.329000	112.679000	6410.3811
ES (Mezura-Montes & Coello, 2008)	0.812500	0.437500	42.098087	176.640518	6059.7456
Lagrangian multiplier (Kannan & Kramer, 1994)	1.125000	0.625000	58.291000	43.690000	7198.0428
Branch-and-bound (Sandgren, 1990)	1.125000	0.625000	47.700000	117.71000	8129.1036

Table 12
Comparison results of RDWOA for the WBD problem.

Technique	Best variables				Best cost
	h	l	t	b	
RDWOA	0.205618	3.252958	9.04447	0.20569	1.6961
RO (Kaveh & M Khayatazad, 2012)	0.203687	3.528467	9.004233	0.207241	1.735344
HS (Kang & Zong, 2005)	0.2442	6.2231	8.2915	0.2433	2.3807
IHS (Mahdavi et al., 2007b)	0.20573	3.47049	9.03662	0.20573	1.7248
Random (Ragsdell & Phillips, 1976)	0.4575	4.7313	5.0853	0.6600	4.1185
Simple (Ragsdell & Phillips, 1976)	0.2792	5.6256	7.7512	0.2796	2.5307
David (Ragsdell & Phillips, 1976)	0.2434	6.2552	8.2915	0.2444	2.3841

Table 13
Comparison results of the TCSD problem.

Algorithm	Optimal values for variables			Optimum weight
	d	D	N	
RDWOA	0.0517112	0.35725	11.257788	0.0126652417
SSA	0.051207	0.345215	12.004032	0.0126763
GSA	0.050276	0.323680	13.525410	0.0127022
PSO (He & Wang, 2007)	0.015728	0.357644	11.244543	0.0126747
ES (Mezura-Montes & Coello, 2008)	0.051989	0.363965	10.890522	0.0126810
GA (Coello, 2000)	0.051480	0.351661	11.632201	0.0127048
RO (Kaveh & Khayatazad, 2012)	0.051370	0.349096	11.76279	0.0126788
Improved HS (Mahdavi, Fesanghary & Damangir, 2007a)	0.051154	0.349871	12.076432	0.0126706
DE (Li, Huang, Liu & Wu, 2007)	0.051609	0.354714	11.410831	0.0126702
Mathematical optimization	0.053396	0.399180	9.1854000	0.0127303
Constraint correction	0.050000	0.315900	14.250000	0.0128334

6. Conclusions and future works

In this paper, a new WOA variant, RDWOA, is proposed to alleviate the shortcomings of original WOA, such as inadequate optimization aptitude and slow convergence speed. In the proposed RDWOA, we introduce a random spare strategy, which enhances the search and convergence ability of the algorithm and makes it easier to escape from local optimum. On the other hand, the double adaptive weight strategy is introduced to enhance the global search ability and local search ability of the original WOA. Also, in order to further verify the impact of random spare strategy and double adaptive weight strategy strategies, different variants of WOA were constructed and compared. The experimental results show that the combination of both strategies with WOA, RDWOA, has achieved the best performance among all combinations. Comparing this algorithm with the original solver and five advanced algorithms, the numerical results of 29 representative test functions have shown that the developed reinforced RDWOA is significantly superior to other counterparts. The proposed RDWOA can not only find a high-quality solution effectively but also alleviate the premature convergence of original WOA. The comparison between the reinforced algorithm and the original WOA in high dimension is also tested, which verifies the effectiveness of the proposed RDWOA algorithm performing in the high dimensional environment. It can also jump out of the local optimum, which significantly improves the accuracy and diversity of WOA even faced with the high dimensional problems. Additionally, the simulation results show that RDWOA can improve the solution quality when solving three practical engineering design problems.

The research and application of WOA are still in its infancy, and there are still many problems to be further studied. First, the traditional meta-heuristic algorithm can be combined with WOA to balance the global and local search capabilities better and improve the overall optimization potential of WOA. Then, how to apply RDWOA to solve multi-objective problems and dynamic optimization is also the focus of our next research direction.

Declaration of competing interest

The authors declare that there is no conflict of interests regarding the publication of article.

