Neural Computing and Applications (2024) 36:533–568
https://doi.org/10.1007/s00521-023-09023-9

ORIGINAL ARTICLE

A novel reinforcement learning-based reptile search algorithm for solving optimization problems

Mohamed Ghetas[1] · **Mohamed Issa**[2,3]

Received: 26 March 2023 / Accepted: 5 September 2023 / Published online: 9 October 2023

Abstract

This work proposes a novel reptile search algorithm (RSA) to solve optimization problems called reinforcement reptile search algorithm (RLRSA). The basic RSA performs exploitation through highly walking in the first half of searching process while the exploration phase is executed through the hunting phase in the second half. Therefore, the algorithm is not able to balance exploration and exploitation and this behavior results in trapping in local optima. A novel learning method based on reinforcement learning and Q-learning model is proposed to balance the exploitation and exploration phases when the solution starts deteriorating. Furthermore, the random opposite-based learning (ROBL) is introduced to increase the diversity of the population and so enhance the obtained solutions. Twenty-three typical benchmark functions, including unimodal, multimodal and fixed-dimension multimodal functions, were employed to assess the performance of RLRSA. According to the findings, the RLRSA method surpasses the standard RSA approach in the majority of benchmark functions evaluated, specifically in 12 out of 13 unimodal functions, 9 out of 13 multimodal functions, and 8 out of 10 fixed multimodal functions. Furthermore, the RLRSA is applied to vessel solve pressure and tension/compression spring design problems. The results show that RLRSA significantly found the solution with minimum cost. The experimental results reveal the superiority of the RLRSA compared to RSA and other optimization methods in the literature.

Keywords Optimization algorithms · Reptile search algorithm (RSA) · Metaheuristics · Reinforcement learning

1 Introduction

Optimizations techniques categorized into two main categories are deterministic and stochastic methods. Deterministic techniques are partitioned into nonlinear and linear techniques [1]. The conventional methods utilize gradient learning to navigate through the search space with the goal of identifying the best possible solution [2, 3]. However, the deterministic techniques are very useful for unimodal search problems (linear), they are sensitive for trapping in local minima if they are implemented for multimodal search problems (nonlinear). This obstacle was solved by developing several techniques based on modification or hybridization of algorithm [4].

Stochastic methods employ randomized search parameters to explore the search space and identify a solution that is close to optimal [5]. The most common stochastic technique is metaheuristic techniques which have the advantage of easy implementation, simplicity, flexibility and independency to the problem [6–8]. Lately, metaheuristic algorithms were proposed to find solution for complex optimization problems [8, 9] [10]. Metaheuristic algorithm finds the near optimal solution with acceptable accuracy in a reasonable time.

Metaheuristic technique has two main merits that are diversification (exploration) and intensification (exploitation) [8]. Diversification search scheme explores the search space widely; however, intensification intensifies the searching process in a narrow region of the promising solutions to obtain the best solution. Metaheuristic

✉ Mohamed Ghetas
Mohamed.ghetas@gu.edu.eg

1 Faculty of Computer Science and Engineering, Galala University, Suez, Egypt

2 Computer and Systems Department, Faculty of Engineering, Zagazig University, Zagazig, Egypt

3 Faculty of Computer Science and Information Technology, Egypt-Japan University of Science and Technology, New Borg El Arab, Egypt

algorithms can be grouped into four essential classes [11]: (1) swarm-based intelligence (SI) [12], (2) human-based methods (HM) [13], (3) physics-based methods (PM) [14], and (4) evolutionary-based algorithms (EA) [15].

SI mimics the social behavior of some animal swarms such as herds, flocks and schools where communications between animals are the main characteristics and this behavior is emulated via optimization operation. The particle swarm optimization (PSO) algorithm is an example of a stochastic method within this category [16], whale optimization algorithm (WOA) [17], reptile search algorithm (RSA) [18], moth–flame optimization algorithm [19], Harris hawk optimization algorithm [20], salp optimization algorithm [21], bat optimization algorithm [22], and gray wolf optimization algorithm [23].

HMs mimic the behavior of human communication in communities. The most common HM algorithms are teaching learning-based optimization (TLBO) [24], brain storm optimization algorithm [25], human mental search [26], poor and rich optimization algorithm [27], simple human learning algorithm [28], and imperialist competitive algorithm (ICA) [29].

PMs' search strategy is mimicked by the physical and mathematical laws in life. The most common PM algorithms are simulated annealing (SA) [30], sine–cosine optimization algorithm (SCA) [31], gravitational search algorithm (GSA) [32], arithmetic optimization algorithm (AOA) [33], electromagnetic field optimization [34], ions motion optimization algorithm [35], and Henry gas solubility optimization [36]. EAs simulate the attitude of natural evolution where it uses the operators such as mutation and crossover which is inspired by biology. The most common EA algorithms are genetic algorithm (GA) [37], differential evolution (DE) [38], and evolution strategy (ES) [39].

The after-mentioned metaheuristic methods are being used to solve various optimization problems in different areas such as bioinformatics [40–46], photovoltaic design [47, 48], fuel cell design [49, 50], PID controller design [51–54], passive suspension system design [55, 56], combinatorial optimization problems [57], structural design [58], image watermarking [59, 60], image segmentation [61, 62], mining high-utility item sets [63], privacy-preserving data mining [64, 65], maximization occupancy of GPU [66, 67], human activity recognition [68], fuel cell [49, 69], exploiting GPU parallelism technique using MA [70], and others. While these algorithms have shown success in solving various optimization problems, it is widely acknowledged, as per the no-free-lunch theory of optimization, that no single algorithm can attain the optimal solution for all optimization problems [71].

The reptile search algorithm (RSA) is a recently developed population-based metaheuristic algorithm that models the hunting behavior of crocodiles [18]. The main advantages of RSA are concluded as achieving a near optimum solution with reasonable quality for the tested optimization problems, ease of implementation and its control parameters are few. These advantages motivate many researchers to use it for optimization of many engineering problems such as optimize an adaptive neuro-fuzzy inference system (ANFIS) to predict the swelling potentiality for fine-grained soils in the foundation bed [72], optimization of switching angle of the selective harmonic elimination (SHE) [73], parameter extraction of photovoltaic models [74], structure design optimization [75], routing and clustering in cognitive radio sensor network [76], and image retrieval [77].

However, there are some disadvantages of RSA such as the influence of objective value on the updating mechanism of solution, self-learning mechanism vanish, slow convergence, poor balancing mechanism between exploitation and exploration, and high chance for trapping in local optima. These drawbacks motivated many researchers to enhance the behavior of RSA.

In [78], RSA was enhanced by using Levy flight and crossover strategy (LICRSA) to improve the poor and slow convergence accuracy of optimization problems. Levy flight was used to increase the flexibility and variety of solutions to ban early convergence and enhance the robustness of the solution. LICRSA was applied to solve CEC2020 functions and 5 mechanical engineering optimization problems against other methods from the literature. LICRSA had superiority and powerful stability for gripping the engineering optimization problems. Furthermore, RSA has been utilized in power system applications to estimate the parameters of a PID controller, where it incorporates the Levy flight and Nelder–Mead algorithm [79]. Besides, an integration of RSA and Levy flight was presented for parameters estimation of PID controller for vehicle cruise control system [80].

A hybrid between RSA and remora optimization algorithm (ROA) was presented to enhance the data clustering process of data mining [81]. The proposed hybrid methods avoid the weakness of RSA and enhance the quality of found solution. The proposed method was tested on 20 mathematical benchmark functions and 8 data clustering problems where it beats comparative methods in terms of accuracy of the obtained solution. A binary RSA version was implemented based on different chaotic maps (CRSA) to solve feature selection problem in machine learning [82]. Three objectives combined as objective function were maximization of classification accuracy, produced wrapper models' complexity, and the size of features. Twenty UCI datasets were applied as test data and the results shown the superiority of CRSA over well-known techniques in the literature.

To tackle the feature selection problem in telecommunication applications, RSA has been combined with the ant colony optimization (ACO) algorithm [83]. Seven customer churn prediction datasets were used as test data for evaluation of ACO–RSA FS technique where its performance beat that of other comparative algorithms. In [84], the weakness of conventional RSA was enhanced based on mutation technique. The proposed algorithm is evaluated using CEC2019 benchmark functions and 5 industrial engineering optimization design problems. Snake optimizer (SO) algorithm was merged with RSA in parallel manner to achieve the optimal features subset in a given datasets [85]. The ideas of hybridization avoid RSA for trapping in local optima by enhancing the balancing between exploration and exploitation of the search space.

A modified version of RSA was presented to enhance the population diversity using an adaptive chaotic reverse learning strategy [86]. In addition, the elite alternative pooling strategy is introduced to balance between exploitation and exploration process. The improved algorithm was evaluated using both benchmark CEC2017 functions and the robot path planning problem, and it demonstrated superior performance in comparison to other algorithms in terms of accuracy, convergence speed, and stability. In [87], RSA was enhanced by merging the chaos theory to enhance the exploration capability while simulated annealing algorithm was used to avoid trapping in local optima. The proposed technique's performance was evaluated to optimize the feature selection of medical datasets. However, these trials of enhancement of RSA achieve a reasonable accuracy of the obtained solutions but still trials to enhance the avoidance of RSA from trapping in local optima and quick the convergence rate. This paper aims to hybrid the reinforcement learning technique and opposition operators to enhance the performance of RSA.

Reinforcement learning (RL) which is a category of machine learning is used to solve many optimization problems [88]. Machine learning techniques categorized into four kinds are supervised learning, semi supervised learning, unsupervised learning, and reinforcement learning (RL). In RL techniques, the agents are learned to be trained on the optimal behavior in the ganglion environment. The agents are trained to use their training knowledge the ulterior actions. Two categories of RL algorithms are model-based and model-free approaches. The model-free approach can be categorized into two kinds which are policy-based and value-based techniques. The value-based techniques suitable for coordination with metaheuristic techniques since they are policy-free and model-free which supply more flexibility.

The agent is trained from the experience of environment and its action in the value-based RL techniques via penalty and reward. The agent mensuration success for performing the task aim via the reward penalty then its decision was made according to its achievement. The Q-learning technique is most the common one of the values-based RL techniques. The agent is obtained penalty or reward after it did random actions. According to the subsequent agent's actions, the experience is progressively constructed. The Q table is constructed through the operation of constructing experience [89]. All allowable actions are considered by each agent and according to the values of Q table agent's state is updated to choose the optimum action which maximizes the current reward's state. Hence, the agent decides to exploit or explore the search space. The advantage of RL algorithms over metaheuristic algorithms that it balances between the exploitation and exploration of the search space and also can determine the optimum design of parameters. The dynamic search of metaheuristic algorithm is lower than that of RL techniques (for value-based techniques especially) due to metaheuristic algorithm working with determined policies in specific situations. However, in value-based techniques the agent determines its action according to the reward penalty strategy with no need for any specified policies and this operation of change is online.

RL techniques were merged with many metaheuristic techniques to enhance it. In [90], GA was improved by merging with RL with mutation (RMGA) to solve traveling salesman problem (TSP). In RMGA, the pairing selection is performed heterogeneously instead of randomly in the edge assembly crossover (EAX). In addition, the reinforcement mutation operator is performed by adjusting the Q-learning technique which is applied to the agent produced from altered EAX. The developed technique RMGA was applied on TSP with small and large instances and had superiority over than traditional GA and EAX–GA in terms of running time and quality of solutions. Teaching learning-based optimization (TLBO) was improved by merging RL technique [88]. RL–TLBO was modified via two phases: First, the effect of the teacher is presented as a new learner process. Second, a switching process between two learning modes is introduced in the learner phase based on the Q-learning method in RL technique. This enhancement aimed to enhance the speed of convergence and the accuracy of solution. CEC mathematical benchmark functions were employed to evaluate the performance of the algorithm, which was also applied to solve engineering design problems.

The particle swarm optimization (PSO) was enhanced using reinforcement learning and the per-training concept for enhancing the parameter adaptation ability to enhance the convergence speed [91]. The adaptation of the single parameter of gray wolf optimizer (GWO) was presented based on reinforcement learning techniques merged with neural networks to improve the performance of GWO [92].

The performance of the enhanced GWO was evaluated on choosing the weight' values of the neural network for feature selection to produce the superior over the convenient GWO and other comparative algorithms.

Artificial bee colony algorithm with modified by using Q-learning technique to reduce the tardiness of distributed three-stage assembly scheduling problem to the minimum level [93]. The Q-learning technique is applied to choose the search operator dynamically and it consists of 12 states which is based on the quality evaluation of the population. Besides, its actions are 8 which is determined by the neighborhood and global search where a new and effective action selection and reward were presented.

To enhance the exploration capability and enhance the search efficiency, two employed bee swarms are constructed where an adaptive competition operation and communication between them are embraced.

Opposition-based learning (OBL) operators were used to enhance the exploration capability of the metaheuristic algorithms and increase the avoidance of trapping in local optima. For example, in [94] salp swarm algorithm (SSA) was improved through the integration of OBL to enhance the accuracy of solutions, balance the exploration and exploitation scheme, and improve the convergence process of SSA. SSA–OBL was evaluated using CEC2015 benchmark mathematical functions and was applied to solve real-world optimization problems such as spacecraft trajectory optimization problem and circular antenna array design problem.

The imbalance between exploitation and exploration of PSO was enhanced by merging OBL for optimizing the feature selection operators [95]. Datasets consist of 24 benchmarks that are used to evaluate PSO-OBL algorithm against other metaheuristic algorithms in the literature. The algorithm shows superiority over other techniques in terms of high prediction accuracy. Hunger games search algorithm (HGSA) was merged with OBL to tune the fractional order PID controller [96]. Besides, the enhanced version of HGSA was evaluated on CEC2017 test functions.

An improved SCA by applying the opposition of solutions to increase the exploration of SCA (ISCA) [97]. In m-SCA [98], SCA enhanced by applying the opposition on the solutions beside adding a self-adaptive parameter was added in the updating equations of SCA to enhance the exploitation of promising regions of search space.

Improved Harris hawks optimization (HHO) based on OBL was introduced to address the problem of trapping in local optima and quick the convergence of search process [99]. HHO–OBL's performance was evaluated on CEC2017 benchmark mathematical test functions and 5 constrained engineering problems such as feature selection using 7 UCI datasets.

OBL has been introduced to enhance the arithmetic optimization algorithm (AOA) and prevent it from getting stuck in local optima [100]. This integration aims to improve the algorithm's ability to find the global optimal solution. Besides, a spiral model was used for accelerating the convergence speed of AOA. The enhanced AOA's performance was evaluated on 23 mathematical benchmark functions and 4 engineering optimization problems were the tubular column design, the cantilever beam design, the three-bar truss design, and the pressure vessel design. TLBO was merged with random OBL to enhance the avoidance of local optima beside using RL techniques to accelerated the convergence speed [88].

Based on the literature study, it is concluded that RL technique can effectively balance the exploration and exploitation of many metaheuristic algorithms and accelerate its convergence speed. Moreover, OBL was embedded in many algorithms to enhance the avoidance of trapping in local optima. These studies motivate us to embed RL technique and OBL in RSA to overcome its drawbacks.

The novelty of this work is listed as follows:

1: Introduce reinforcement learning is applied to find a proper balance between exploration and exploitation and therefore prevent premature convergence.

2: Prevent the tapping in local minimum through exploring the search space by introducing random opposition-based learning.

The paper is organized as follows: Sect. 2 represents preliminaries including the standard RSA and the concept of reinforcement learning. Section 3 shows the novelty of the RLRSA. Section 4 demonstrates experimental results and discussion. Finally, Sect. 5 concludes the work.

2 Preliminaries

2.1 Reptile search algorithm (RSA)

RSA is a novel optimization algorithm inspired by the hunting behavior of crocodiles, which involves encircling and hunting prey. The algorithm is modeled into two mechanisms that are divided into the global search phase, represented by the encircling behavior and is shown in Fig. 1, and the local search phase, represented by the hunting behavior and is shown in Fig. 2. These mechanisms occur in the first and second half of the iteration, respectively. The following two subsections discuss the characteristics of these two mechanisms.

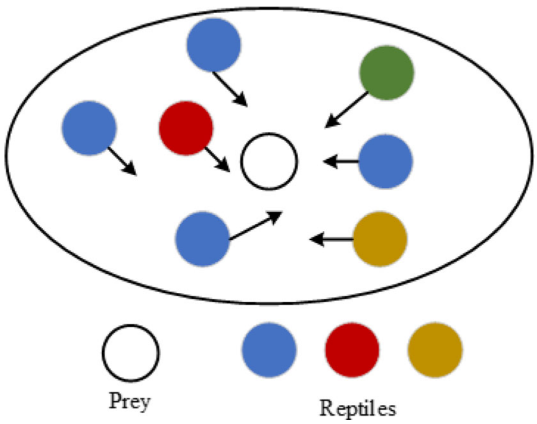

Fig. 1 Encircling phase

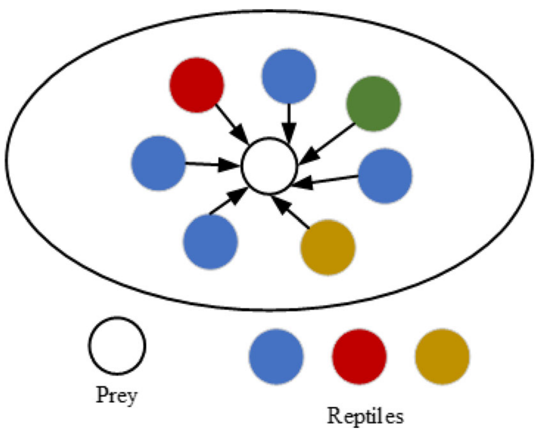

Fig. 2 Hunting phase

2.1.1 Encircling phase

The global exploration phase of RSA employs two main operators, high walking and belly walk, which facilitate the hunting phase and help identify areas with a high density of prey. High walking and belly walk occur in the first and second half of the exploration phase, respectively. Accordingly, the crocodile positions are updated as demonstrated in Eq. 1

$$x_{(i,j)}(t+1) = \begin{cases} x_{\text{best}(j)}(t) \times -\eta_{(i,j)}(t) \times B - R_{(i,j)}(t) \times rand t \leq \dfrac{T}{4} \\ x_{\text{best}(j)}(t) \times x_{(r1,j)} \times ES(t) \times rand \dfrac{T}{4} < t \leq 2.\dfrac{T}{4} \end{cases} \tag{1}$$

where $x_{\text{best}(j)}(t)$ at index j corresponds to the jth position of the best solution discovered in the current iteration t, $\eta_{(i,j)}(t)$ is the jth position hunting operator for the ith solution and based on Eq. 2,, $\mathcal{B}$ is a control parameter used to increase the accuracy of high walking phase, $R_{(i,j)}(t)$ determine the search region and is can be estimated using Eq. 3, rand is a random number $\in$ (0,1) drawn from uniform distribution, $x_{(r1,j)}$ is the jth position of random selected solution, and ES(t) is evolutionary sense that represents a probability ratio between 2 and -2 and is calculated according to Eq. 4.

$$\eta_{(i,j)}(t) = x_{\text{best}(j)}(t) \times p_{(i,j)} \tag{2}$$

$$R_{(i,j)}(t) = \frac{x_{\text{best}(j)}(t) - x_{(r2,j)}}{x_{\text{best}(j)}(t) + \in} \tag{3}$$

$$ES(t) = 2 \times r_3 \times \left(1 - \frac{1}{T}\right) \tag{4}$$

where $p_{(i,j)}$ represents the distance between the jth positions of fittest individual and the position of the current individual and is estimated using Eq. 5, ε represents a small value, r_2 random number $\in [1, N]$, and r_3 is random number $\in [-1, 1]$

$$p_{(i,j)} = \propto + \frac{x_{(i,j)} - M(x_i)}{x_{\text{best}(j)}(t) \times \left(\text{UB}_j - \text{LB}_j\right) + \varepsilon} \tag{5}$$

where $\propto$ is also a parameter that control the exploration accuracy and equals to 0.1, $M(x_i)$ is the average position of the ith solution and is calculated from Eq. 6, UB_jandLB_j are the upper and lower boundaries of the j-th position.

$$M(x_i) = \frac{1}{n}\sum_{j=1}^{n} x_{(i,j)} \tag{6}$$

2.1.2 Hunting phase

The exploitative phase of RSA occurs in the second half of the iteration, following the encircling phase, and consists of hunting coordination and cooperation strategies. These two strategies involve an intensive search near optimal solution and can be demonstrated in Eq. 7.

$$x_{(i,j)}(t+1) = \begin{cases} x_{best(j)}(t) \times P_{(i,j)}(t) \times rand \dfrac{2T}{4} < t \leq \dfrac{3T}{4} \\ x_{best(j)}(t) \times -\eta_{(i,j)}(t) \times \in -R_{(i,j)}(t) \times rand \dfrac{3T}{4} < t \leq T \end{cases} \tag{7}$$

The first part of Eq. 7 represents coordination hunting, whereas the second part represents cooperation hunting. In this context, when $t \leq \frac{T}{2}$,the encircling phase takes place (global search), otherwise when$t < \frac{T}{2}$, the hunting phase happens (local search) to find near/optimal solution. Algorithm 1 presents the pseudocode for RSA. The algorithm initializes the parameters and the population in the first line. The algorithm searches for the solution from line 3 to line 13 until the termination condition is satisfied as demonstrated in line 3. Line 9 of the algorithm updates the position of each individual in every iteration using either Eq. 1 or Eq. 7.

The best solution obtained thus far is output by the algorithm in line 14.

Algorithm 1: Pseudo-code of RSA

```
1:   Initialize RSA parameters, initialize population with random positions
2:   While t < T loop
3:       Estimate the fitness of promising solutions
4:       Find the best solution
5:       Calculate ES using Eq. 4
6:       For (i =1 to N) do
7:           For (j=1 to n) do
8:               Calculate η(i,j), P, R using Eq 2, 3, and 5
9:               Update position of Crocodile using either using Eq. 1 or Eq. 7
10:          End for
11:      End for
12:      t = t+1
13:  End While
14:  Output the best solution
```

2.2 Reinforcement learning

RL has been widely used to solve problems across various domains. The fundamental idea of RL involves an agent taking an action that alters the environment state, and receiving a reward based on that action. RL has two different categories: policy-based method and value-based method. Q learning (QL) is an example of a value-based method. It is model-free in which the agent learns how to act properly in Markovian domain [101]. During the learning phase, the agent executes the action with the highest expected Q value. The simplest form of Q learning is one-step Q learning in which the Q value is updated according to the state action in one step. The one-step Q learning is applied in this work. The Q table is dynamically updated according to the reward of each state action as in Eq. 8.

$$Q(s_t, a_t) \leftarrow (1 - \odot)Q(s_t, a_t) + \odot(r_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1})) \tag{8}$$

where $\odot$ and γ are the learning rate and discount factor, respectively, and both $\in [0, 1]$. The $Q(s_t, a_t)$ is the Q value of taking action a_t in the current state s_t whereas $\max_a Q(s_{t+1}, a_{t+1})$ is the highest expected Q value in Q table when taking action a_{t+1} on state s_{t+1}. It is worth mentioning that the highest learning rate $\odot$ makes the algorithm learn from expected Q value, whereas the low value of learning rate makes the algorithm exploit the previous Q value. Therefore, the learning rate balances between exploitation and exploration.

Algorithm 2 shows pseudo-code for Q learning. The algorithm 2 initializes the Q table and the reward table with random values then the algorithm randomly selects random state. The algorithm then chooses the action corresponding to the selected state that maximize the future reward as indicated in line 4 and 5. Consequently, the Q table, reward table, and the new state are updated. The algorithm repeats the line 3–9 until the termination condition is stratified.

Algorithm 2: Pseudo-code of Q-Learning

1: **Initialize** Q-table and reward table with random values
2: Choose random state s_t
3: **While** terminal condition is not satisfied **loop**
4: Choose the best action a_t for the current state s_t from Q-table
5: Execute the action and the reward r_{t+1}
6: Get the new state s_{t+1}
7: Update Q-table using Eq. 8
8: $s_t \leftarrow s_{t+1}$
9: **End While**

3 The development of the proposed RLRSA

3.1 Main idea of the proposed method

The basic RSA algorithm involves two main strategies to find solutions: exploration and exploitation. During the exploration phase, highly efficient walking and belly walking strategies are used to search for new solutions. On the other hand, during the exploitation phase, hunting coordination and hunting cooperation are used to refine the best solutions. However, the algorithm's ability to change direction is limited because exploration is done in the first half of the iterations, while exploitation is done in the second half. Consequently, the algorithm is susceptible to being trapped in local optima. This presents a limitation in that RSA is unable to change direction during the iterations, which makes it susceptible to being trapped in local optima. Therefore, the global minimum is not guaranteed by static search behavior. The adaptive search operation becomes a better choice for obtaining the global minimum and reinforcement learning is introduced to accomplish this task effectively. Opposition-based learning introduced to enhance the search for potential solutions is by augmenting the variety within the population.

3.2 The RLRSA structure

In RLRSA, the search space is considered as the interactive environment, whereas each individual is regarded as the train agent of RL. The Q-learning method is utilized to dynamically switch between exploration and exploitation. In Q-learning method, the Q value corresponding to state action is updated based on the current best fitness and average fitness in the previous iterations. In addition, a reward table is used to reward or penalize the agent based on the current action and state. The details are demonstrated in the following:

3.2.1 State set and action set

The proposed RLRSA has three actions corresponding to the value of the exploration rate (φ), namely increase exploration rate, decrease the exploration rate, and no change. The value of φ in the next iteration is updated based on the best fitness in the current iteration and the accumulated average fitness value as follows

$$\varphi^{t+1} = \begin{cases} \varphi^t * (1+\Delta) & \text{if } f\left(X^t_{\text{best}}\right) > M \\ \varphi^t * (1-\Delta) & \text{if } f\left(X^t_{\text{best}}\right) < M \\ \varphi^t & \text{otherwise} \end{cases} \tag{9}$$

where φ^{t+1} is the exploration rate at next iteration, and Δ is the incremental value, $f\left(X^t_{\text{best}}\right)$ is the fitness of the best position in the current iteration. M is the average fitness of the fittest individuals found so far and can be calculated as

$$M = \frac{1}{n}\sum_{t=1}^{n} w_i X^t_{\text{best}} \tag{10}$$

where n is the number of executed iterations until the current one, and w_i is the weighted factor for the fittest individual X^t_{best} at the iteration t and can be calculated as follows.

$$w_i = e^{t/T} \tag{11}$$

where t is the current iteration and T is the total number of the iteration. It is worth mentioning that the recently fittest individuals have more significant contributions in calculating the value of M. In more details, if they obtained fitness is greater than the mean average fitness, the algorithm needs to narrow the searching area and enhance the obtained solutions otherwise, the algorithm explores more searching area to find new solutions and avoid to be trapped in local optima and premature convergence. To sum up, the first case of Eq. 9 commonly happens when the agent successfully obtains better fitness than the average fitness. The second case shows that the fitness of the agent begins to deteriorate compared to the previous agent's experience. Finally, the exploration rate does not change if the agent is not motivated to change the exploration rate.

The RLRSA has three states $s = \{1, -1, 0\}$ corresponding to the after-mentioned actions as demonstrated on Eq. 12

$$s_t = \mathrm{sign}(f(x^t) - M), \mathrm{sign}(x) = \begin{cases} 1 & \text{if } x > 1 \\ -1 & \text{if } x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where s_t is the state obtained by the agent at the iteration t.

3.2.2 Reward

The reward table in this work contains the positive (+ 1) corresponding to the state $s_t = 1$ and negative (− 1) otherwise. The current state s_t is equal to 1 in case of the obtained fitness at iteration t is better than the average fitness of the last $t - 1$ iterations. Consequently, Eq. 13 demonstrates the reward method.

$$\mathrm{Reward} = \begin{cases} +1 & \text{if } s_t = 1 \\ -1 & \text{otherwise} \end{cases} \tag{13}$$

3.2.3 Adaptive learning rate

In addition, the proposed RLRSA carefully adapts the learning rate based on the accumulated performance because it has a significant impact on getting optimal solution. On the one hand, when the learning rate is close to one, the newly acquired information has a large impact on the future reward. When the learning rate is low, existing information is more valuable than newly acquired information. To achieve the best results, the learning rate is reduced adaptively through iteration using the equation below.

$$\odot = \frac{\odot_{\mathrm{init}} + \odot_{\mathrm{final}}}{2} - \frac{\odot_{\mathrm{init}} - \odot_{\mathrm{final}}}{2} . \cos\left(\pi\left(1 - \frac{t}{T}\right)\right) \tag{14}$$

where $\odot_{\mathrm{init}}, \odot_{\mathrm{final}}$ are the initial and the final value of the learning rate, respectively.

3.2.4 Random opposition-based learning

One of the most robust and effective methods to increase the diversity of the population is the opposition-based learning (OBL) is proposed by [102]. To enhance the performance of the algorithm, this method explores the opposite location of the individual being evaluated. Random opposition learning (ROBL) is introduced by [103] and uses randomization to improve the OBL methods which is defined as follows.

$$x_j\prime = l_j + u_j - \mathrm{rand} \times x_j, j = 1, 2, \ldots\ldots, n \tag{15}$$

where x_j', x_j represent the opposite and original solutions, and l_j and u_j indicate the lower and upper bound of the variables. OBL is introduced in RLRSA to adaptively help the algorithm to avoid trapping in local optima and therefore improve the obtained solution.

3.3 Overview of the RLRSA

Figure 3 shows the flowchart of the proposed RLRSA and algorithm 3 demonstrates more details on how the algorithm searches for the global solution. In the initialization phase, the algorithm initializes the basic parameters, states, actions, Q table, and reward table. The algorithm searches for the global if the stopping criteria is not reached. The searching process begins by calculating the fitness of the individuals and finding the best solution then the best action is selected from the Q table and the exploration rate is updated accordingly based on this selection using Eq. 9. Based on random number and exploration rate, either exploration or the exploitation takes place, and the new solution is generated. The reverse solution is then calculated using OBL based on the newly obtained solution after that the elitism mechanism is used to select the fittest solution between the reverse solution and the new solution. Finally, the Q table, reward table and the current state are updated. The learning rate is updated after each iteration and the process is repeated until the stopping criteria is satisfied.

3.4 Time complexity

As demonstrated in Fig. 3, the RLRSA algorithm divides the finding of the global solution into two phases. In the first phase, the algorithm initializes the population, and this phase is executed on $T = \boldsymbol{O}(\boldsymbol{N})$ and in the second phase, the searching for the global solution is repeated T_{max} times. Accordingly, the computation complexity of each phase can be demonstrated as follows.

4 Phase 1: Initialization.

The algorithm initializes N population with random values for the next stage and thus the complexity of this phase is $\boldsymbol{O}(\boldsymbol{N})$.

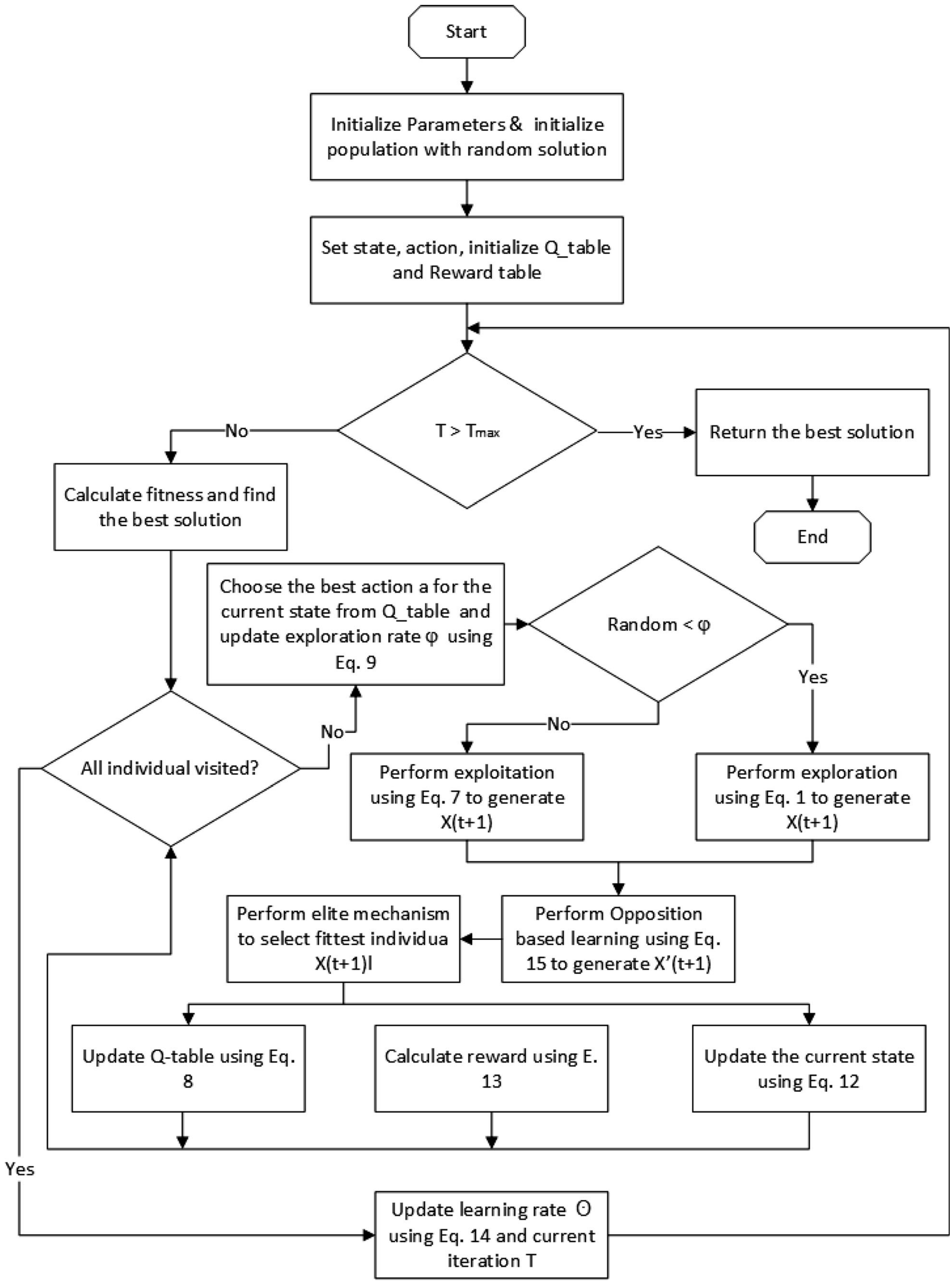

Fig. 3 The flowchart of RLRSA algorithm

5 Phase 2: Searching

The searching process of the RLRSA includes four operations: exploitation, exploration, random opposite-based learning, and elitism mechanisms. Because the algorithm switch between exploration and exploration based on the accumulated performance; therefore, it is assumed that each process is executed for all individuals. In the exploration phases, all individuals are directed toward the global solution, whereas during the exploitation process, all the individuals move toward the local solution. The total complexity of exploration and exploration process is $\boldsymbol{O}(\boldsymbol{N} \times \boldsymbol{T})+ \boldsymbol{O}(\boldsymbol{N} \times \boldsymbol{T} \times \boldsymbol{D})$. Therefore, the computational complexity of RLRSA can be expressed as: $\boldsymbol{O}(\boldsymbol{N}) + \boldsymbol{O}(\boldsymbol{N} \times \boldsymbol{T})+ \boldsymbol{O}(\boldsymbol{N} \times \boldsymbol{T} \times \boldsymbol{D})$. Here, T represents the maximum number of iterations, and D denotes the dimensionality of the solution.

Table 1 Parameters setting

Algorithm	Parameters	Values
RSA/LIRSA	α	0.1
	β	0.1
SSA	c1, c2,c3	Random between 0 and 1
GWO	Convergence parents a	Reduced linearly from 2 to 0
PSO	Topology	Fully connected
	Vmax	6
	wMax	0.9
	wMin	0.6
	c1, c2	2
MPA	Human caused (FADs)	0.2
	High velocity	$\geq$ 10
GOA	Cmax, Cmin	1, 0.00001
WOA	α	Decreased from 2 to zero
ALO	Iratio	10^{w}
	w	2 to 6
EO	GP	0.5
	a1, a2	2, 1
DA	α	0.5
	w	0.2–0.9
CMA-ES	Number of parents	$\frac{\lambda}{4}$

Table 2 Unimodal benchmark functions

Function	Description	Dimensions	Range	$f_{\min}$
F1	$f(x) = \sum_{i=1}^{n} x_i^2$	30, 100, 500, 1000	[− 100, 100]	0
F2	$f(x) = \sum_{i=0}^{n} \lvert x_i\rvert + \prod_{i=0}^{n} \lvert x_i\rvert$	30, 100, 500, 1000	[− 10, 10]	0
F3	$f(x) = \sum_{i=1}^{d} (\sum_{j=1}^{n} x_j^2)$	30, 100, 500, 1000	[− 100, 100]	0
F4	$f(x) = \max_i\{\lvert x_i\rvert, 1 \leq i \leq n\}$	30, 100, 500, 1000	[− 100, 100]	0
F5	$f(x) = \sum_{i=1}^{n} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$	30, 100, 500, 1000	[− 30, 30]	0
F6	$f(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30, 100, 500, 1000	[− 100, 100]	0
F7	$f(x) = \sum_{i=0}^{n} ix_i^4 + \text{random}[0, 1)$	30, 100, 500, 1000	[− 128, 128]	0

Algorithm 3: Pseudo-code of RLRSA

```
Initialize RSA parameters, initialize population with random positions.
Set the state s = {s1, s2, s3} and action a = {a1, a2, a2}.
Initialize Q-table and reward table.
Randomly select current state
While t < T loop
    Estimate the fitness of promising solutions
    Find the best solution
    Calculate ES using Eq. 4
    Calculate exploration rate φ using Eq. 9
    For (i =1 to N) do
        For (j=1 to n) do
        If rand < φ              // do exploration
            If rand < 0.5        // high walking
                Update position of Crocodile from first part of Eq. 1
            Else                 // belly walk
                Update position of Crocodile from second part of Eq. 1
            End if
        Else                     // do exploitation
            If rand < 0.5        // hunting coordination
                Update position of Crocodile from first part of Eq. 7
            Else                 // hunting cooperation
                Update position of Crocodile from second part of Eq. 7
            End if
        End if
        Calculate X'_j(t + 1) using Eq. 15
        End for
        If F(X'_j(t + 1) is better than F(X(t + 1))
            X(t + 1) = X'_j(t + 1)
        End if
    Update the current state using Eq. 12
    Calcutta reward using Eq. 13
    Update Q-table using Eq. 8
    End for
    t = t+1
    Update learning rate using Eq. 14
End While
Output the best solution
```

6 Experimental setup and discussions

In this study, twenty-three global optimization problems are used to evaluate the performance of the proposed RLRSA. In addition, the proposed RLRSA is applied to find the best solution for one of the most common engineering problems, namely pressure vessel design problem. In order to evaluate the effectiveness of the proposed RLRSA, it was compared against several other global optimization algorithms, which included: ant lion optimizer (ALO) [104], covariance matrix adaptation evolution strategy (CMAES) [105], slap swarm algorithm

(SSA) [106], marine predator algorithm (MPA) [107], dragonfly algorithm (DA) [108], whale optimization algorithm (WOA) [17], equilibrium optimizer (EO) [109], particle swarm optimization algorithm (PSO) [110], grasshopper optimization algorithm (GOA) [111], gray wolf optimizer (GWO) [23], sine–cosine algorithm (SCA) [31], and the standard RSA.