Credit authorship contribution statement

Huiling Chen: Conceptualization, Methodology, Software, Writing - original draft, Investigation, Writing - review & editing. **Chenjun Yang:** Writing - review & editing, Software, Visualization. **Ali Asghar Heidari:** Writing - original draft, Writing - review & editing, Software, Visualization, Investigation. **Xuehua Zhao:** Conceptualization, Funding acquisition, Resources.

Acknowledgments

This research is supported by National Natural Science Foundation of China (U1809209), Science and Technology Plan Project of Wenzhou, China (ZG2017019), and Guangdong Natural Science Foundation (2018A030313339), MOE (Ministry of Education in China) Youth Fund Project of Humanities and Social Sciences (17YJCZH261), Scientific Research Team Project of Shenzhen Institute of Information Technology (SZIIT2019KJ022).

References

Abd Elaziz, M., Oliva, D., & Xiong, S. (2017). An improved opposition-based sine cosine algorithm for global optimization. *Expert Systems with Applications, 90*, 484–500.
Aljarah, I., Faris, H., & Mirjalili, S. (2016). Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, 1–15.
Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H., Zhang, Y., & Mirjalili, S. (2018). Asynchronous accelerating multi-leader salp chains for feature selection. *Applied Soft Computing, 71*, 964–979.
Arora, J. S. (2004). *Introduction to optimum design*.
Aziz, M. A. E., Ewees, A. A., & Hassanien, A. E. (2017). Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. *Expert Systems with Applications, 83*, 242–256.
Aziz, M. A. E., Ewees, A. A., & Hassanien, A. E. (2018). Multi-objective whale optimization algorithm for content-based image retrieval. *Multimedia Tools and Applications, 77*, 26135–26172.
Booker, L. B., Goldberg, D. E., & Holland, J. H. (1989). Classifier systems and genetic algorithms. *Artificial Intelligence, 40*, 235–282.
Cai, Z., Gu, J., Luo, J., Zhang, Q., Chen, H., Pan, Z., et al. (2019). Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy. *Expert Systems with Applications*. doi:10.1016/j.eswa.2019.07.031.
Chen, H., Jiao, S., Heidari, A. A., Wang, M., Chen, X., & Zhao, X. (2019). An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models. *Energy Conversion and Management, 195*, 927–942.
Coello, C. A. C. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry, 41*, 113–127.
Deb, K. (1997). *GeneAS: A robust optimal design technique for mechanical component design*. Berlin Heidelberg: Springer.
Deng, W., Xu, J., & Zhao, H. (2019). An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. *IEEE Access, 7*, 20281–20292.
Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*, 3–18.
Digalakis, J. G., & Margaritis, K. G. (2001). On benchmarking functions for genetic algorithms. *International Journal of Computer Mathematics, 77*, 481–506.
Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science, 344*, 243–278.
Faris, H., Ala'M, A.-. Z., Heidari, A. A., Aljarah, I., Mafarja, M., & Hassonah, M. A. (2019). An intelligent system for spam detection and identification of the most relevant features based on evolutionary Random Weight Networks. *Information Fusion, 48*, 67–83.
Faris, H., Heidari, A. A., Al-Zoubi, A. M., Mafarja, M., Aljarah, I., Eshtay, M., et al. (2019). Time-varying hierarchical chains of salps with random weight networks for feature selection. *Expert Systems with Applications, 140*, 112898.
Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Ala'M, A.-. Z., Mirjalili, S., et al. (2018). An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowledge-Based Systems, 154*, 43–67.
Gao, W., Guirao, J. L. G., Abdel-Aty, M., & Xi, W. (2019). An independent set degree condition for fractional critical deleted graphs. *Discrete and Continuous Dynamical Systems-Series S, 12*, 877–886.
Gao, W., Guirao, J. L. G., Basavanagoud, B., & Wu, J. (2018). Partial multi-dividing ontology learning algorithm. *Information Sciences, 467*, 35–58.
Gao, W., Wang, W., Dimitrov, D., & Wang, Y. (2018). Nano properties analysis via fourth multiplicative ABC indicator calculating. *Arabian journal of chemistry, 11*, 793–801.
Gao, W., Wu, H., Siddiqui, M. K., & Baig, A. Q. (2018). Study of biological networks using graph theory. *Saudi Journal of Biological Sciences, 25*, 1212–1219.