In experiments, the common parameters setting, such as the number population is set to 30, the number of independent runs is set to 25, and the number of function evaluations is set to 15,000. In addition, other parameters values of algorithms are set as their implementation and are shown in Table 1. Furthermore, the parameter settings of RLRSA are identical to those of the standard RSAseco [18]. Statistical measurements are used to quantify the quality of the obtained solution including the worst, best, average, and standard deviation of the fitness values.

Three scenarios are used to evaluate the effectiveness of RLRSA against other methods. The first scenario shows the comparative analysis of RLRSA and other methods to unimodal and multimodal benchmark functions with solve 10, 500, 100, and 500 dimensions. The second scenario evaluates the performance of RLRSA on fixed-dimensional multimodal benchmark functions. The fourth scenario shows how the RLRSA can be used to solve real-world problems and compare the performance of RLRSA against other methods.

6.1 Description of the benchmark

The twenty-three benchmarks used to evaluate the performance of the proposed RLRSA are classified based on the number of extreme solutions. The first type is known as unimodal functions and the main feature of this type is that it has only one single solution in the search space. Table 2 shows the unimodal functions (F1–F7) and their definition. The 2nd and 3rd categories of benchmark functions are, respectively, classified as multimodal and fixed-dimension multimodal. The main characteristics of these types are that they have more than one extreme solution in the search space. Functions F8–F13 and their definition shown in Table 3 are examples for multimodal function, whereas Table 4 shows the functions F13–F23 and their description as examples of multimodal with fixed dimensions.

6.2 Results and discussion

This section presents a comprehensive comparison of the performance of the proposed reinforcement learning-based RSA (RLRSA) against the standard RSA and other methods from the literature. The objective is to identify the best solution for both unimodal and multimodal functions. In all

Table 3 Multimodal benchmark functions

Function	Description	Dimensions	Range	$f_{\min}$
F8	$f(x)=\sum_{i=}^{n}\left(-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)\right)$	30, 100, 500, 1000	[− 500, 500]	− 418.9829 × n
F9	$f(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30, 100, 500, 1000	[− .521, 5.21]	
F10	$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)$	30, 100, 500, 1000	[− 32, 32]	0
F11	$f(x)=1+\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	30, 100, 500, 1000	[− 600, 600]	0
F12	$f(x)=\frac{\pi}{n}\{10\sin(\pi y_1)\}$ $+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2\left(\pi y_{i+1}\right)+\sum_{i=1}^{n}u(x_i,10,100,4)\right]$, where $y_i=1+\frac{x_i+1}{4},\ u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m if x_i>a\\ 0-a\le x_i\ge a\\ k(-x_i-a)^m-a\le x_i\end{cases}$	30, 100, 500, 1000	[− 50, 50]	0
F13	$f(x)=0.1(\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]+(x_n-1)^2$ $+\sin^2(2\pi x_n)+\sum_{i=1}^{n}u(x_i,5,100,4)$	30, 100, 500, 1000	[− 50, 50]	0

Table 4 Fixed-dimension multimodal benchmark functions

Function	Description	Dimensions	Range	fmin
F14	$f(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})}\right)^{-1}$	2	[− 65, 65]	
F15	$f(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_i\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[− 5, 5]	0.0030
F16	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[− 5, 5]	− 1.0316
F17	$f(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)cosx_1+10$	2	[− 5, 5]	0.398
F18	$f(x)=\left[1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]$ $\times\left[30+(2x_1-3x_2)^2\times\left(18-32x_1+12x_1^2=48x_2-36x_1x_2+27x_2^2\right)\right]^2$	2	[− 2, 2]	3
F19	$f(x)=\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}(x_i-p_{ij})^2\right)$	3	[− 1, 2]	− 3.86
F20	$f(x)=\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}(x_i-p_{ij})^2\right)$		[0, 1]	− 0.32
F21	$f(x)=-\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	4	[0, 1]	− 10.1532
F22	$f(x)=-\sum_{i=1}^{7}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	4	[0, 1]	− 10.4028
F23	$f(x)=-\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	4	[0, 1]	− 10.5363

tables, the algorithms are ranked based on average performance.

Various algorithms are compared in Table 5, which presents the worst, average, best, and standard deviation (STD) fitness values for unimodal and multimodal functions with 10 dimensions. The results highlight that RLRSA achieves the smallest average values with small standard deviation for 12 out of 13 test functions, outperforming other optimization methods. The CMA-ES algorithm comes second and achieves better performance in 2 out of 13 test functions. This indicates that RLRSA demonstrates remarkable stability and accuracy in solving both unimodal and multimodal functions. Additionally, the results confirm that RLRSA exhibits superior exploration and exploitation capabilities, thanks to the learning behavior of the reinforcement agent and the ROBL strategy.

Furthermore, the performance of RLRSA is evaluated at different dimensions: 50, 100, and 500, as illustrated in Tables 6, 7, and 8, respectively. Table 6 reveals that RLRSA consistently achieves minimum fitness values for most problems at 50 dimensions. Specifically, RLRSA ranks first in 9 out of 13 functions. CMA-ES performs better on three functions (F6, F8, and F12). In addition, RSA, RICRSA, MPA, and WOA show a good performance on three functions.

Tables 7 and 8 show the performance of RLRSA, the basic RSA, and other methods at 100 and 500 dimensions, respectively. In Table 7, RLRSA demonstrates a superior average fitness value in 11 out of 13 functions (84%). However, RSA can find the best average fitness on only 2 out of 13 test functions. In addition, RICRSA ranks second with high performance on three test functions.

Moving to high dimensions (500), RLRSA continues to outperform other methods significantly, as indicated in Table 7. On the one hand, RLRSA shows a superior performance on 10 test functions compared to other methods. On the other hand, MPA comes second with good performance on four test functions.

Finally, Table 9 provides insights into the performance of different algorithms in discovering solutions for fixed-dimensional multimodal functions. On the one hand, RLRSA shows competitive results when compared to the basic RSA. Specifically, RLRSA surpasses the basic RSA and other methods in 8 out of 10 benchmarks. On the other hand, RSA comes second with good performance on 5 test functions. To sum up, the reinforcement agent enables RLRSA to effectively search for the global solution with minimum number of iterations and find the best solution in most benchmark functions.

Table 5 Comparison between the performance of RLRSA and other methods on unimodal and multimodal functions with 10 dimensions

Fun	Measure	RLRSA	RSA	LICRSA	SCA	GWO	PSO	MPA
F1	Worst	0.00E+00	6.38E-171	1..50E-09	8.11E-10	1.47E-48	6.24E-19	1.10E-29
	Average	0.00E+00	1.28E-171	1.05E-179	2.05E-10	7.04E-49	1.69E-19	5.17E-30
	Best	0.00E+00	2.86E-181	4.54E-10	2.31E-15	3.41E-51	2.90E-21	4.84E-31
	STD	0.00E+00	1.00E-02	4.07E-10	3.50E-10	5.59E-49	2.60E-19	3.91E-30
	Rank	1	3	2	10	6	9	8
F2	Worst	0.00E+00	6.90E-86	1.61E-01	1.94E-08	5.06E-27	5.69E-09	2.67E-16
	Average	0.00E+00	1.39E-86	3.31E-91	7.17E-09	1.84E-27	1.83E-09	1.12E-16
	Best	0.00E+00	4.06E-94	7.87E-06	2.28E-11	5.24E-29	1.86E-10	1.57E-17
	STD	0.00E+00	3.09E-86	7.12E-02	9.80E-09	2.28E-27	2.33E-09	1.31E-16
	Rank	1	3	2	10	6	9	8
F3	Worst	0.00E+00	2.68E-142	1.62E-02	1.59E+00	5.94E-19	8.95E-05	8.91E-13
	Average	0.00E+00	6.36E-143	3.12E-131	3.20E-01	1.46E-19	2.46E-05	2.97E-13
	Best	0.00E+00	1.21E-149	5.51E-06	8.80E-06	5.86E-22	3.03E-07	4.34E-15
	STD	0.00E+00	1.16E-142	7.22E-03	7.08E-01	2.53E-19	3.73E-05	4.11E-13
	Rank	1	2	3	9	6	8	7
F4	Worst	0.00E+00	1.05E-81	6.64E-04	4.26E-03	5.97E-15	9.08E-04	8.97E-13
	Average	0.00E+00	2.71E-82	1.09E-91	1.34E-03	2.15E-15	2.48E-04	4.84E-13
	Best	0.00E+00	2.18E-89	1.63E-05	4.39E-06	2.37E-16	2.10E-05	2.33E-13
	STD	0.00E+00	4.45E-82	2.80E-04	1.89E-03	2.56E-15	3.74E-04	2.72E-13
	Rank	1	3	2	9	6	8	7
F5	Worst	2.59E+00	3.99E+00	6.44E+02	8.24E+00	8.07E+00	7.66E+00	8.83E+00
	Average	5.60E-03	3.18E+00	1.21E+00	7.69E+00	6.98E+00	4.93E+00	8.79E+00
	Best	8.75E-04	2.39E+00	7.90E+00	7.29E+00	6.25E+00	1.03E-01	8.77E+00
	STD	2.72E-01	6.11E-01	2.61E+02	4.46E-01	7.65E-01	2.85E+00	2.39E-02
	Rank	1	4	3	9	7	5	10
F6	Worst	2.250E-12	1.740E-09	9.370E-01	5.170E-01	5.330E-06	2.040E-16	8.62–11
	Average	6.920E-23	1.320E-09	6.210E-08	4.120E-01	3.590E-06	4.370E-17	2.890E-11
	Best	1.360E-12	1.130E-09	3.140E-01	2.920E-01	2.530E-06	2.990E-21	3.770E-12
	STD	1.670E-10	2.460E-10	2.370E-01	9.680E-02	1.110E-06	8.970E-17	3.290E-11
	Rank	1	5	7	12	8	2	4
F7	Worst	7.49E-07	9.05E-05	3.65E-02	3.85E-03	1.35E-03	1.62–02	2.18E-03
	Average	5.82E-07	6.77E-05	5.23E-06	2.43E-03	6.45E-04	9.42E-03	1.09E-03
	Best	2.46E-07	2.03E-05	1.41E-02	1.49E-03	3.96E-04	3.97E-03	3.96E-04
	STD	1.52E-07	2.86E-05	8.75E-03	9.67E-04	3.91E-04	4.38E-03	7.21E-04
	Rank	1	3	2	8	4	10	5
F8	Worst	-1.70E+03	-2.89E+03	-2.50E+03	-1.91E+03	-2.30E+03	-1.53E+03	-3.25E+03
	Average	-5.08E+03	-2.05E+03	-2.99E+03	-2.01E+03	-2.72E+03	-2.12E+03	-3.54E+03
	Best	-2.42E+03	-3.60E+03	-3.27E+03	-2.11E+03	-3.15E+03	-2.60E+03	-3.99E+03
	STD	4.18E+01	5.63E+02	3.02E+02	7.81E+01	3.49E+02	4.44E + 02	2.723 + 02
	Rank	1	11	5	12	8	10	2
F9	Worst	0.00E + 00	1.00E-02	2.59E + 01	6.38E + 00	1.52E + 01	1.30E + 01	5.04E-07
	Average	0.00E + 00	4.00E-03	1.52E-08	1.28E + 00	4.32E + 00	5.62E + 00	1.02E-07
	Best	0.00E + 00	2.00E-03	5.98E + 00	2.72E-12	0.00E + 00	2.00E + 00	0.00E + 00
	STD	0.00E + 00	0.00E + 00	8.76E + 00	2.86E + 00	6.67E + 00	4.32E + 00	2.26E-07
	Rank	1	6	4	7	8	9	5
F10	Worst	0.00E + 00	8.89E-16	2.82E + 00	6.92E-06	1.52E-14	1.21E-09	8.00E-15
	Average	0.00E + 00	8.89E-16	6.32E-21	1.73E-06	1.16E-14	7.07E-10	5.16E-15
	Best	0.00E + 00	8.89E-16	9.24E-06	5.67E-08	8.00E-15	3.22E-10	4.45E-15
	STD	0.00E + 00	0.00E + 00	1.11E + 00	2.96E-06	3.65E-15	3.74E-10	1.60E-15

Table 5 (continued)

Fun	Measure	RLRSA	RSA	LICRSA	SCA	GWO	PSO	MPA
	Rank	1	3	2	10	8	9	7
F11	Worst	0.00E + 00	0.00E + 00	2.66E-01	4.74E-02	5.93E-02	5.35E-01	0.00E + 00
	Average	0.00E + 00	0.00E + 00	0.00E + 00	1.23E-02	1.40E-02	2.46E-01	0.00E + 00
	Best	0.00E + 00	0.00E + 00	1.34E-01	6.64E-12	0.00E + 00	1.02E-01	0.00E + 00
	STD	0.00E + 00	0.00E + 00	5.84E-02	2.05E-02	2.58E-02	1.80E-01	0.00E + 00
	Rank	1	1	1	7	8	11	1
F12	Worst	4.25E-25	9.18E-02	5.19E-01	2.20E-01	2.04E-02	2.82E-19	4.36E-11
	Average	4.05E-25	6.19E-03	1.23E-16	1.35E-01	1.03E-02	5.83E-20	2.55E-11
	Best	4.05E-25	3.56E-04	5.16E-11	6.61E-02	2.75E-06	2.05E-21	1.18E-11
	STD	6.05E-25	2.82E-02	2.15E-01	6.03E-02	9.99E-03	1.25E-19	1.33E-11
	Rank	2	7	4	10	8	3	6
F13	Worst	0.00E + 00	3.95E-01	1.12E-02	4.76E-01	2.01E-01	1.13E-02	9.51E-11
	Average	0.00E + 00	1.45E-04	2.23E-03	3.94E-01	4.01E-02	2.21E-03	5.68E-11
	Best	0.00E + 00	3.77E-09	5.79E-10	3.36E-01	6.56E-06	8.62E-22	3.60E-11
	STD	0.00E + 00	1.59E-01	4.92E-03	5.20E-02	8.95E-02	4.92E-03	2.39E-11
	Rank	1	4	6	12	10	5	3
Total		12	1	1	0	0	0	1

Fun	Measure	GOA	WOA	ALO	EO	DA	CMA-ES
F1	Worst	3.71E-04	3.01E-69	1.44E-07	7.77E-55	2.37E+01	7.54E-46
	Average	9.24E-05	1.03E-69	4.39E-08	1.96E-55	1.32E+01	3.87E-46
	Best	4.54E-06	1.31E-74	1.40E-08	4.35E-58	9.42E-01	9.03E-47
	STD	1.60E-04	1.45E-69	5.63E-08	3.35E-55	8.57E+00	2.48E-46
	Rank	12	4	11	5	13	7
F2	Worst	8.08E+00	7.06E-49	1.83E+00	1.76E-32	6.00E+00	1.32E-22
	Average	1.98E+00	1.42E-49	6.58E-01	4.77E-33	2.69E+00	7.55E-23
	Best	4.29E-02	2.34E-54	2.08E-05	4.48E-34	1.13E+00	4.55E-23
	STD	3.47E+00	3.16E-49	9.10E-01	7.13E-33	1.50E+00	3.54E-23
	Rank	12	4	11	5	13	7
F3	Worst	7.93E+00	9.10E+02	3.58E+01	1.39E-26	2.31E+02	2.12E-37
	Average	2.38E+00	4.21E+02	9.78E+00	2.82E-27	1.52E+02	7.70E-38
	Best	7.68E-01	1.90E+01	7.72E-01	3.30E-30	4.95E+01	8.53E-39
	STD	3.13E+00	3.75E+02	1.50E+01	6.10E-27	6.80E+01	8.76E-38
	Rank	10	13	11	5	12	4
F4	Worst	3.69E-01	1.03E+01	1.45E-01	4.52E-18	5.50E+00	9.82E-21
	Average	1.61E-01	3.17E+00	3.43E-02	9.12E-19	3.13E+00	4.97E-21
	Best	8.76E-02	2.11E-03	1.06E-03	7.27E-22	1.60E+00	2.75E-21
	STD	1.19E-01	4.06E+00	6.18E-02	2.03E-18	1.63E+00	2.90E-21
	Rank	11	13	10	5	12	4
F5	Worst	4.74E+02	8.94E+00	2.14E+02	5.84E+00	1.21E+04	2.38E-02
	Average	1.33E+02	7.66E+00	1.17E+02	5.55E+00	2.71E+03	1.89E-02
	Best	3.60E+00	6.83E+00	6.00E+00	5.32E+00	1.36E+02	1.48E-02
	STD	1.99E+02	8.27E-01	1.02E+02	2.46E-01	5.28E+03	3.33E-03
	Rank	12	8	11	6	13	2
F6	Worst	7.240E-04	2.350E-01	1.190E-07	8.570E-14	4.590E+01	4.200E-04
	Average	2.240E-04	5.030E-02	3.300E-08	4.130E-14	1.691E+01	3.310E-04
	Best	3.710E-07	1.210E-03	7.100E-09	4.860E-16	1.040E+00	7.920E-04
	STD	2.960E-04	1.040E-01	4.780E-08	3.310E-14	1.870E+01	4.790E-04
	Rank	9	11	6	3	13	10
F7	Worst	1.53E-01	8.92E-03	6.19E-02	1.87E-03	4.23E-02	1.94E-03
	Average	1.09E-01	5.22E-03	3.50E-02	1.31E-03	2.41E-02	1.09E-03

Table 5 (continued)

Fun	Measure	GOA	WOA	ALO	EO	DA	CMA-ES
	Best	5.46E-02	2.00E-03	1.65E-02	7.19E-04	7.40E-03	5.76E-04
	STD	4.37E-02	3.34E-03	1.92E-02	4.90E-04	1.65E-02	5.83E-04
	Rank	13	9	12	7	11	5
F8	Worst	-2.42E+03	-2.61E+03	-2.04E+03	-2.64E+03	-2.59E+03	6.55E+04
	Average	-2.76E+03	-3.10E+03	-2.47E+03	-3.20E+03	-2.76E+03	6.55E+04
	Best	-3.33E+03	-3.83E+03	-3.23E+03	-3.59E+03	-3.01E+03	6.55E+04
	STD	3.48E + 02	4.48E + 02	4.98E + 02	3.91E + 02	1.66E + 02	6.55E + 04
	Rank	6	4	9	3	6	13
F9	Worst	6.28E + 01	1.43E-14	4.29E + 01	0.00E + 00	4.74E + 01	1.61E + 01
	Average	4.35E + 01	2.85E-15	3.15E + 01	0.00E + 00	3.14E + 01	6.33E + 00
	Best	2.89E + 01	0.00E + 00	1.60E + 01	0.00E + 00	1.07E + 01	0.00E + 00
	STD	1.42E + 01	6.37E-15	1.15E + 01	0.00E + 00	1.52E + 01	7.50E + 00
	Rank	13	3	12	1	11	10
F10	Worst	3.58E + 00	8.00E-15	2.02E + 00	4.45E-15	1.28E + 01	8.89E-16
	Average	1.69E + 00	4.45E-15	4.04E-01	4.45E-15	4.99E + 00	8.89E-16
	Best	1.05E-03	8.89E-16	8.19E-05	4.45E-15	1.44E + 00	8.89E-16
	STD	1.31E + 00	3.56E-15	9.01E-01	0.00E + 00	4.45E + 00	0.00E + 00
	Rank	12	5	11	5	13	3
F11	Worst	2.96E-01	2.47E-01	3.77E-01	2.95E-02	1.06E + 00	0.00E + 00
	Average	2.59E-01	9.76E-02	2.36E-01	5.90E-03	5.90E-01	0.00E + 00
	Best	1.97E-01	0.00E + 00	1.22E-01	0.00E + 00	6.53E-02	0.00E + 00
	STD	4.20E-02	1.34E-01	1.14E-01	1.32E-02	3.76E-01	0.00E + 00
	Rank	12	9	10	6	13	1
F12	Worst	3.54E + 00	2.33E-02	5.64E + 00	6.86E-14	4.65E + 00	4.72E-32
	Average	1.28E + 00	1.16E-02	2.91E + 00	1.79E-14	1.64E + 00	4.75E-32
	Best	8.36E-04	1.49E-03	2.64E-01	6.00E-17	7.59E-02	4.74E-32
	STD	1.44E + 00	9.03E-03	1.96E + 00	2.88E-14	1.81E + 00	6.13E-48
	Rank	11	9	13	5	12	1
F13	Worst	5.57E-02	1.58E-01	1.11E-02	4.40E-02	1.60E + 00	1.37E-32
	Average	2.31E-02	7.28E-02	4.41E-03	8.79E-03	9.40E-01	1.36E-32
	Best	8.26E-04	2.50E-03	2.26E-07	4.03E-16	2.90E-01	1.36E-32
	STD	2.30E-02	6.47E-02	6.03E-03	1.97E-02	4.70E-01	0.00E + 00
	Rank	9	11	7	8	13	2
Total		0	0	1	1	0	2

6.3 Analysis of the convergence curve of RLRSA versus standard RSA

The convergence curve of the proposed RLRSA compared to other methods in the literature is shown in Fig. 4. The convergence curves show that the RLRSA has a noticeable improvement in exploration and exploitation when reinforcement learning is applied. For example, the convergence rate of RLRSA on F1-F4, F7-F11 is faster than that of other methods, and the convergence accuracy is also better. The RLRSA outperforms the other methods on F5 in the first twenty-two iterations and RSA achieves the same convergence rate as RLRSA after iteration 23. The RLRSA does not perform well on benchmarks F6 and F12. To recap, reinforcement learning enables the algorithm to efficiently explore the search space from the first iteration and successfully find the best position of the agent.

Table 6 Comparison between the performance of RLRSA and other methods on unimodal and multimodal functions with 50 dimensions

Fun	Measure	RLRSA	RSA	LICRSA	SCA	GWO	PSO	MPA
F1	Worst	2.30E-162	1.20E-159	1.02E + 03	7.02E + 03	3.70E-11	6.97E + 00	1.56E-21
	Average	2.04E-171	3.00E-160	2.32E-148	3.28E + 03	1.21E-11	3.99E + 00	4.26E-22
	Best	1.23E-171	8.94E-169	2.80E + 02	5.74E + 02	3.01E-12	2.36E + 00	3.67E-24
	STD	4.21E-170	5.98E-160	3.05E + 02	2.70E + 03	1.66E-11	2.03E + 00	7.49E-22
	Rank	1	2	3	11	7	9	5
F2	Worst	3.24E-82	5.79E-81	1.76E + 06	1.58E + 01	7.17E-05	1.09E + 02	7.93E-12
	Average	1.46E-84	1.45E-81	2.35E-83	7.97E + 00	6.72E-05	9.39E + 01	3.45E-12
	Best	1.02E-82	3.55E-86	2.31E + 02	1.43E + 00	6.35E-05	7.56E + 01	5.31E-13
	STD	5.21E-83	2.89E-81	8.75E + 05	7.01E + 00	3.52E-06	1.39E + 01	3.56E-12
	Rank	1	3	2	9	7	11	5
F3	Worst	0.00E + 00	2.24E-104	6.14E + 04	1.52E + 05	3.37E + 03	1.65E + 04	1.22E + 01
	Average	0.00E + 00	5.60E-105	5.64E + 04	8.90E + 04	1.62E + 03	1.39E + 04	3.28E + 00
	Best	0.00E + 00	5.26E-128	5.21E + 04	3.35E + 04	2.77E + 02	9.04E + 03	8.41E-02
	STD	0.00E + 00	1.12E-104	4.66E + 03	5.65E + 04	1.58E + 03	3.44E + 03	5.94E + 00
	Rank	1	2	8	10	6	7	4
F4	Worst	5.02E + 08	7.22E + 08	8.00E + 08	7.30E + 08	8.53E + 08	7.56E + 08	5.93E + 08
	Average	5.52E + 08	5.52E + 08	1.20E + 09	6.31E + 08	6.45E + 08	6.51E + 08	5.37E + 08
	Best	4.25E + 08	4.80E + 08	4.58E + 08	4.50E + 08	4.83E + 08	5.53E + 08	4.91E + 08
	STD	1.60E + 08	6.25E + 07	1.51E + 08	1.24E + 08	1.56E + 08	8.32E + 07	4.23E + 07
	Rank	2	2	13	6	7	8	1
F5	Worst	3.13E + 01	4.90E + 01	3.21E + 07	5.68E + 07	4.88E + 01	1.45E + 05	4.88E + 01
	Average	2.59E + 01	4.60E + 01	5.23E + 01	2.52E + 07	4.87E + 01	6.08E + 04	4.88E + 01
	Best	2.01E + 01	4.90E + 01	1.78E + 07	1.97E + 06	4.86E + 01	1.68E + 04	4.85E + 01
	STD	1.02E-01	1.80E-02	6.57E + 06	2.76E + 07	1.06E-01	6.03E + 04	1.09E-01
	Rank	1	2	7	11	4	9	5
F6	Worst	2.18E-05	1.23E + 01	1.66E + 04	8.98E + 03	7.91E + 00	8.91E + 01	5.27E + 00
	Average	2.23E-05	1.20E + 01	1.07E + 04	4.02E + 03	7.11E + 00	6.90E + 01	4.34E + 00
	Best	1.25E-06	1.17E + 01	5.78E + 03	4.04E + 02	6.23E + 00	4.91E + 01	3.70E + 00
	STD	3.13E-08	3.55E-01	4.48E + 03	3.76E + 03	6.91E-01	1.79E + 01	7.15E-01
	Rank	2	7	10	9	6	8	3
F7	Worst	0.00E + 00	3.03E-04	1.22E + 01	2.02E + 01	5.04E-02	3.20E + 02	8.15E-03
	Average	0.00E + 00	1.76E-04	2.02E + 00	1.52E + 01	2.69E-02	2.01E + 02	4.81E-03
	Best	0.00E + 00	7.87E-07	8.51E + 00	4.56E + 00	1.21E-02	1.22E + 02	2.48E-03
	STD	0.00E + 00	1.50E-04	1.69E + 00	7.22E + 00	1.74E-02	9.00E + 01	2.50E-03
	Rank	1	2	8	9	7	12	3
F8	Worst	− 3.84E + 10	− 6.66E + 03	− 8.65E + 03	− 3.68E + 03	− 6.85E + 03	− 2.83E + 03	− 1.06E + 04
	Average	− 2.45E + 10	− 7.51E + 03	− 1.09E + 01	− 4.03E + 03	− 7.71E + 03	− 3.91E + 03	− 1.16E + 04
	Best	− 8.45E + 10	− 8.81E + 03	− 9.62E + 03	− 4.26E + 03	− 8.89E + 03	− 5.28E + 03	− 1.28E + 04
	STD	2.54E + 08	9.25E + 02	4.38E + 02	2.49E + 02	8.92E + 02	1.03E + 03	9.66E + 02
	Rank	2	9	13	11	8	12	4
F9	Worst	0.00E + 00	0.00E + 00	4.36E + 02	3.56E + 02	3.06E + 01	4.74E + 02	0.00E + 00
	Average	0.00E + 00	0.00E + 00	0.00E + 00	2.03E + 02	2.08E + 01	3.59E + 02	0.00E + 00
	Best	0.00E + 00	0.00E + 00	3.11E + 02	1.07E + 02	8.94E + 00	3.02E + 02	0.00E + 00
	STD	0.00E + 00	0.00E + 00	5.84E + 01	1.10E + 02	1.08E + 01	8.07E + 01	0.00E + 00
	Rank	1	1	1	8	7	10	1

Table 6 (continued)

Fun	Measure	RLRSA	RSA	LICRSA	SCA	GWO	PSO	MPA
F10	Worst	0.00E + 00	4.45E-15	1.91E + 01	2.06E + 01	2.18E-04	6.38E + 00	2.32E-12
	Average	0.00E + 00	1.79E-15	1.82E-20	1.95E + 01	1.44E-04	4.97E + 00	1.61E-12
	Best	0.00E + 00	8.89E-16	1.73E + 01	1.62E + 01	9.61E-05	4.37E + 00	2.93E-13
	STD	0.00E + 00	1.79E-15	7.77E-01	2.20E + 00	5.22E-05	9.42E-01	8.96E-13
	Rank	1	3	2	12	7	9	5
F11	Worst	0.00E + 00	0.00E + 00	1.63E + 02	2.21E + 01	7.68E-02	1.33E + 02	1.06E-02
	Average	0.00E + 00	0.00E + 00	0.00E + 00	1.46E + 01	4.71E-02	1.17E + 02	2.64E-03
	Best	0.00E + 00	0.00E + 00	7.03E + 01	2.58E + 00	5.22E-06	8.97E + 01	0.00E + 00
	STD	0.00E + 00	0.00E + 00	4.16E + 01	8.81E + 00	3.30E-02	1.94E + 01	5.28E-03
	Rank	1	1	1	9	8	11	7
F12	Worst	2.45E-07	1.37E + 00	1.94E + 07	9.21E + 07	7.40E-01	2.29E + 01	2.02E-01
	Average	1.53E-07	1.27E + 00	1.02E + 00	6.45E + 07	6.13E-01	1.44E + 01	1.38E-01
	Best	1.01E-07	9.55E-01	2.10E + 06	1.33E + 07	4.20E-01	9.81E + 00	7.76E-02
	STD	1.25E-09	2.03E-01	7.46E + 06	3.50E + 07	1.53E-01	5.81E + 00	5.48E-02
	Rank	2	8	7	13	6	9	3
F13	Worst	0.00E + 00	5.00E + 00	4.00E + 07	2.12E + 08	4.62E + 00	1.24E + 03	4.63E + 00
	Average	0.00E + 00	4.49E + 00	1.20E + 00	9.36E + 07	4.05E + 00	3.38E + 02	4.19E + 00
	Best	0.00E + 00	4.49E + 00	7.59E + 06	1.91E + 07	3.59E + 00	3.22E + 01	3.69E + 00
	STD	0.00E + 00	1.28E-03	1.34E + 07	8.26E + 07	4.91E-01	5.99E + 02	4.07E-01
	Rank	1	8	3	11	5	9	7
Total		9	2	2	0	0	0	2

Fun	Measure	GOA	WOA	ALO	EO	DA	CMA-ES
F1	Worst	7.80E + 03	5.58E-42	1.11E + 04	4.08E-21	3.64E + 03	2.92E-06
	Average	6.72E + 03	1.40E-42	6.73E + 03	1.09E-21	3.02E + 03	2.12E-06
	Best	4.78E + 03	4.56E-58	3.74E + 03	3.21E-23	2.48E + 03	1.04E-06
	STD	1.36E + 03	2.79E-42	3.11E + 03	2.00E-21	5.52E + 02	8.48E-07
	Rank	12	4	13	6	10	8
F2	Worst	4.13E + 24	9.37E-36	3.83E + 02	1.30E-11	1.23E + 02	2.50E-01
	Average	1.04E + 24	3.66E-36	2.16E + 02	9.37E-12	8.00E + 01	2.14E-01
	Best	5.80E + 02	2.97E-43	1.44E + 02	5.37E-12	5.45E + 01	1.90E-01
	STD	2.08E + 24	4.07E-36	1.12E + 02	3.38E-12	3.03E + 01	2.71E-02
	Rank	13	4	12	6	10	8
F3	Worst	8.16E + 04	5.53E + 05	1.67E + 05	2.16E + 02	1.64E + 05	6.54E + 00
	Average	7.19E + 04	3.89E + 05	1.20E + 05	6.02E + 01	1.28E + 05	3.25E + 00
	Best	5.74E + 04	3.11E + 05	8.76E + 04	1.83E + 00	8.91E + 04	1.71E + 00
	STD	1.03E + 04	1.11E + 05	3.72E + 04	1.04E + 02	3.89E + 04	2.27E + 00
	Rank	9	13	11	5	12	3
F4	Worst	7.04E + 08	7.98E + 08	8.89E + 08	6.87E + 08	6.80E + 08	9.03E + 08
	Average	6.51E + 08	6.98E + 08	7.61E + 08	6.18E + 08	5.59E + 08	7.87E + 08
	Best	5.58E + 08	6.49E + 08	6.73E + 08	4.40E + 08	4.47E + 08	5.91E + 08
	STD	6.40E + 07	6.78E + 07	9.23E + 07	1.19E + 08	9.54E + 07	1.39E + 08
	Rank	8	10	11	5	4	12
F5	Worst	7.21E + 07	4.88E + 01	2.54E + 07	4.87E + 01	9.10E + 07	1.46E + 02
	Average	5.04E + 07	4.88E + 01	1.65E + 07	4.80E + 01	4.29E + 07	6.90E + 01
	Best	3.80E + 07	4.88E + 01	5.52E + 06	4.72E + 01	7.77E + 06	4.29E + 01
	STD	1.50E + 07	2.87E-02	8.22E + 06	7.60E-01	3.70E + 07	5.13E + 01
	Rank	13	5	10	3	12	8

Table 6 (continued)

Fun	Measure	GOA	WOA	ALO	EO	DA	CMA-ES
F6	Worst	2.86E + 04	8.39E + 00	2.90E + 04	6.79E + 00	3.52E + 04	1.81E-06
	Average	2.03E + 04	6.55E + 00	2.41E + 04	6.40E + 00	2.37E + 04	1.48E-06
	Best	1.42E + 04	5.44E + 00	2.17E + 04	5.69E + 00	1.36E + 04	8.62E-07
	STD	6.06E + 03	1.35E + 00	3.34E + 03	4.80E-01	9.34E + 03	4.30E-07
	Rank	11	5	13	4	12	1
F7	Worst	4.04E + 02	3.08E-02	3.15E + 01	1.71E-02	3.05E + 01	6.33E-03
	Average	3.00E + 02	1.62E-02	1.88E + 01	6.79E-03	1.52E + 01	5.88E-03
	Best	2.48E + 02	7.29E-03	9.03E + 00	1.09E-03	1.90E + 00	5.22E-03
	STD	7.02E + 01	1.09E-02	1.03E + 01	7.13E-03	1.18E + 01	4.97E-04
	Rank	13	6	11	5	9	4
F8	Worst	− 8.26E + 03	− 8.61E + 03	− 9.04E + 03	− 7.93E + 03	− 4.21E + 03	− 5.79E + 12
	Average	− 9.31E + 03	− 1.25E + 04	− 9.04E + 03	− 1.09E + 04	− 5.68E + 03	− 2.55E + 17
	Best	− 9.86E + 03	− 1.51E + 04	− 9.04E + 03	− 1.30E + 04	− 6.62E + 03	− 1.00E + 18
	STD	7.24E + 02	3.00E + 03	0.00E + 00	2.10E + 03	1.13E + 03	4.99E + 17
	Rank	6	3	7	5	10	1
F9	Worst	6.92E + 02	0.00E + 00	4.37E + 02	9.98E-01	5.32E + 02	3.34E + 02
	Average	6.02E + 02	0.00E + 00	3.80E + 02	4.99E-01	4.51E + 02	3.23E + 02
	Best	5.22E + 02	0.00E + 00	2.95E + 02	1.14E-13	3.09E + 02	3.08E + 02
	STD	7.00E + 01	0.00E + 00	6.74E + 01	5.76E-01	9.89E + 01	1.13E + 01
	Rank	13	1	11	6	12	9
F10	Worst	2.07E + 01	1.48E-13	1.79E + 01	3.53E-08	1.89E + 01	4.16E-04
	Average	2.01E + 01	5.07E-14	1.27E + 01	1.77E-08	1.55E + 01	2.99E-04
	Best	1.94E + 01	4.45E-15	1.73E + 00	8.24E-09	1.27E + 01	2.51E-04
	STD	5.44E-01	6.57E-14	7.42E + 00	1.21E-08	2.56E + 00	7.84E-05
	Rank	13	4	10	6	11	8
F11	Worst	2.17E + 02	0.00E + 00	2.57E + 02	8.00E-15	1.28E + 02	1.67E-05
	Average	1.53E + 02	0.00E + 00	2.26E + 02	2.48E-15	9.05E + 01	1.61E-05
	Best	1.13E + 02	0.00E + 00	2.05E + 02	0.00E + 00	4.30E + 01	1.47E-05
	STD	4.45E + 01	0.00E + 00	2.47E + 01	3.76E-15	3.79E + 01	9.96E-07
	Rank	12	1	13	5	10	6
F12	Worst	7.29E + 07	7.04E-01	4.71E + 07	3.58E-01	7.48E + 05	5.42E-08
	Average	3.61E + 07	4.79E-01	1.57E + 07	3.02E-01	4.76E + 05	4.07E-08
	Best	1.69E + 07	2.25E-01	2.12E + 06	2.59E-01	1.68E + 01	2.58E-08
	STD	2.57E + 07	2.38E-01	2.13E + 07	4.59E-02	3.38E + 05	1.22E-08
	Rank	12	5	11	4	10	1
F13	Worst	1.77E + 08	4.45E + 00	1.78E + 08	4.30E + 00	5.52E + 07	1.29E-06
	Average	1.27E + 08	3.75E + 00	9.67E + 07	4.09E + 00	1.91E + 07	1.12E-06
	Best	7.40E + 07	3.21E + 00	2.65E + 07	3.79E + 00	4.13E + 06	8.30E-07
	STD	4.41E + 07	5.45E-01	7.71E + 07	2.19E-01	2.43E + 07	2.13E-07
	Rank	13	4	12	6	10	2
Total		0	2	0	0	0	3

Table 7 Comparison between the performance of RLRSA and other methods on unimodal and multimodal functions with 100 dimensions