García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences, 180*, 2044–2064.
Hassan, G., & Hassanien, A. E. (2017). Retinal fundus vasculature multilevel segmentation using whale optimization algorithm. *Signal, Image and Video Processing*, 1–8.
He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence, 20*, 89–99.
Heidari, A. A., Abbaspour, R. A., & Jordehi, A. R. (2017). An efficient chaotic water cycle algorithm for optimization tasks. *Neural Computing and Applications, 28*, 57–85.
Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems, 97*, 849–872.
Heidari, A. A., & Pahlavani, P. (2017). An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. *Applied Soft Computing Journal, 60*, 115–134.
Kang, S. L., & Zong, W. G. (2005). A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics & Engineering, 194*, 3902–3933.
Kannan, B. K., & Kramer, S. N. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design, 116*, 405–411.
Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*, 459–471.
Kaveh, A., & Khayatazad, M. (2012). A new meta-heuristic method: Ray optimization. *Computers and Structures, 112-113*, 283–294.
Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *IEEE international conference on neural networks – Conference proceedings: 4* (pp. 1942–1948).
Khashan, N., A. Elhosseini, M., Y. Haikal, A., & Badawy, M. (2018). Biped robot stability based on an A-C parametric whale optimization algorithm. *Computational Science*.
Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.
Li, L. J., Huang, Z. B., Liu, F., & Wu, Q. H. (2007). A heuristic particle swarm optimizer for optimization of pin connected structures. *Computers & Structures, 85*, 340–349.
Li, X. L., Shao, Z. J., & Qian, J. X. (2002). Optimizing method based on autonomous animats: Fish-swarm algorithm. *Xitong Gongcheng Lilun yu Shijian/System Engineering Theory and Practice, 22*, 32.
Liang, J., Qu. , B., & Suganthan, P. (2014). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Computational Intelligence Laboratory*.
Ling, Y., Zhou, Y., & Luo, Q. (2017). Lévy flight trajectory-based whale optimization algorithm for global optimization. *IEEE Access, 5*, 6168–6186.
Mafarja, M., Aljarah, I., Heidari, A. A., Faris, H., Fournier-Viger, P., Li, X., et al. (2018). Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowledge-Based Systems, 161*, 185–204.
Mafarja, M., & Mirjalili, S. (2018). Whale optimization approaches for wrapper feature selection. *Applied Soft Computing Journal, 62*, 441–453.
Mahdavi, M., Fesanghary, M., & Damangir, E. (2007a). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188*, 1567–1579.
Mahdavi, M., Fesanghary, M., & Damangir, E. (2007b). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics & Computation, 188*, 1567–1579.
Meng, X., Liu, Y., Gao, X., & Zhang, H. (2014). A new bio-inspired algorithm: Chicken swarm optimization. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics): 8794* (pp. 86–94). Springer Verlag.
Mezura-Montes, E., & Coello, C. A. C. (2008). An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems, 37*, 443–473.
Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89*, 228–249.
Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120–133.
Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.
Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications, 27*, 495–513.
Molga M., S.C. (2005). Test functions for optimzation needs.
Ragsdell, K. M., & Phillips, D. T. (1976). Optimal design of a class of welded structures using geometric programming. *Journal of Engineering for Industry, 98*, 97.
Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences, 179*, 2232–2248.

Ratnaweera, A., Halgamuge, S. K., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation, 8*, 240–255.

Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design, 112*, 223–229.

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software, 105*, 30–47.

Shan, D., Cao, G., & Dong, H. (2013). LGMS-FOA: An improved fruit fly optimization algorithm for solving optimization problems. *Mathematical Problems in Engineering, 2013*.

Shen, L., Chen, H., Yu, Z., Kang, W., Zhang, B., Li, H., et al. (2016). Evolving support vector machines using fruit fly optimization for medical data classification. *Knowledge-Based Systems, 96*, 61–75.