Fun	Measure	RLRSA	RSA	LICRSA	SCA	GWO	PSO	MPA
F1	Worst	0.000E + 00	2.59E − 157	2.050E + 04	1.580E + 04	1.05E − 06	1.400E + 02	2.60E − 21
	Average	0.000E + 00	6.490E-158	1.290E-160	9.450E + 03	5.200E-07	1.040E + 02	1.570E-21
	Best	0.000E + 00	9.30E − 165	1.760E + 04	4.830E + 03	2.14E − 07	7.620E + 01	3.73E − 22
	STD	0.000E + 00	1.30E − 157	1.200E + 03	4.980E + 03	3.65E − 07	2.920E + 01	1.04E − 21
	Rank	1	3	2	10	7	9	5
F2	Worst	3.270E-85	2.67E − 83	2.970E + 06	3.750E + 00	1.64E − 04	7.780E + 01	4.22E − 13
	Average	2.120E-85	7.240E-84	7.124E-85	2.150E + 00	9.650E-05	6.060E + 01	1.570E-13
	Best	1.130E-92	4.16E − 89	1.440E + 02	6.80E − 01	4.55E − 05	4.200E + 01	5.08E − 14
	STD	2.470E-85	1.31E − 83	1.490E + 06	1.620E + 00	5.33E − 05	1.870E + 01	1.78E − 13
	Rank	1	3	2	9	7	10	5
F3	Worst	7.340E-93	1.29E − 91	6.600E + 04	9.570E + 04	6.850E + 02	1.580E + 04	1.850E + 00
	Average	5.340E-94	3.220E-92	1.236E-93	6.790E + 04	4.620E + 02	1.250E + 04	4.950E-01
	Best	6.450E-120	3.71E − 116	4.620E + 04	4.040E + 04	1.410E + 02	1.080E + 04	5.13E − 07
	STD	2.650E-101	6.43E − 92	8.620E + 03	2.820E + 04	2.300E + 02	2.350E + 03	9.00E − 01
	Rank	1	3	2	10	7	8	4
F4	Worst	9.520E-81	4.43E − 71	6.970E + 01	8.980E + 01	1.490E + 00	2.850E + 01	1.10E − 08
	Average	3.210E-81	1.120E-71	1.532E-74	7.960E + 01	1.180E + 00	2.490E + 01	5.980E-09
	Best	2.520E-84	1.24E − 81	5.280E + 01	6.910E + 01	8.82E − 01	2.200E + 01	2.90E − 09
	STD	6.520E-80	2.22E − 71	8.290E + 00	8.660E + 00	3.13E − 01	3.300E + 00	3.72E − 09
	Rank	1	3	2	12	7	8	4
F5	Worst	2.520E + 01	4.910E + 01	3.340E + 07	3.130E + 07	4.890E + 01	3.670E + 04	4.880E + 01
	Average	6.036E + 01	4.910E + 01	3.420E + 01	1.670E + 07	4.880E + 01	2.810E + 04	4.850E + 01
	Best	1.920E + 01	4.910E + 01	3.820E + 06	7.660E + 05	4.870E + 01	1.930E + 04	4.800E + 01
	STD	6.320E-04	8.26E − 03	1.420E + 07	1.680E + 07	1.13E − 01	9.630E + 03	3.75E − 01
	Rank	8	7	1	11	5	9	4
F6	Worst	0.000E + 00	1.220E + 01	1.420E + 04	5.280E + 03	7.370E + 00	1.340E + 02	5.190E + 00
	Average	0.000E + 00	1.210E + 01	1.052E + 03	3.030E + 03	7.000E + 00	6.400E + 01	4.060E + 00
	Best	0.000E + 00	1.190E + 01	7.120E + 03	4.940E + 02	5.970E + 00	3.090E + 01	2.800E + 00
	STD	0.000E + 00	1.63E − 01	3.400E + 03	1.990E + 03	6.85E − 01	4.770E + 01	1.120E + 00
	Rank	1	7	9	10	6	8	3
F7	Worst	0.000E + 00	1.70E − 03	1.650E + 01	2.250E + 01	2.77E − 02	2.620E + 02	4.57E − 03
	Average	8.120E-03	5.640E-04	1.120E + 01	1.410E + 01	1.980E-02	9.590E + 01	2.130E-03
	Best	0.000E + 00	1.39E − 04	8.750E + 00	5.790E + 00	1.29E − 02	3.180E + 01	3.03E − 04
	STD	0.000E + 00	7.54E − 04	3.560E + 00	7.500E + 00	7.10E − 03	1.120E + 02	1.79E − 03
	Rank	1	2	8	9	7	12	3
F8	Worst	− 8.45E + 03	− 9.53E + 03	− 8.16E + 03	− 3.47E + 03	− 6.69E + 03	− 2.78E + 03	− 1.05E + 04
	Average	− 1.230E + 18	− 1.040E + 04	− 8.590E + 03	− 3.900E + 03	− 7.860E + 03	− 3.930E + 03	− 1.150E + 04
	Best	− 2.18E + 04	− 1.23E + 04	− 9.07E + 03	− 4.09E + 03	− 8.87E + 03	− 5.86E + 03	− 1.32E + 04
	STD	1.150E + 04	1.250E + 03	4.100E + 02	2.890E + 02	1.170E + 03	1.350E + 03	1.270E + 03
	Rank	1	4	9	13	10	12	3
F9	Worst	0.000E + 00	0.000E + 00	4.040E + 02	1.720E + 02	1.800E + 01	3.400E + 02	0.000E + 00
	Average	0.000E + 00	0.000E + 00	0.000E + 00	7.290E + 01	1.410E + 01	2.950E + 02	0.000E + 00
	Best	0.000E + 00	0.000E + 00	3.280E + 02	2.660E + 01	8.790E + 00	2.480E + 02	0.000E + 00
	STD	0.000E + 00	0.000E + 00	3.620E + 01	6.620E + 01	4.180E + 00	3.790E + 01	0.000E + 00
	Rank	1	1	1	8	7	9	1
F10	Worst	0.000E + 00	8.890E-16	3.410E + 00	4.350E-01	7.860E-10	1.940E-02	2.570E-14
	Average	0.000E + 00	8.890E-16	7.230E-16	1.110E-01	4.160E-10	9.460E-03	9.780E-15
	Best	0.000E + 00	8.890E-16	1.170E + 00	6.620E-04	6.490E-11	3.390E-04	4.420E-15
	STD	0.000E + 00	0.000E + 00	1.080E + 00	2.170E-01	3.230E-10	8.030E-03	1.080E-14

Table 7 (continued)

Fun	Measure	RLRSA	RSA	LICRSA	SCA	GWO	PSO	MPA
	Rank	1	3	2	10	8	9	5
F11	Worst	0.000E + 00	0.000E + 00	3.300E-01	6.760E-01	5.950E-02	3.110E + 01	3.960E-02
	Average	0.000E + 00	0.000E + 00	0.000E + 00	3.810E-01	2.760E-02	9.090E + 00	1.680E-02
	Best	0.000E + 00	0.000E + 00	7.160E-02	4.420E-04	1.120E-16	5.450E-01	0.000E + 00
	STD	0.000E + 00	0.000E + 00	1.060E-01	3.320E-01	3.210E-02	1.480E + 01	2.000E-02
	Rank	1	1	1	10	7	13	6
F12	Worst	0.000E + 00	1.090E + 00	9.950E + 00	3.250E-01	6.440E-02	4.240E-04	2.530E-03
	Average	0.000E + 00	3.840E-01	4.600E + 00	2.310E-01	5.640E-02	1.350E-04	7.870E-04
	Best	0.000E + 00	1.630E-01	1.240E + 00	1.510E-01	4.140E-02	3.610E-06	3.900E-10
	STD	0.000E + 00	5.990E-01	3.810E + 00	7.680E-02	1.050E-02	1.970E-04	1.200E-03
	Rank	1	9	10	8	6	3	4
F13	Worst	0.000E + 00	9.700E-01	3.070E-01	6.840E-01	6.020E-01	2.140E-02	1.190E-01
	Average	0.000E + 00	8.130E-02	1.660E-01	5.470E-01	3.810E-01	5.480E-03	3.740E-02
	Best	0.000E + 00	6.110E-02	1.710E-02	4.610E-01	2.160E-01	1.720E-05	2.740E-04
	STD	0.000E + 00	1.500E-01	1.200E-01	9.620E-02	1.660E-01	1.070E-02	5.450E-02
	Rank	1	5	6	10	8	3	4
Total		11	2	3	0	0	0	1

Fun	Measure	GOA	WOA	ALO	EO	DA	CMA-ES
F1	Worst	3.980E + 04	9.73E − 41	5.780E + 04	2.31E − 17	1.730E + 04	1.29E − 02
	Average	3.520E + 04	2.440E-41	3.490E + 04	6.220E-18	1.190E + 04	9.480E-03
	Best	3.130E + 04	2.75E − 47	1.800E + 04	2.10E − 19	7.080E + 03	5.89E − 03
	STD	3.660E + 03	4.87E − 41	2.020E + 04	1.13E − 17	5.300E + 03	3.18E − 03
	Rank	13	4	12	6	11	8
F2	Worst	1.160E + 21	1.02E − 20	2.130E + 02	5.30E − 09	2.020E + 02	5.15E − 03
	Average	2.870E + 20	2.540E-21	2.020E + 02	3.360E-09	1.480E + 02	3.330E-03
	Best	2.290E + 07	6.12E − 30	1.830E + 02	1.10E − 09	7.490E + 01	2.31E − 03
	STD	5.740E + 20	5.07E − 21	1.340E + 01	1.86E − 09	5.820E + 01	1.27E − 03
	Rank	13	4	12	6	11	8
F3	Worst	8.730E + 04	4.800E + 05	1.790E + 05	6.000E + 01	1.990E + 05	6.600E + 00
	Average	6.720E + 04	3.830E + 05	1.310E + 05	1.800E + 01	1.290E + 05	3.880E + 00
	Best	5.430E + 04	3.050E + 05	9.120E + 04	4.41E − 02	5.600E + 04	4.08E − 01
	STD	1.460E + 04	7.680E + 04	3.910E + 04	2.860E + 01	5.800E + 04	2.800E + 00
	Rank	9	13	12	6	11	5
F4	Worst	6.610E + 01	9.830E + 01	7.100E + 01	7.87E − 03	5.540E + 01	2.08E − 02
	Average	6.280E + 01	8.550E + 01	5.550E + 01	4.210E-03	4.400E + 01	1.630E-02
	Best	5.550E + 01	5.470E + 01	3.520E + 01	4.24E − 04	3.300E + 01	1.28E − 02
	STD	4.960E + 00	2.070E + 01	1.500E + 01	3.20E − 03	9.380E + 00	3.37E − 03
	Rank	11	13	10	5	9	6
F5	Worst	1.390E + 08	4.900E + 01	3.130E + 07	4.810E + 01	1.700E + 07	4.420E + 01
	Average	9.120E + 07	4.890E + 01	1.950E + 07	4.800E + 01	9.540E + 06	4.350E + 01
	Best	4.420E + 07	4.880E + 01	8.890E + 06	4.760E + 01	2.220E + 05	4.280E + 01
	STD	4.460E + 07	6.85E − 02	9.280E + 06	2.61E − 01	8.690E + 06	5.94E − 01
	Rank	13	6	12	3	10	2
F6	Worst	2.580E + 04	7.320E + 00	4.260E + 04	6.230E + 00	3.360E + 04	2.92E − 06
	Average	1.970E + 04	6.920E + 00	2.790E + 04	5.910E + 00	1.630E + 04	2.120E-06
	Best	1.300E + 04	6.340E + 00	2.000E + 04	5.490E + 00	5.280E + 03	1.24E − 06
	STD	5.290E + 03	4.15E − 01	1.050E + 04	3.52E − 01	1.360E + 04	7.54E − 07
	Rank	12	5	13	4	11	2

Table 7 (continued)

Fun	Measure	GOA	WOA	ALO	EO	DA	CMA-ES
F7	Worst	4.850E + 02	1.80E − 02	3.100E + 01	9.20E − 03	4.370E + 01	1.08E − 02
	Average	3.680E + 02	9.310E-03	1.800E + 01	6.440E-03	1.930E + 01	8.130E-03
	Best	2.420E + 02	4.11E − 03	7.190E + 00	5.12E − 03	8.200E + 00	5.91E − 03
	STD	1.190E + 02	6.27E − 03	1.090E + 01	1.88E − 03	1.650E + 01	2.01E − 03
	Rank	13	6	10	4	11	5
F8	Worst	− 8.99E + 03	− 7.84E + 03	− 9.04E + 03	− 8.44E + 03	− 3.63E + 03	− 1.31E + 14
	Average	− 9.880E + 03	− 8.880E + 03	− 9.060E + 03	− 1.020E + 04	− 5.110E + 03	− 4.160E + 17
	Best	− 1.06E + 04	− 9.98E + 03	− 9.14E + 03	− 1.09E + 04	− 6.37E + 03	− 1.14E + 18
	STD	6.560E + 02	5.120E + 02	4.830E + 01	1.160E + 03	1.290E + 03	5.390E + 17
	Rank	6	8	7	5	11	2
F9	Worst	7.090E + 02	0.000E + 00	3.750E + 02	2.100E + 00	5.690E + 02	3.340E + 02
	Average	6.640E + 02	0.000E + 00	3.380E + 02	5.240E-01	4.520E + 02	3.180E + 02
	Best	6.310E + 02	0.000E + 00	2.950E + 02	5.690E-14	3.850E + 02	2.950E + 02
	STD	3.200E + 01	0.000E + 00	4.030E + 01	1.060E + 00	8.210E + 01	1.680E + 01
	Rank	13	1	11	6	12	10
F10	Worst	2.000E + 01	2.940E-14	1.640E + 01	1.650E-13	8.470E + 00	8.890E-16
	Average	1.950E + 01	1.160E-14	9.820E + 00	1.040E-13	7.780E + 00	8.890E-16
	Best	1.910E + 01	8.890E-16	2.330E + 00	4.360E-14	7.100E + 00	8.890E-16
	STD	4.010E-01	1.270E-14	5.760E + 00	5.400E-14	6.810E-01	0.000E + 00
	Rank	13	6	12	7	11	3
F11	Worst	5.890E-01	3.020E-01	4.610E-01	2.030E-02	1.510E + 01	1.150E-02
	Average	5.420E-01	1.330E-01	2.310E-01	5.070E-03	5.960E + 00	2.850E-03
	Best	4.510E-01	0.000E + 00	1.140E-01	0.000E + 00	2.650E + 00	0.000E + 00
	STD	6.180E-02	1.570E-01	1.570E-01	1.020E-02	6.070E + 00	5.700E-03
	Rank	11	8	9	5	12	4
F12	Worst	5.320E + 04	3.550E-01	4.330E + 01	2.000E-02	4.500E + 06	4.720E-32
	Average	1.350E + 04	2.010E-01	2.450E + 01	7.280E-03	1.130E + 06	4.720E-32
	Best	2.230E + 01	1.390E-01	1.370E + 01	4.820E-05	1.030E + 01	4.720E-32
	STD	2.660E + 04	1.040E-01	1.360E + 01	9.440E-03	2.250E + 06	0.000E + 00
	Rank	12	7	11	5	13	2
F13	Worst	3.710E + 04	7.320E-01	3.180E + 01	7.720E-01	8.420E + 05	1.360E-32
	Average	1.500E + 04	4.500E-01	1.320E + 01	2.900E-01	2.410E + 05	1.360E-32
	Best	5.810E + 01	1.550E-01	2.000E + 00	9.940E-02	2.510E + 03	1.360E-32
	STD	1.830E + 04	3.230E-01	1.300E + 01	3.240E-01	4.040E + 05	0.000E + 00
	Rank	12	9	11	7	13	2
Total		0	1	0	0	0	0

6.4 Real-world application

This section addresses two engineering design problems, including pressure vessel design problems and tension and compressing spring problem. The RLRSA method is applied to solve this optimization problem. The results of RLRSA are compared to various optimization algorithms from the literature.

This section focuses on two specific engineering design problems: pressure vessel design and tension/compression spring problems. To tackle these optimization challenges, this work applies the RLRSA method. We then compare the performance of RLRSA with several optimization algorithms found in the literature. This comparison allows us to assess the effectiveness and efficiency of RLRSA in solving these design problems.

6.4.1 pressure vessel design problem (PVD)

To evaluate the performance of the proposed RLRSA, one of the complex engineering problems known as pressure vessel design problem (PVD) is shown in Fig. 5. The parameter settings for all methods are the same as the above experiment. The PVD problem has four variables including the thickness of the shell (Ts), the inner radius R, the length of cylindrical section of the vessel (L), and the

Table 8 Comparison between the performance of RLRSA and other methods on unimodal and multimodal functions with 500 dimensions

Fun	Measure	RLRSA	RSA	LICRSA0	SCA	GWO	PSO	MPA
F1	Worst	2.31E − 153	1.92E − 150	9.460E + 04	2.120E + 04	3.04E − 03	2.290E + 03	2.63E − 22
	Average	5.380E-152	4.790E-151	1.250E-25	1.560E + 04	1.690E-03	1.680E + 03	1.350E-22
	Best	2.79E − 171	2.70E − 162	5.900E + 04	1.100E + 04	7.70E − 04	1.240E + 03	4.73E − 24
	STD	3.25E − 161	9.56E − 151	1.470E + 04	5.200E + 03	9.86E − 04	4.550E + 02	1.07E − 22
	Rank	1	2	3	10	7	9	5
F2	Worst	0.000E + 00	6.29E − 77	1.340E + 171	2.320E + 02	2.960E + 00	1.630E + 161	1.95E − 10
	Average	0.000E + 00	1.610E-77	3.330E + 170	1.790E + 02	1.830E + 00	4.060E + 160	1.070E-10
	Best	0.000E + 00	1.13E − 83	1.700E + 132	1.360E + 02	3.97E − 01	1.050E + 67	1.39E − 11
	STD	0.000E + 00	3.13E − 77	6.510E + 04	4.400E + 01	1.150E + 00	6.560E + 04	9.53E − 11
	Rank	1	2	12	8	6	11	4
F3	Worst	2.57E − 78	1.58E − 73	8.520E + 06	1.240E + 07	1.060E + 06	2.720E + 06	8.780E + 04
	Average	4.980E + 02	5.000E + 02	3.290E + 02	2.550E + 09	6.550E + 05	5.520E + 08	4.990E + 02
	Best	1.74E − 103	1.85E − 95	2.800E + 06	6.300E + 06	7.390E + 05	7.860E + 05	1.520E + 04
	STD	3.23E − 85	7.84E − 74	2.360E + 06	2.610E + 06	1.480E + 05	8.790E + 05	3.930E + 04
	Rank	1	2	7	11	5	6	3
F4	Worst	0.000E + 00	1.88E − 67	9.770E + 01	9.980E + 01	9.100E + 01	6.340E + 01	1.22E − 05
	Average	0.000E + 00	4.930E-68	9.720E + 01	9.970E + 01	8.070E + 01	5.840E + 01	3.580E-06
	Best	0.000E + 00	4.27E − 71	9.650E + 01	9.960E + 01	7.130E + 01	5.540E + 01	2.97E − 07
	STD	0.000E + 00	9.20E − 68	5.56E − 01	1.16E − 01	1.020E + 01	3.520E + 00	5.71E − 06
	Rank	1	2	11	13	7	5	3
F5	Worst	3.590E + 01	5.000E + 02	4.930E + 09	3.220E + 09	1.080E + 06	5.880E + 08	5.000E + 02
	Average	3.590E + 01	5.000E + 02	4.390E + 09	2.550E + 09	6.550E + 05	5.520E + 08	4.990E + 02
	Best	3.890E + 01	5.000E + 02	3.980E + 09	1.780E + 09	1.310E + 05	4.520E + 08	4.990E + 02
	STD	3.05E − 04	5.10E − 03	4.040E + 08	6.140E + 08	3.980E + 05	6.680E + 07	8.99E − 02
	Rank	2	5	1	13	8	9	3
F6	Worst	1.530E + 02	1.260E + 02	1.060E + 06	2.700E + 05	1.640E + 02	2.310E + 05	1.090E + 02
	Average	1.390E + 02	1.250E + 02	9.540E + 05	2.530E + 05	1.460E + 02	2.280E + 05	1.080E + 02
	Best	1.250E + 02	1.250E + 02	8.600E + 05	2.380E + 05	1.210E + 02	2.230E + 05	1.070E + 02
	STD	2.300E + 01	1.98E − 01	9.820E + 04	1.640E + 04	2.250E + 01	4.300E + 03	1.450E + 00
	Rank	5	4	12	9	6	8	2
F7	Worst	0.000E + 00	3.61E − 04	1.880E + 04	2.080E + 04	9.86E − 01	6.190E + 04	8.10E − 03
	Average	0.000E + 00	1.800E-04	1.180E + 04	1.770E + 04	7.120E-01	5.850E + 04	4.490E-03
	Best	0.000E + 00	3.86E − 05	8.660E + 03	1.530E + 04	5.29E − 01	5.240E + 04	2.47E − 03
	STD	0.000E + 00	1.52E − 04	4.710E + 03	2.750E + 03	2.19E − 01	4.310E + 03	2.50E − 03
	Rank	1	2	10	11	7	13	3
F8	Worst	0.000E + 00	− 7.47E + 04	− 2.68E + 04	− 1.22E + 04	− 1.03E + 04	− 9.57E + 03	− 5.11E + 04
	Average	0.000E + 00	− 7.87E + 04	− 3.26E + 04	− 1.34E + 04	− 2.92E + 04	− 1.20E + 04	− 5.54E + 04
	Best	0.000E + 00	− 8.11E + 04	− 3.74E + 04	− 1.46E + 04	− 3.64E + 04	− 1.57E + 04	− 5.97E + 04
	STD	0.000E + 00	2.790E + 03	4.720E + 03	1.050E + 03	1.270E + 04	2.860E + 03	4.730E + 03
	Rank	1	4	9	12	10	13	6
F9	Worst	0.000E + 00	0.000E + 00	6.010E + 03	1.490E + 03	3.590E + 02	8.360E + 03	0.000E + 00
	Average	0.000E + 00	0.000E + 00	1.250E-10	1.280E + 03	2.720E + 02	7.830E + 03	0.000E + 00
	Best	0.000E + 00	0.000E + 00	5.880E + 03	1.150E + 03	1.220E + 02	7.270E + 03	0.000E + 00
	STD	0.000E + 00	0.000E + 00	5.470E + 01	1.450E + 02	1.070E + 02	5.320E + 02	0.000E + 00
	Rank	1	1	5	8	7	12	1

Table 8 (continued)

Fun	Measure	RLRSA	RSA	LICRSA0	SCA	GWO	PSO	MPA
F10	Worst	0.000E + 00	8.88E − 16	2.090E + 01	2.100E + 01	2.530E + 00	1.900E + 01	3.78E − 09
	Average	0.000E + 00	8.880E-16	0.000E + 00	2.100E + 01	2.150E + 00	1.870E + 01	2.430E-09
	Best	0.000E + 00	8.88E − 16	2.030E + 01	2.100E + 01	1.490E + 00	1.840E + 01	1.42E − 09
	STD	0.000E + 00	0.000E + 00	2.61E − 01	2.18E − 02	4.58E − 01	2.53E − 01	1.02E − 09
	Rank	1	3	1	13	7	11	5
F11	Worst	0.000E + 00	0.000E + 00	6.660E + 03	2.260E + 03	1.350E + 00	1.980E + 03	0.000E + 00
	Average	0.000E + 00	0.000E + 00	6.200E-10	1.660E + 03	1.150E + 00	1.900E + 03	0.000E + 00
	Best	0.000E + 00	0.000E + 00	6.160E + 03	7.960E + 02	9.20E − 01	1.820E + 03	0.000E + 00
	STD	0.000E + 00	0.000E + 00	2.420E + 02	6.270E + 02	1.72E − 01	6.680E + 01	0.000E + 00
	Rank	1	1	5	9	7	10	1
F12	Worst	0.000E + 00	0.000E + 00	6.170E + 03	3.370E + 03	1.290E + 00	2.070E + 03	0.000E + 00
	Average	0.000E + 00	0.000E + 00	1.200E-18	2.240E + 03	1.020E + 00	1.940E + 03	0.000E + 00
	Best	0.000E + 00	0.000E + 00	5.260E + 03	1.310E + 03	3.68E − 01	1.840E + 03	0.000E + 00
	STD	0.000E + 00	0.000E + 00	3.840E + 02	8.870E + 02	4.32E − 01	9.740E + 01	0.000E + 00
	Rank	1	1	4	11	7	9	1
F13	Worst	2.300E + 02	5.000E + 01	8.630E + 09	1.070E + 10	3.070E + 02	2.840E + 08	4.990E + 01
	Average	1.990E + 02	5.000E + 01	7.840E + 09	8.830E + 09	2.050E + 02	2.650E + 08	4.990E + 01
	Best	1.850E + 02	5.000E + 01	7.280E + 09	6.570E + 09	1.060E + 02	2.460E + 08	4.990E + 01
	STD	1.420E + 01	1.190E + 03	6.350E + 08	1.730E + 09	8.390E + 01	1.560E + 07	3.72E − 02
	Rank	12	13	7	8	3	10	1
Total		10	3	2	0	0	0	4

Fun	Measure	GOA	WOA	ALO	EO	DA	CMA-ES
F1	Worst	9.200E + 04	9.50E − 24	1.260E + 05	4.24E − 12	4.820E + 04	1.83E − 02
	Average	8.900E + 04	2.680E-24	9.070E + 04	1.540E-12	3.750E + 04	1.250E-02
	Best	8.200E + 04	1.49E − 37	5.930E + 04	1.38E − 13	2.750E + 04	9.98E − 03
	STD	4.710E + 03	4.59E − 24	3.080E + 04	1.92E − 12	8.710E + 03	3.87E − 03
	Rank	12	4	13	6	11	8
F2	Worst	1.720E + 137	3.94E − 21	2.700E + 247	5.09E − 06	1.250E + 03	9.250E + 01
	Average	4.320E + 136	1.190E-21	6.730E + 246	3.810E-06	1.050E + 03	8.490E + 01
	Best	2.260E + 103	6.42E − 25	2.070E + 03	2.77E − 06	7.170E + 02	7.950E + 01
	STD	8.620E + 136	1.88E − 21	6.560E + 04	1.13E − 06	2.260E + 02	5.920E + 00
	Rank	10	3	13	5	9	7
F3	Worst	7.650E + 06	1.450E + 08	1.060E + 07	2.020E + 05	1.710E + 07	9.860E + 06
	Average	2.420E + 09	4.990E + 02	2.210E + 09	5.000E + 02	1.170E + 09	8.000E + 04
	Best	3.400E + 06	3.390E + 07	4.820E + 06	3.620E + 04	5.980E + 06	5.900E + 06
	STD	1.880E + 06	5.720E + 07	2.400E + 06	9.330E + 04	4.700E + 06	1.670E + 06
	Rank	8	13	10	4	12	9
F4	Worst	9.560E + 01	9.970E + 01	8.840E + 01	9.830E + 01	9.960E + 01	3.590E + 01
	Average	8.980E + 01	9.250E + 01	8.420E + 01	7.560E + 01	9.870E + 01	3.330E + 01
	Best	8.710E + 01	8.000E + 01	7.940E + 01	5.260E + 01	9.690E + 01	3.150E + 01
	STD	3.920E + 00	8.770E + 00	4.420E + 00	1.950E + 01	1.240E + 00	1.950E + 00
	Rank	9	10	8	6	12	4
F5	Worst	2.600E + 09	4.990E + 02	3.380E + 09	5.000E + 02	1.820E + 09	1.180E + 05
	Average	2.420E + 09	4.990E + 02	2.210E + 09	5.000E + 02	1.170E + 09	8.000E + 04
	Best	2.300E + 09	4.980E + 02	1.780E + 09	5.000E + 02	6.910E + 08	6.190E + 04
	STD	1.280E + 08	4.22E − 01	7.820E + 08	2.52E − 02	5.340E + 08	2.550E + 04
	Rank	12	3	11	5	10	7

Table 8 (continued)

Fun	Measure	GOA	WOA	ALO	EO	DA	CMA-ES
F6	Worst	8.090E + 05	1.050E + 02	1.100E + 06	1.200E + 02	3.590E + 05	5.010E + 02
	Average	7.620E + 05	1.020E + 02	1.010E + 06	1.190E + 02	3.160E + 05	4.900E + 02
	Best	7.340E + 05	9.870E + 01	8.740E + 05	1.180E + 02	2.320E + 05	4.820E + 02
	STD	4.140E + 04	2.630E + 00	1.150E + 05	9.19E − 01	7.250E + 04	9.400E + 00
	Rank	11	1	13	3	10	7
F7	Worst	5.120E + 04	4.93E − 02	1.400E + 04	3.06E − 02	7.930E + 03	5.02E − 01
	Average	4.740E + 04	3.320E-02	1.030E + 04	1.830E-02	4.070E + 03	4.240E-01
	Best	4.260E + 04	1.31E − 02	5.960E + 03	9.01E − 03	1.220E + 03	3.62E − 01
	STD	3.580E + 03	1.56E − 02	4.140E + 03	1.02E − 02	2.830E + 03	5.91E − 02
	Rank	12	5	9	4	8	6
F8	Worst	− 3.33E + 04	− 1.27E + 05	− 9.04E + 04	− 3.86E + 04	− 1.51E + 04	− 6.07E + 04
	Average	− 3.71E + 04	− 1.40E + 05	− 9.04E + 04	− 4.27E + 04	− 1.71E + 04	− 6.65E + 04
	Best	− 3.96E + 04	− 1.51E + 05	− 9.04E + 04	− 4.59E + 04	− 1.91E + 04	− 7.18E + 04
	STD	2.700E + 03	9.980E + 03	0.000E + 00	3.270E + 03	1.650E + 03	5.670E + 03
	Rank	8	2	3	7	11	5
F9	Worst	8.490E + 03	0.000E + 00	6.190E + 03	1.030E + 00	6.660E + 03	4.880E + 03
	Average	8.170E + 03	0.000E + 00	5.800E + 03	2.560E-01	5.980E + 03	4.790E + 03
	Best	8.050E + 03	0.000E + 00	5.300E + 03	5.07E − 10	5.280E + 03	4.720E + 03
	STD	2.120E + 02	0.000E + 00	3.800E + 02	5.10E − 01	5.820E + 02	5.830E + 01
	Rank	13	1	10	6	11	9
F10	Worst	2.050E + 01	3.93E − 09	1.990E + 01	1.59E − 03	1.920E + 01	4.390E + 00
	Average	2.020E + 01	1.220E-09	1.330E + 01	7.100E-04	1.860E + 01	4.190E + 00
	Best	1.990E + 01	1.12E − 10	8.89E − 16	1.92E − 04	1.740E + 01	3.950E + 00
	STD	2.62E − 01	1.83E − 09	9.090E + 00	6.28E − 04	8.66E − 01	1.80E − 01
	Rank	12	4	9	6	10	8
F11	Worst	6.380E + 03	0.000E + 00	8.290E + 03	1.47E − 09	2.810E + 03	5.010E + 00
	Average	5.730E + 03	0.000E + 00	6.470E + 03	7.460E-10	2.060E + 03	4.900E + 00
	Best	5.300E + 03	0.000E + 00	5.580E + 03	2.63E − 10	1.450E + 03	4.820E + 00
	STD	4.850E + 02	0.000E + 00	1.240E + 03	5.20E − 10	5.650E + 02	8.54E − 02
	Rank	12	1	13	6	11	8
F12	Worst	6.360E + 03	1.12E − 16	6.250E + 03	7.09E − 09	3.250E + 03	5.340E + 00
	Average	5.880E + 03	2.790E-17	5.500E + 03	3.670E-09	2.140E + 03	5.060E + 00
	Best	5.450E + 03	0.000E + 00	4.300E + 03	5.51E − 10	1.240E + 03	4.620E + 00
	STD	3.760E + 02	5.56E − 17	8.620E + 02	2.81E − 09	8.660E + 02	3.62E − 01
	Rank	13	5	12	6	10	8
F13	Worst	7.810E + 09	4.770E + 01	7.840E + 09	4.990E + 01	2.770E + 09	2.160E + 02
	Average	7.210E + 09	3.910E + 01	7.240E + 09	4.970E + 01	1.600E + 09	1.980E + 02
	Best	6.580E + 09	3.020E + 01	6.640E + 09	4.930E + 01	1.130E + 09	1.830E + 02
	STD	6.060E + 08	8.060E + 00	5.800E + 08	2.94E − 01	7.850E + 08	1.400E + 01
	Rank	11	2	9	5	9	5
Total		0	3	0	0	0	0

Table 9 Comparison of RLRSA and other methods on the fixed-dimension multimodal benchmark functions (F14–F23)

Fun	Measure	RLRSA	RSA	LICRSA	SCA	GWO	PSO	MPA
F14	Worst	0.000E + 00	9.990E-01	2.990E + 00	2.990E + 00	1.090E + 01	3.980E + 00	9.990E-01
	Average	0.000E + 00	9.990E-01	1.600E + 00	1.500E + 00	6.300E + 00	2.790E + 00	9.990E-01
	Best	0.000E + 00	9.990E-01	9.990E-01	9.990E-01	2.990E + 00	2.000E + 00	9.990E-01
	STD	0.000E + 00	0.000E + 00	8.890E-01	8.600E-01	4.110E + 00	1.090E + 00	0.000E + 00
	Rank	1	2	7	6	11	8	2
F15	Worst	0.000E + 00	8.380E-04	2.090E-02	1.920E-03	2.050E-02	9.940E-04	3.000E-04
	Average	0.000E + 00	5.770E-04	4.810E-03	1.290E-03	8.460E-03	9.210E-04	3.000E-04
	Best	0.000E + 00	3.650E-04	7.440E-04	6.930E-04	3.090E-04	8.670E-04	3.000E-04
	STD	0.000E + 00	2.250E-04	8.960E-03	4.910E-04	1.100E-02	6.650E-05	2.900E-15
	Rank	1	5	9	8	12	7	3
F16	Worst	− 1.030E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00
	Average	− 1.030E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00
	Best	− 1.030E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00
	STD	1.190E-03	1.200E-03	1.530E-14	2.180E-05	2.810E-08	1.930E-16	0.000E + 00
	Rank	12	1	1	1	1	1	1
F17	Worst	2.280E-01	3.990E-01	0.000E + 00	4.030E-01	3.990E-01	3.990E-01	3.990E-01
	Average	2.310E-01	3.990E-01	0.000E + 00	4.010E-01	3.990E-01	3.990E-01	3.990E-01
	Best	2..20E-01	3.990E-01	0.000E + 00	4.000E-01	3.990E-01	3.990E-01	3.990E-01
	STD	1.120E-03	1.580E-02	0.000E + 00	1.380E-03	1.060E-05	0.000E + 00	4.380E-14
	Rank	4	5	1	13	5	5	5
F18	Worst	2.130E + 00	3.010E + 00	3.010E + 00	3.010E + 00	8.410E + 01	3.010E + 00	3.010E + 00
	Average	2.130E + 00	3.010E + 00	3.010E + 00	3.010E + 00	1.930E + 01	3.010E + 00	3.010E + 00
	Best	2.130E + 00	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00
	STD	1.200E-15	1.760E-12	5.960E-13	1.140E-03	3.630E + 01	1.270E-15	1.140E-15
	Rank	1	2	2	2	13	2	2
F19	Worst	− 3.860E + 00	− 3.870E + 00	− 3.870E + 00	− 3.860E + 00	− 3.870E + 00	− 3.870E + 00	− 3.870E + 00
	Average	− 3.860E + 00	− 3.870E + 00	− 3.870E + 00	− 3.860E + 00	− 3.870E + 00	− 3.870E + 00	− 3.870E + 00
	Best	− 3.860E + 00	− 3.870E + 00	− 3.870E + 00	− 3.860E + 00	− 3.870E + 00	− 3.870E + 00	− 3.870E + 00
	STD	3.370E-01	3.380E-01	2.820E-10	1.330E-03	1.830E-03	3.150E-16	3.150E-16
	Rank	10	1	1	10	1	1	1
F20	Worst	− 1.030E + 00	− 1.040E + 00	− 3.170E + 00	− 2.610E + 00	− 3.090E + 00	− 3.210E + 00	− 3.210E + 00
	Average	− 2.050E + 00	− 2.060E + 00	− 3.250E + 00	− 2.950E + 00	− 3.210E + 00	− 3.260E + 00	− 3.310E + 00
	Best	− 2.630E + 00	− 2.640E + 00	− 3.330E + 00	− 3.080E + 00	− 3.330E + 00	− 3.330E + 00	− 3.330E + 00
	STD	6.340E-01	6.350E-01	7.710E-02	1.900E-01	8.600E-02	6.520E-02	5.330E-02
	Rank	13	12	7	11	10	6	1
F21	Worst	− 1.020E + 01	− 5.070E + 00	− 5.070E + 00	− 4.970E-01	− 1.020E + 01	− 2.640E + 00	− 1.030E + 01
	Average	− 1.002E + 02	− 9.140E + 00	− 6.009E + 01	− 8.950E-01	− 1.020E + 01	− 7.160E + 00	− 1.030E + 01
	Best	− 1.020E + 01	− 1.030E + 01	− 1.030E + 01	− 2.110E + 00	− 1.030E + 01	− 1.030E + 01	− 1.020E + 01
	STD	1.210E + 01	2.290E + 00	2.280E + 00	6.940E-01	3.000E-03	4.120E + 00	1.790E-11
	Rank	1	6	2	12	5	8	3
F22	Worst	− 1.020E + 01	− 5.100E + 00	− 3.730E + 00	− 9.070E− 01	− 1.050E + 01	− 2.760E + 00	− 1.050E + 01
	Average	− 3.602E + 01	− 9.280E + 00	− 2.564E + 01	− 1.910E + 00	− 1.050E + 01	− 6.480E + 00	− 1.050E + 01
	Best	− 1.020E + 01	− 1.050E + 01	− 1.050E + 01	− 4.030E + 00	− 1.050E + 01	− 1.050E + 01	− 1.050E + 01
	STD	0.000E + 00	2.350E + 00	3.320E + 00	1.440E + 00	1.350E-03	3.690E + 00	1.050E-11
	Rank	1	7	2	12	3	10	3
F23	Worst	− 1.050E + 01	− 2.880E + 00	− 1.060E + 01	− 2.470E + 00	− 1.060E + 01	− 1.060E + 01	− 7.900E-01
	Average	− 3.621E + 01	− 6.600E + 00	− 1.060E + 01	− 4.150E + 00	− 1.060E + 01	− 1.060E + 01	− 2.100E + 00
	Best	− 1.050E + 01	− 1.060E + 01	− 1.060E + 01	− 7.280E + 00	− 1.060E + 01	− 1.060E + 01	− 5.140E + 00
	STD	1.200E-01	3.700E + 00	2.110E-11	1.920E + 00	1.990E-03	1.550E-15	1.910E + 00
	Rank	1	9	2	12	2	2	13

Table 9 (continued)