Shi, Y., & Eberhart, R. C. (1999). In *Empirical study of particle swarm optimization: 3* (pp. 1945–1950). IEEE Computer Society.

Shi, Y., & Eberhart, R.C. (2001). Fuzzy adaptive particle swarm optimization. In (Vol. 1, pp. 101–106).

Sun, W. Z., & Wang, J. S. (2017). Elman neural network soft-sensor model of conversion velocity in polymerization process optimized by chaos whale optimization algorithm. *IEEE Access, 5*, 13062–13076.

Thanga Revathi, S., Ramaraj, N., & Chithra, S. (2018). Brain storm-based whale optimization algorithm for privacy-protected data publishing in cloud computing. *Cluster Computing*, 1–10.

Tubishat, M., Abushariah, M. A. M., Idris, N., & Aljarah, I. (2018). Improved whale optimization algorithm for feature selection in Arabic sentiment analysis. *Applied Intelligence*.

Wang, J., Du, P., Niu, T., & Yang, W. (2017). A novel hybrid system based on a new proposed algorithm—Multi-objective whale optimization algorithm for wind speed forecasting. *Applied Energy, 208*, 344–360.

Wei, G., Darko, D., & Hosam, A. (2018). Tight independent set neighborhood union condition for fractional critical deleted graphs and ID deleted graphs. *Discrete & Continuous Dynamical Systems, 12*, 711–721.

Xu, Y., Chen, H., Heidari, A. A., Luo, J., Zhang, Q., Zhao, X., et al. (2019). An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks. *Expert Systems with Applications, 129*, 135–155.

Xu, Y., Chen, H., Luo, J., Zhang, Q., Jiao, S., & Zhang, X. (2019). Enhanced moth-flame optimizer with mutation strategy for global optimization. *Information Sciences, 492*, 181–203.

Yang, X. S. (2009). Firefly algorithms for multimodal optimization. In *LNCS: 5792* (pp. 169–178).

Yang, X.-S. (2010). Firefly algorithm, stochastic test functions and design optimisation. 2*(2)*, 78–84.

Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. In *Studies in computational intelligence: 284* (pp. 65–74).

Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*, 82–102.

Yong, J., He, F., Li, H., & Zhou, W. (2018). A novel bat algorithm based on collaborative and dynamic learning of opposite population. In *Proceedings of the 2018 IEEE 22nd international conference on computer supported cooperative work in design, 8464759* (pp. 541–546).

Yousri, D., Allam, D., & Eteiba, M. B. (2019). Chaotic whale optimizer variants for parameters estimation of the chaotic behavior in permanent magnet synchronous motor. *Applied Soft Computing Journal, 74*, 479–503.

Yu, H., Zhao, N., Wang, P., Chen, H., & Li, C. (2019). Chaos-enhanced synchronized bat optimizer. *Applied Mathematical Modelling*. doi:10.1016/j.apm.2019.09.029.

Zhan, Z. H., Zhang, J., Li, Y., & Chung, H. S. H. (2009). Adaptive particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39*, 1362–1381.

Zhang, X., Hu, W., Qu, W., & Maybank, S. (2010). Multiple object tracking via species-based particle swarm optimization. *IEEE Transactions on Circuits & Systems for Video Technology, 20*, 1590–1602.

Zhang, X., Wang, D., Zhou, Z., & Ma, Y. (2019). Robust low-rank tensor recovery with rectification and alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1. doi:10.1109/TPAMI.2019.2929043.

Zhang, X., Xu, Y., Yu, C., Heidari, A. A., Li, S., Chen, H., et al. (2019). Gaussian mutational chaotic fruit fly-built optimization and feature selection. *Expert Systems with Applications*, 112976. doi:10.1016/j.eswa.2019.112976.

Zheng, Y., Li, Y., Wang, G., Chen, Y., Xu, Q., Fan, J., et al. (2018). A novel hybrid algorithm for feature selection based on whale optimization algorithm. *IEEE Access*.

Zhou, Y., Ling, Y., & Luo, Q. (2018). Lévy flight trajectory-based whale optimization algorithm for engineering optimization. *Engineering Computations, 35*, 2406–2428.