Fun	Measure	RLRSA	RSA	LICRSA	SCA	GWO	PSO	MPA
Total		8	5	4	1	2	2	3

Fun	Measure	GOA	WOA	ALO	EO	DA	CMA-ES
F14	Worst	1.470E + 01	1.090E + 01	5.940E + 00	9.990E-01	2.000E + 00	7.990E + 00
	Average	1.240E + 01	4.150E + 00	3.580E + 00	9.990E-01	1.410E + 00	6.930E + 00
	Best	9.810E + 00	9.990E-01	9.990E-01	9.990E-01	9.990E-01	4.010E + 00
	STD	1.870E + 00	4.240E + 00	2.280E + 00	1.580E-16	5.450E-01	1.670E + 00
	Rank	13	10	9	2	5	12
F15	Worst	2.190E-02	1.530E-03	2.050E-02	3.540E-04	2.050E-02	0.000E + 00
	Average	9.800E-03	7.750E-04	4.960E-03	3.220E-04	5.730E-03	0.000E + 00
	Best	7.860E-04	3.140E-04	7.770E-04	3.090E-04	1.510E-03	0.000E + 00
	STD	1.060E-02	4.570E-04	8.660E-03	1.990E-05	8.200E-03	0.000E + 00
	Rank	13	6	10	4	11	1
F16	Worst	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	0.000E + 00
	Average	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	0.000E + 00
	Best	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	− 1.040E + 00	0.000E + 00
	STD	2.150E-13	1.190E-09	2.940E-13	1.930E-16	4.240E-07	0.000E + 00
	Rank	1	1	1	1	1	13
F17	Worst	0.000E + 00	3.990E-01	3.990E-01	3.990E-01	3.990E-01	0.000E + 00
	Average	0.000E + 00	3.990E-01	3.990E-01	3.990E-01	3.990E-01	0.000E + 00
	Best	0.000E + 00	3.990E-01	3.990E-01	3.980E-01	3.990E-01	0.000E + 00
	STD	0.000E + 00	1.990E-04	1.810E-13	0.000E + 00	1.200E-06	0.000E + 00
	Rank	1	5	5	5	5	1
F18	Worst	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00
	Average	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00
	Best	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00	3.010E + 00
	STD	1.760E-12	1.560E-04	1.550E-12	1.140E-15	4.640E-13	0.000E + 00
	Rank	2	2	2	2	2	2
F19	Worst	− 3.090E + 00	− 3.860E + 00	− 3.870E + 00	− 3.870E + 00	− 3.870E + 00	− 3.880E + 00
	Average	− 3.550E + 00	− 3.860E + 00	− 3.870E + 00	− 3.870E + 00	− 3.870E + 00	− 3.870E + 00
	Best	− 3.870E + 00	− 3.870E + 00	− 3.870E + 00	− 3.870E + 00	− 3.870E + 00	− 3.880E + 00
	STD	3.300E-01	4.470E-03	4.660E-11	6.270E-16	1.490E-03	0.000E + 00
	Rank	13	10	1	1	1	1
F20	Worst	− 3.210E + 00	− 3.100E + 00	− 3.200E + 00	− 3.210E + 00	− 3.190E + 00	− 3.210E + 00
	Average	− 3.310E + 00	− 3.280E + 00	− 3.280E + 00	− 3.310E + 00	− 3.220E + 00	− 3.240E + 00
	Best	− 3.330E + 00	− 3.330E + 00	− 3.330E + 00	− 3.330E + 00	− 3.330E + 00	− 3.330E + 00
	STD	5.570E-02	9.930E-02	6.910E-02	5.330E-02	6.060E-02	5.330E-02
	Rank	1	4	4	1	9	8
F21	Worst	− 2.690E + 00	− 5.060E + 00	− 2.690E + 00	− 2.950E-01	− 2.640E + 00	− 1.030E + 01
	Average	− 6.160E + 00	− 8.080E + 00	− 4.620E + 00	− 4.670E-01	− 6.640E + 00	− 1.030E + 01
	Best	− 1.030E + 01	− 1.020E + 01	− 5.110E + 00	− 7.120E-01	− 1.030E + 01	− 1.030E + 01
	STD	3.790E + 00	2.760E + 00	1.090E + 00	1.600E-01	3.380E + 00	0.000E + 00
	Rank	10	7	11	13	9	3
F22	Worst	− 2.760E + 00	− 2.950E-01	− 3.730E + 00	− 5.100E + 00	− 2.760E + 00	− 1.050E + 01
	Average	− 7.350E + 00	− 5.570E-01	− 7.740E + 00	− 9.350E + 00	− 6.390E + 00	− 1.050E + 01
	Best	− 1.050E + 01	− 8.590E-01	− 1.050E + 01	− 1.050E + 01	− 1.040E + 01	− 1.050E + 01
	STD	4.200E + 00	2.730E-01	3.670E + 00	2.389 + 00	3.560E + 00	0.000E + 00
	Rank	9	13	8	6	11	3
F23	Worst	− 1.690E + 00	− 5.110E + 00	− 3.850E + 00	− 3.850E + 00	− 1.690E + 00	− 1.060E + 01
	Average	− 4.270E + 00	− 9.430E + 00	− 9.210E + 00	− 9.210E + 00	− 6.530E + 00	− 1.060E + 01
	Best	− 1.060E + 01	− 1.060E + 01	− 1.060E + 01	− 1.060E + 01	− 1.060E + 01	− 1.060E + 01
	STD	3.610E + 00	2.430E + 00	3.010E + 00	3.010E + 00	3.750E + 00	0.000E + 00
	Rank	11	6	7	7	10	2
Total		3	1	1	3	2	3

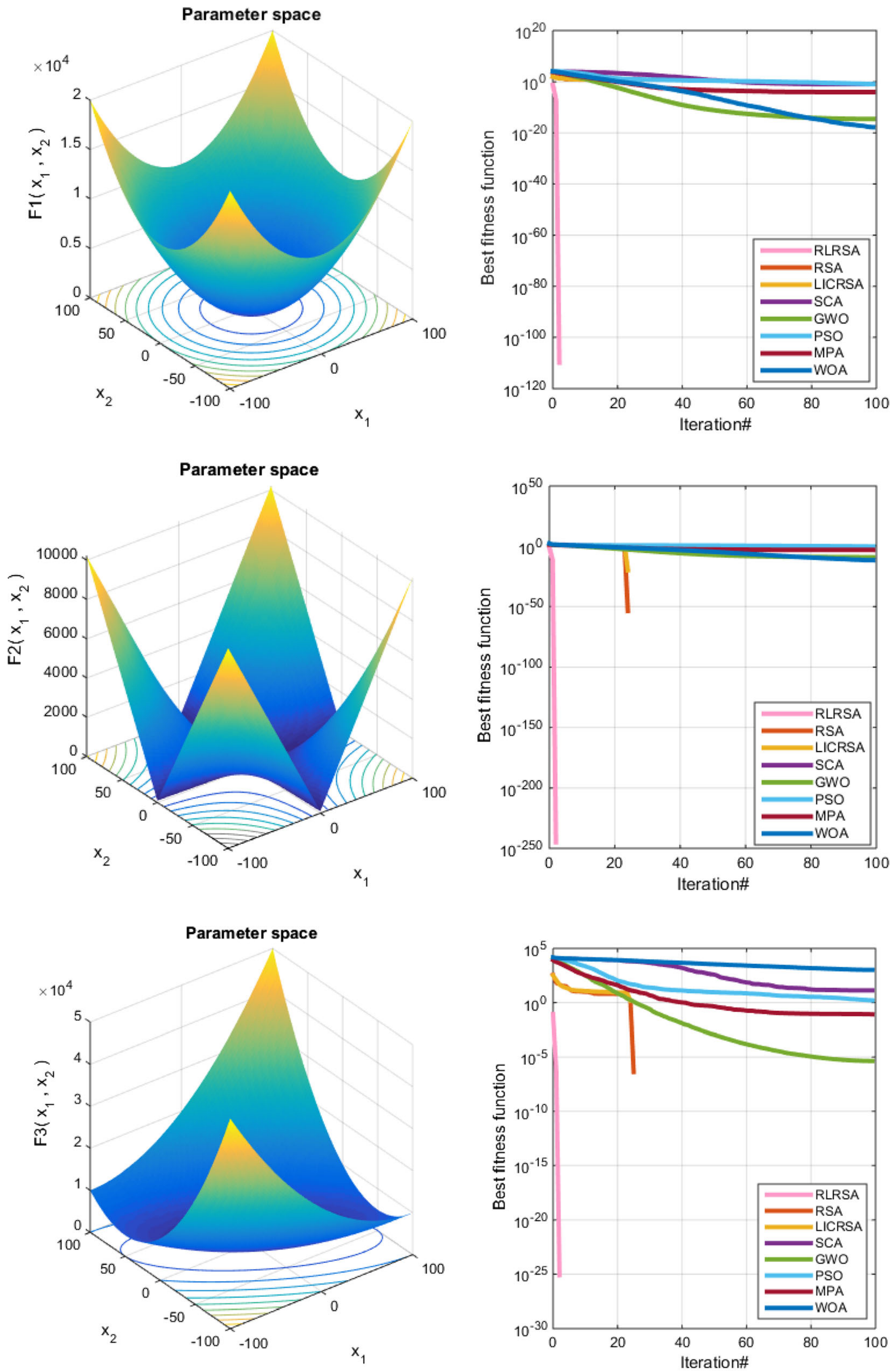

Fig. 4 Convergence of RLRSA against other method

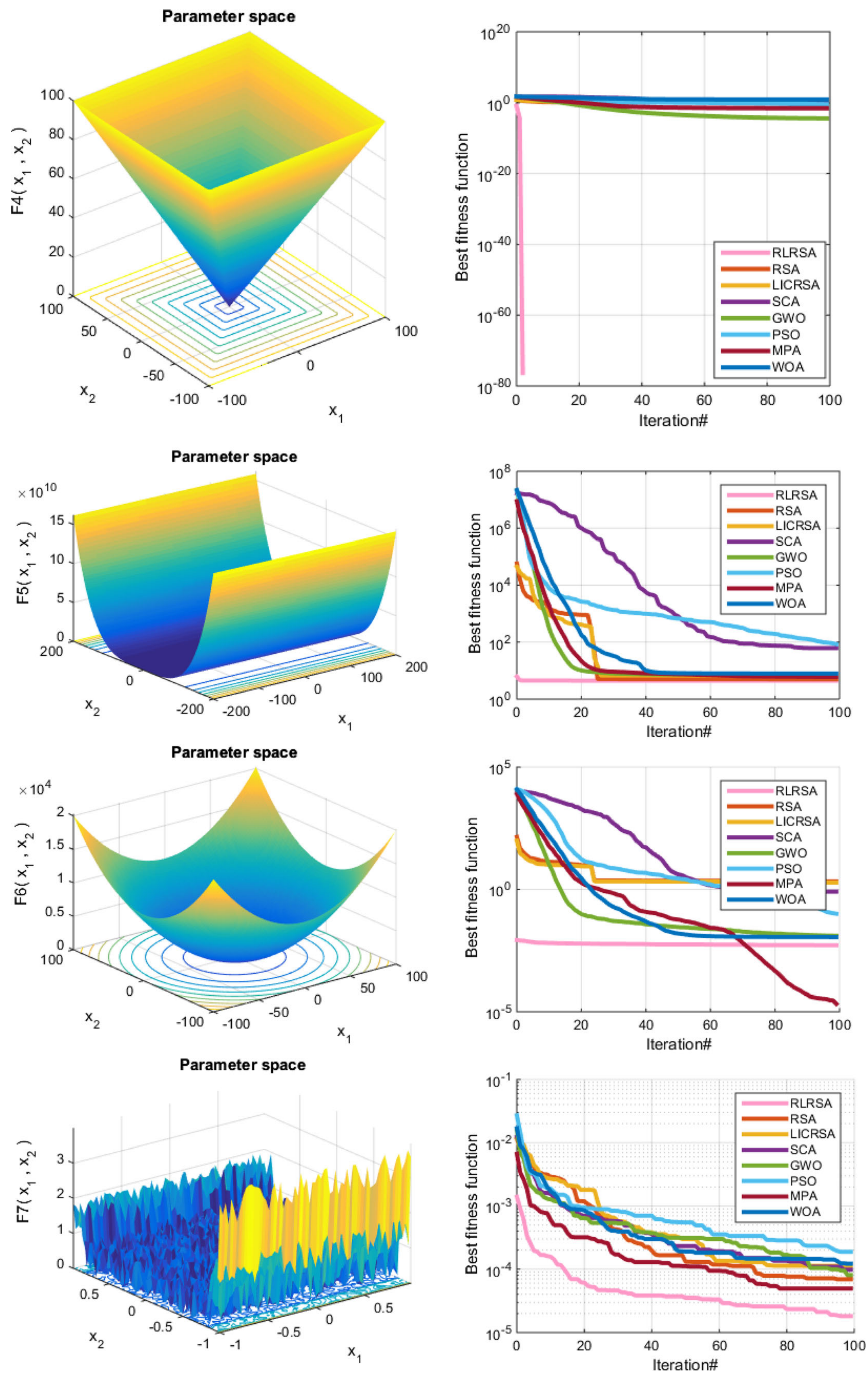

Fig. 4 continued

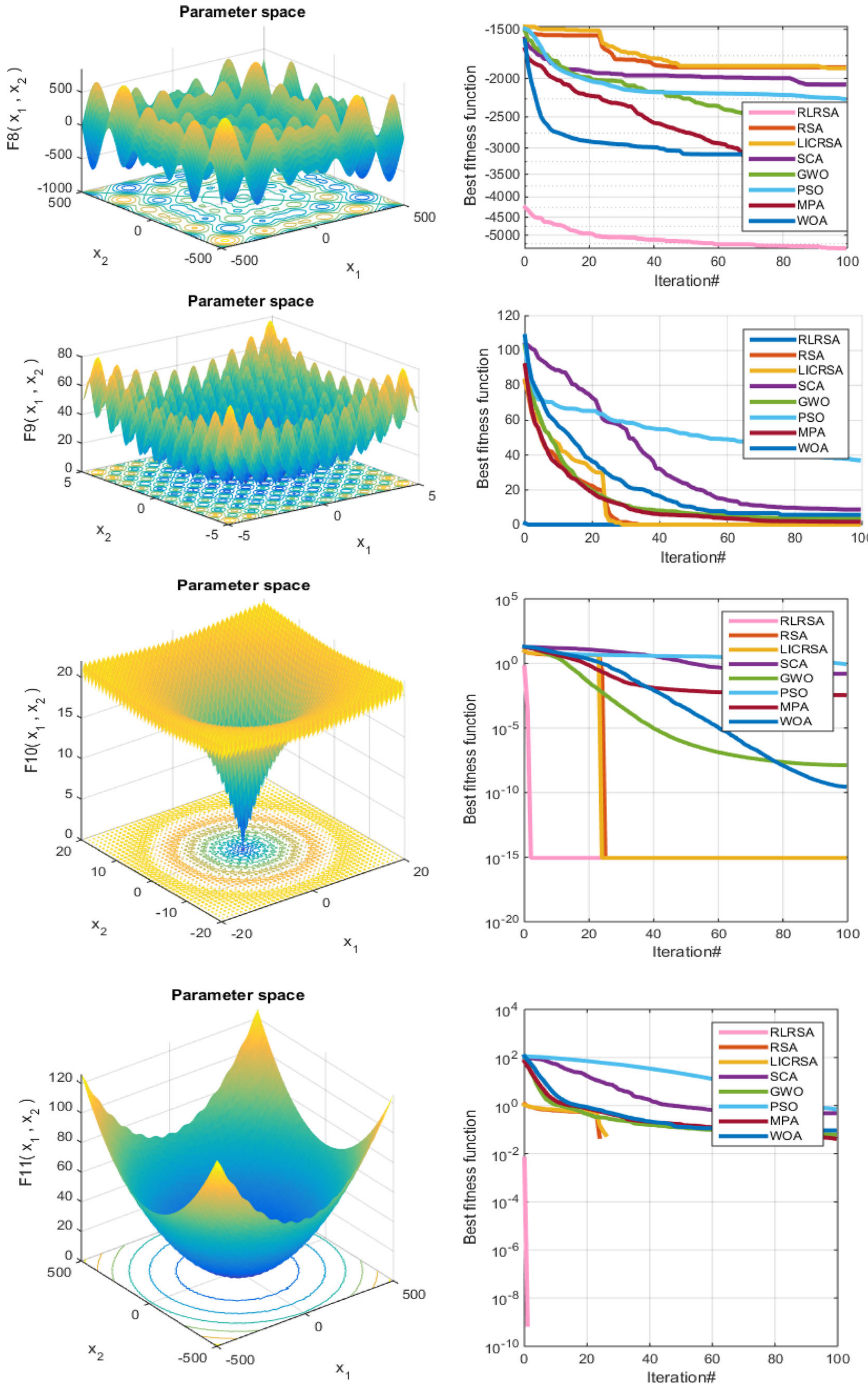

Fig. 4 continued

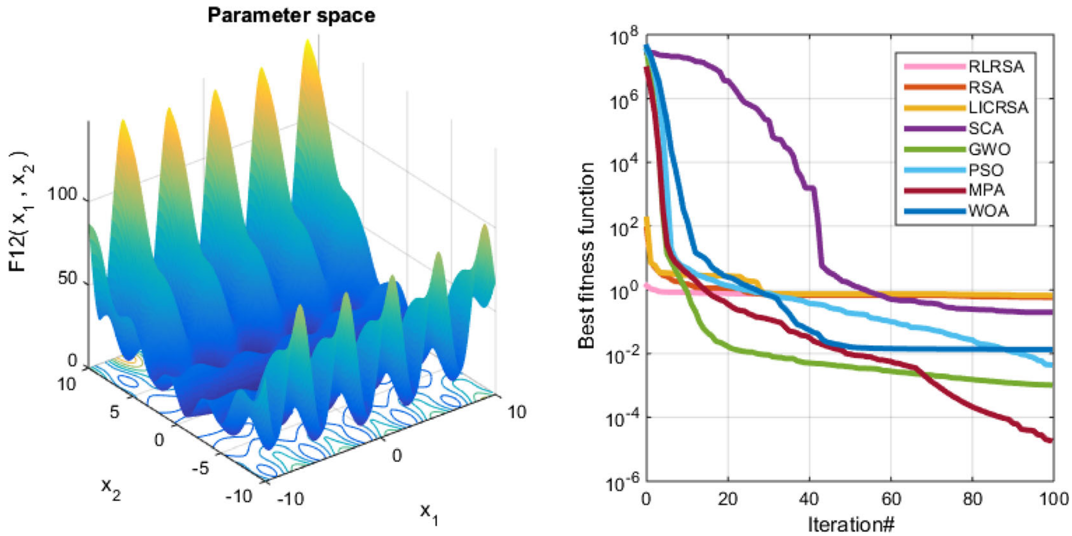

Fig. 4 continued

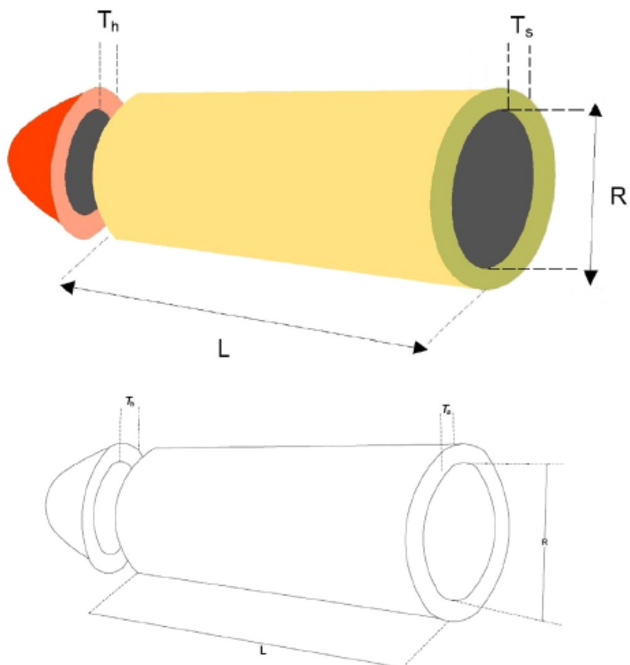

Fig. 5 Pressure vessel design problem [18]

thickness of the head (T_h). The main aim of the design is to minimize the cost of welding, forming of the pressure vessel and the materials.

The mathematical formulation of this problem is:

Consider $\vec{x} = [x_1 x_2 x_3 x_4]$

Minimize $f\left(\vec{x}\right) = 0.6224x_1x_2x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$

Subject to $g_1\left(\vec{x}\right) = -x_1 + 0.0193x_3 \leq 0, g_2\left(\vec{x}\right) = -x_3 + 0.000954x_3 \leq 0,$

$$g_3\left(\vec{x}\right) = -3.14x_4x_3^2 - \frac{4}{3}x_3^2 + 1.296000 \leq 0, g_4\left(\vec{x}\right) = x_4 - 240 \leq 0$$

Variables range $0 \leq x_1, x_2 \leq 99), (10 \leq x_3, x_4 \leq 200)$

It can be seen from the results in Table 10 that the proposed RLRSA can get the best solution which achieves the smallest cost with nearly 5888.4833 with a difference of 146.275 compared with the standard RSA. The main reason for the superiority of the RLRSA compared to RSA is that the reinforcement learning as well as ROL enhance exploration and exploitation phase and balance between them when it is necessary.

6.4.2 Tension/compression spring design problem

The design of tension/compression spring problem shown in Fig. 6 includes the optimization of the wire diameter (d), the mean coil diameter (D), the number of coils (N), and the objective function and the constraints for this problem can be described as follows.

$X = [x_1, x_2, x_3,]$

Minimize $f(X) = (x_3 + 2)x_2{x_1}^2$

subject to

$$g_1(X) = 1 - \frac{x_2^3 x_3}{71785x_1^4} \leq 0$$

$$g_2(X) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0$$

$$g_3(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$

Table 10 Compassion between RLRSA and different algorithms for solving pressure vessel design problem

Algorithm	Optimal values for variables				Optimal cost
	T_s	T_h	R	L	
Branch-bound [112]	1.1250	0.6250	48.9700	106.7200	7982.5000
GWO [23]	0.8125	0.4345	42.0892	176.7587	6051.5639
MVO [6]	0.8125	0.4375	42.0907	176.7387	6060.8066
WOA [17]	0.8125	0.4375	42.0983	176.6390	6059.7410
ES [113]	0.8125	0.4375	42.0981	176.6405	6059.7456
CPSO [114]	0.8125	0.4375	42.0913	176.7465	6061.0777
CSCA	0.8125	0.4375	42.0984	176.6377	6059.7340
HS [115]	1.1250	0.6250	58.2902	43.6927	7197.7300
GA [116]	0.8125	0.4375	42.0974	176.6541	6059.9463
PSO-SCA [117]	0.8125	0.4375	42.0984	176.6366	6059.7143
HPSO [117]	0.8125	0.4375	42.0984	176.6366	6059.7143
ACO [118]	0.8125	0.4375	42.0984	176.6378	6059.7258
GSA [32]	1.1250	0.6250	55.9887	84.4542	8538.8359
RSA [18]	0.8401	0.4190	43.3812	161.5556	6034.7591
RLRSA	0.7787	0.3849	40.3332	199.8388	5888.4833

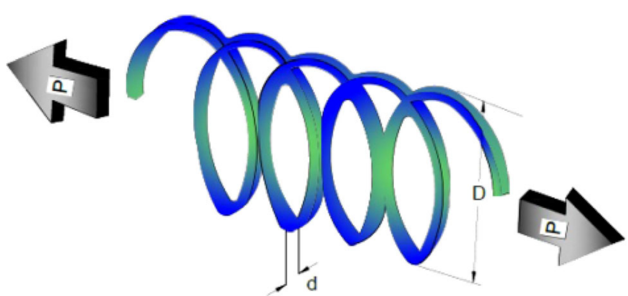

Fig. 6 Tesnion/compression spring design problem [18]

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

the range of variables

$$0.05 \leq x_1 \leq 2.0$$

$$0.25 \leq x_2 \leq 1.3$$

$$2.0 \leq x_3 \leq 15$$

Table 11 shows the solutions to the problem using different optimization methods. It is obvious that the RLRSA attains the better performance and get the best solution of

Table 11 Compassion between RLRSA and different algorithms for Tension/compression spring design problem

Algorithm	Optimal values for variables			Optimal cost
	d	D	N	
GWO	0.0516900000	0.3567370000	11.2888500000	0.0126660000
MVO	0.0525100000	0.3760100000	10.3351400000	0.0127910000
WOA	0.0512080000	0.3452140000	12.0040310000	0.0126762000
ES	0.0516440000	0.3553610000	11.3979250000	0.0126970000
CPSO	0.0517280000	0.3576430000	11.2445440000	0.0126746000
CSCA	0.0516090000	0.3547150000	11.4108320000	0.0126703000
HS	0.0511540000	0.3498700000	12.0764310000	0.0126705000
GA	0.0514801000	0.3516600000	11.6322000000	0.0127047800
PSO	0.0517280000	0.3576440000	11.2445430000	0.0126747000
GSA	0.0502760000	0.3236800000	13.5254100000	0.0127022000
RSA	0.0578140000	0.5847800000	4.0167000000	0.0117600000
RLRSA	0.0551180000	0.5059000000	5.1167000000	0.0109380000

the problem and achieves minimum weight of the objective function.

7 Conclusion

This work presents an enhanced reptile search algorithm (RLRSA) by integrating the Q-learning model and random opposite-based learning (ROBL) strategies. The Q-learning model helps the individual to learn from the results and to switch between encircling and hunting phases, and hence increase the convergence speed of the algorithm. In addition, ROBL prevents the algorithm from trapping in local optima through increasing the diversity of the population. The experimental results show competitive performance compared to the basic RSA and other state-of-the-art algorithms. Moreover, the proposed is applied to solve pressure vessel design and tension/compression spring engineering design problems. The results also show the competitive performance of RLRSA compared to other methods from the literature. One drawback of the proposed RLRSA is that it has only significant impact when the search space is too large but in case of simple functions with only one extreme solution, the algorithm does not show significant difference than the standard RSA.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB).

Data availability All data generated or analyzed during this study are included in this published article.

Declarations

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

References

1. Horst R, Tuy H (2013) Global optimization: deterministic approaches. Springer Science & Business Media
2. Abualigah L (2021) Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. Neural Comput Appl 33(7):2949–2972
3. Abualigah L, Diabat A (2020) A comprehensive survey of the Grasshopper optimization algorithm: results, variants, and applications. Neural Comput Appl 32(19):15533–15556
4. Luenberger DG, Ye Y (1984) Linear and nonlinear programming, vol 2. Springer
5. Gardiner CW (1985) Handbook of stochastic methods, vol 3. Springer, Berlin
6. Abualigah L (2020) Multi-verse optimizer algorithm: a comprehensive survey of its results, variants, and applications. Neural Comput Appl 32(16):12381–12401
7. Abualigah L, Diabat A, Geem ZW (2020) A comprehensive survey of the harmony search algorithm in clustering applications. Appl Sci 10(11):3827
8. Talbi E-G (2009) Metaheuristics: from design to implementation, vol 74. John Wiley & Sons
9. Soler-Dominguez A, Juan AA, Kizys R (2017) A survey on financial applications of metaheuristics. ACM Comput Surv (CSUR) 50(1):1–23
10. Valadi J, Siarry P (2014) Applications of metaheuristics in process engineering, vol 31. Springer
11. Abualigah L, Diabat A (2021) Advances in sine cosine algorithm: a comprehensive survey. Artif Intell Rev 54(4):2567–2608
12. Parpinelli RS, Lopes HS (2011) New inspirations in swarm intelligence: a survey. Int J Bio-Inspired Comput 3(1):1–16
13. Kosorukoff A (2001) Human based genetic algorithm. In: 2001 IEEE international conference on systems, man and cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat. No. 01CH37236). IEEE
14. Biswas A et al (2013) Physics-inspired optimization algorithms: a survey. J Optim 2013
15. Fonseca CM, Fleming PJ (1995) An overview of evolutionary algorithms in multiobjective optimization. Evol Comput 3(1):1–16
16. Kennedy (1995) Particle swarm optimization. Neural Netw
17. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
18. Abualigah L et al (2022) Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer. Expert Syst Appl 191:116158
19. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
20. Heidari AA et al (2019) Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872
21. Mirjalili S et al (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw
22. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. Nature inspired cooperative strategies for optimization (NICSO 2010). Springer, pp 65–74
23. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
24. Rao RV, Savsani VJ, Vakharia D (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
25. Shi Y (2011) Brain storm optimization algorithm. International conference in swarm intelligence. Springer

26. Mousavirad SJ, Ebrahimpour-Komleh H (2017) Human mental search: a new population-based metaheuristic optimization algorithm. Appl Intell 47(3):850–887
27. Moosavi SHS, Bardsiri VK (2019) Poor and rich optimization algorithm: a new human-based and multi populations algorithm. Eng Appl Artif Intell 86:165–181
28. Wang L et al (2014) A simple human learning optimization algorithm. Computational intelligence, networked systems and their applications. Springer, pp 56–65
29. Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE congress on evolutionary computation. IEEE
30. Liñán-García E, Gallegos-Araiza LM (2012) Simulated annealing with previous solutions applied to DNA sequence alignment. ISRN Artif Intell 2012
31. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133
32. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248
33. Abualigah L et al (2021) The arithmetic optimization algorithm. Comput Methods Appl Mech Eng 376:113609
34. Abedinpourshotorban H et al (2016) Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm. Swarm Evol Comput 26:8–22
35. Javidy B, Hatamlou A, Mirjalili S (2015) Ions motion algorithm for solving optimization problems. Appl Soft Comput 32:72–79
36. Ekinci S, Hekimoğlu B, Izci D (2021) Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor. Eng Sci Technol Int J 24(2):331–342
37. Holland JH (1992) Genetic algorithms. Sci Am 267(1):66–73
38. Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
39. Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol Comput 11(1):1–18
40. Issa M, Abd Elaziz, M (2020) Analyzing COVID-19 virus based on enhanced fragmented biological Local Aligner using improved Ions Motion Optimization algorithm. Appl Soft Comput 106683
41. Issa M, Hassanien AE (2017) Multiple sequence alignment optimization using meta-Heuristic techniques. Handbook of research on machine learning innovations and trends. IGI Global, pp 409–423
42. Issa M et al (2018) Pairwise global sequence alignment using Sine-Cosine optimization algorithm. International conference on advanced machine learning technologies and applications. Springer
43. Ali AF, Hassanien A-E (2016) A survey of Metaheuristics methods for bioinformatics applications. Applications of intelligent optimization in biology and medicine. Springer, pp 23–46
44. Issa M (2021) Expeditious Covid-19 similarity measure tool based on consolidated SCA algorithm with mutation and opposition operators. Appl Soft Comput 104:107197
45. Issa M et al (2022) A biological sub-sequences detection using integrated BA-PSO based on infection propagation mechanism: case study COVID-19. Expert Syst Appl 189:116063
46. Issa M, Helmi, A (2021) Two layer hybrid scheme of IMO and PSO for optimization of local aligner: COVID-19 as a case study. Artif Intell COVID-19 363–381
47. Chen H et al (2020) Parameters identification of photovoltaic cells and modules using diversification-enriched Harris hawks optimization with chaotic drifts. J Clean Prod 244:118778
48. Yu K et al (2018) Multiple learning backtracking search algorithm for estimating parameters of photovoltaic models. Appl Energy 226:408–422
49. Abd Elaziz M et al (2023) Optimal parameters extracting of fuel cell based on Gorilla Troops Optimizer. Fuel 332:126162
50. Almodfer R et al (2022) Improving parameter estimation of fuel cell using Honey Badger optimization algorithm. Front Energy Res 10:875332
51. Issa M (2022) Parameter tuning of PID controller based on arithmetic optimization algorithm in IOT systems. Integrating Meta-Heuristics and machine learning for real-world optimization problems. Springer, pp 399–417
52. Issa M (2022) Enhanced arithmetic optimization algorithm for parameter estimation of PID controller. Arab J Sci Eng 1–15
53. Issa M (2021) Performance optimization of PID controller based on parameters estimation using Meta-Heuristic techniques: a comparative study. Metaheuristics in machine learning: theory and applications. Springer, pp 691–709
54. Issa M et al (2019) PID controller tuning parameters using Meta-heuristics algorithms: comparative analysis. Machine learning paradigms: theory and application. Springer, pp 413–430
55. Mehdizadeh SA (2015) Optimization of passive tractor cabin suspension system using ES, PSO and BA. J Agric Technol 11(3):595–607
56. Issa M, Samn A (2022) Passive vehicle suspension system optimization using Harris Hawk Optimization algorithm. Math Comput Simul 191:328–345
57. Shao Y et al (2021) Multi-objective neural evolutionary algorithm for combinatorial optimization problems. IEEE Trans Neural Netw Learn Syst
58. Issa M, Mostafa Y (2022) Gradient-based optimizer for structural optimization problems. Integrating Meta-Heuristics and machine learning for real-world optimization problems. Springer, pp 461–480
59. Soliman MM, Hassanien AE (2017) 3D watermarking approach using particle swarm optimization algorithm. Handbook of research on machine learning innovations and trends. IGI Global, pp 582–613
60. Issa M (2018) Digital image watermarking performance improvement using bio-inspired algorithms. Advances in soft computing and machine learning in image processing. Springer, pp 683–698
61. Mookiah S, Parasuraman K, Kumar Chandar S (2022) Color image segmentation based on improved sine cosine optimization algorithm. Soft Comput 1–11
62. Giuliani D (2022) Metaheuristic algorithms applied to color image segmentation on HSV space. J Imaging 8(1):6
63. Lin JC-W et al (2016) Mining high-utility itemsets based on particle swarm optimization. Eng Appl Artif Intell 55:320–330
64. Lin JC-W et al (2016) A sanitization approach for hiding sensitive itemsets based on particle swarm optimization. Eng Appl Artif Intell 53:1–18
65. Lin JC-W et al (2019) Hiding sensitive itemsets with multiple objective optimization. Soft Comput 23:12779–12797
66. Mohamed Issa AH, Ibrahim Ziedan, Ahmed Alzohairy (2017) Maximizing occupancy of GPU for fast scanning biological database using sequence alignment. J Appl Sci Res 13(6)
67. Elloumi M, Issa MAS, Mokaddem A (2013) Accelerating pairwise alignment algorithms by using graphics processor units. In: Biological knowledge discovery handbook: preprocessing, mining, and postprocessing of biological data, pp 969–980
68. Issa ME et al (2022) Human activity recognition based on embedded sensor data fusion for the internet of healthcare things. In: Healthcare. MDPI

69. Almodfer R et al, Improving parameters estimation of fuel cell using Honey Badger optimization algorithm. Front Energy Res 565
70. Djenouri Y et al (2019) Exploiting GPU parallelism in improving bees swarm optimization for mining big transactional databases. Inf Sci 496:326–342
71. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
72. El Shinawi A et al (2021) Enhanced adaptive neuro-fuzzy inference system using reptile search algorithm for relating swelling potentiality using index geotechnical properties: a case study at El Sherouk City. Egypt Math 9(24):3295
73. Khan RA et al (2022) Reptile search algorithm (RSA)-based selective harmonic elimination technique in packed E-Cell (PEC-9) inverter. Processes 10(8):1615
74. Chauhan S, Vashishtha G, Kumar A (2022) Approximating parameters of photovoltaic models using an amended reptile search algorithm. J Ambient Intell Humaniz Comput 1–16
75. Yildiz BS et al (2022) Reptile search algorithm and kriging surrogate model for structural design optimization with natural frequency constraints. Mater Test 64(10):1504–1511
76. Sunitha D et al (2022) Congestion centric multi-objective reptile search algorithm-based clustering and routing in cognitive radio sensor network. Trans Emerg Telecommun Technol e4629
77. Raja D, Karthikeyan M (2022) Content based image retrieval using reptile search algorithm with deep learning for agricultural crops. In: 2022 7th international conference on communication and electronics systems (ICCES). IEEE
78. Huang L et al (2022) An improved reptile search algorithm based on lévy flight and interactive crossover strategy to engineering application. Mathematics 10(13):2329
79. Ekinci S et al (2022) Development of Lévy flight-based reptile search algorithm with local search ability for power systems engineering design problems. Neural Comput Appl 34(22):20263–20283
80. Ekinci S, Izci D (2022) Enhanced reptile search algorithm with Lévy flight for vehicle cruise control system design. Evol Intell 1–13
81. Almotairi KH, Abualigah L (2022) Hybrid reptile search algorithm and remora optimization algorithm for optimization tasks and data clustering. Symmetry 14(3):458
82. Abualigah L, Diabat A (2022) Chaotic binary reptile search algorithm and its feature selection applications. J Ambient Intell Humaniz Comput 1–17
83. Al-Shourbaji I et al (2022) Boosting ant colony optimization with reptile search algorithm for churn prediction. Mathematics 10(7):1031
84. Almotairi KH, Abualigah L (2022) Improved reptile search algorithm with novel mean transition mechanism for constrained industrial engineering problems. Neural Comput Appl 1–21
85. Al-Shourbaji I et al (2022) An efficient parallel reptile search algorithm and snake optimizer approach for feature selection. Mathematics 10(13):2351
86. Yuan Q et al (2022) a modified reptile search algorithm for numerical optimization problems. Comput Intell Neurosci 2022
87. Elgamal Z et al (2022) Improved reptile search optimization algorithm using chaotic map and simulated annealing for feature selection in medical filed. IEEE Access
88. Wu D et al (2022) An improved teaching-learning-based optimization algorithm with reinforcement learning strategy for solving optimization problems. Comput Intell Neurosci 2022
89. Lingam G, Rout RR, Somayajulu DV (2019) Adaptive deep Q-learning model for detecting social bots and influential users in online social networks. Appl Intell 49(11):3947–3964
90. Liu F, Zeng G (2009) Study of genetic algorithm with reinforcement learning to solve the TSP. Expert Syst Appl 36(3):6995–7001
91. Xu Y, Pi D (2020) A reinforcement learning-based communication topology in particle swarm optimization. Neural Comput Appl 32(14):10007–10032
92. Emary E, Zawbaa HM, Grosan C (2017) Experienced gray wolf optimization through reinforcement learning and neural networks. IEEE Trans Neural Netw Learn Syst 29(3):681–694
93. Wang J, Lei D, Cai J (2022) An adaptive artificial bee colony with reinforcement learning for distributed three-stage assembly scheduling with maintenance. Appl Soft Comput 117:108371
94. Si T, Miranda PB, Bhattacharya D (2022) Novel enhanced Salp Swarm Algorithms using opposition-based learning schemes for global optimization problems. Expert Syst Appl 207:117961
95. Too J, Sadiq AS, Mirjalili SM (2022) A conditional opposition-based particle swarm optimisation for feature selection. Connect Sci 34(1):339–361
96. Izci D et al (2022) A novel modified opposition-based hunger games search algorithm to design fractional order proportional-integral-derivative controller for magnetic ball suspension system. Adv Control Appl Eng Ind Syst 4(1):e96
97. Abd Elaziz M, Oliva D, Xiong S (2017) An improved opposition-based sine cosine algorithm for global optimization. Expert Syst Appl 90:484–500
98. Gupta S, Deep K (2019) A hybrid self-adaptive sine cosine algorithm with opposition based learning. Expert Syst Appl 119:210–230
99. Hussien AG, Amin M (2022) A self-adaptive Harris Hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection. Int J Mach Learn Cybern 13(2):309–336
100. Yang Y et al (2022) An opposition learning and spiral modelling based arithmetic optimization algorithm for global continuous optimization problems. Eng Appl Artif Intell 113:104981
101. Watkins CJCH D P (1992) Q-learning. mach learn
102. Tizhoosh HR (2005) Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06). IEEE
103. Long W et al (2019) A random opposition-based learning grey wolf optimizer. IEEE Access 7:113810–113825
104. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
105. Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. Evol Comput 9(2):159–195
106. Mirjalili S et al (2017) Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv Eng Softw 114:163–191
107. Faramarzi A et al (2020) Marine predators algorithm: a nature-inspired metaheuristic. Expert Syst Appl 152:113377
108. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073
109. Faramarzi A et al (2020) Equilibrium optimizer: a novel optimization algorithm. Knowl-Based Syst 191:105190
110. Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the sixth international symposium on micro machine and human science. IEEE
111. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47
112. Sandgren E (1988) Nonlinear integer and discrete programming in mechanical design. In: International design engineering

technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers
113. Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37(4):443–473
114. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20(1):89–99
115. Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188(2):1567–1579
116. Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41(2):113–127
117. He Q, Wang L (2007) A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Appl Math Comput 186(2):1407–1422
118. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput