Research Article

Balwin-Teaching-Learning-Based Artificial Raindrop Algorithm for UAV Route Planning

Bin Xin,1 Fan Wang,1 and Zhibo Zhai2

1School of Optoelectronic Engineering, Xi’an Technological University, 2 Xuefu Middle Road Xi’an, Xi’an 710021, Shaanxi, China
2College of Mechanical and Equipment Engineering, Hebei University of Engineering, 19 Taiji Road Handan, Handan 056038, Hebei, China

Correspondence should be addressed to Bin Xin; xinbin1227@163.com

Received 15 September 2020; Revised 21 January 2021; Accepted 10 March 2021; Published 18 March 2021

1.Introduction

Optimization problems such as UAV route planning are quite common in the academia, and they have become an exciting new area of mathematical programming for several decades. Since optimization issues are mostly nonlinear or nondifferentiable, approaches based on gradient are inapplicable. To address these problems, many optimization algorithms have been developed and proved effective. Among them, the typical algorithms include particle swarm optimization (PSO) [1], differential evolution (DE) [2], and grey wolf optimizer with ranking-based mutation operator [3].

Artificial raindrop algorithm (ARA) [4] was initially introduced by Jiang in 2014 to simulate the natural rainfall process. In the ARA, each raindrop stands for a potential solution to the optimization problem, whose location information is evaluated by altitude, and the lowest altitude is the global optimal. ARA is one nature-based metaheuristic searching algorithm which is capable of assessing their performance simultaneously by their corresponding altitudes in the search space. Each raindrop reaches the ground via free fall from the cloud and then moves towards the problem’s optimal solution, finally leading to convergence [5]. Since its introduction, ARA has become a preferred optimization technology because of its easy implementation and powerful search capabilities on different benchmarks as well as in digital IIR filters design problems [6, 7].

Despite the strong performance, ARA is subject to several undesirable dynamical characteristics which reduce its search capability. Among them, premature convergence is the most severe, in which case the raindrop tends to fall into the local optimal solution because of the higher convergence speed and the smaller scope of solutions at the later stages of evolution.

Although a great deal of work has been done to address the abovementioned issues, most of the existing variants fail to offer raindrops of alternative learning strategies when they are incapable of updating the fitness search process. As a result, the search ability of ARA is restricted. Inspired by teaching and learning, Rao et al. put forward a teaching-learning-based optimization (TLBO) algorithm [8] with fast speed. Based on these as well as Baldwin learning strategy, a hybrid algorithm, Balwin-teaching-learning-based artificial...
raindrop algorithm (BTLARA), which has both the advantages of ARA and TLBO, is put forward in the present paper. This proposed algorithm improves solution quality and accelerates the convergence by improving the current TLBO mode and introducing the enhanced framework to ARA. Similar to ARA, the BTLARA adopts eight phases: raindrop generation, raindrop descent, raindrop collision, raindrop flowing, vapors replacement, existing TLBO, and Baldwin learning phases. Specifically, the raindrops first enter the ARA phase and then the global optimal information. Those raindrops failing to improve its fitness in the ARA phase will enter the existing TLBO and the Baldwin learning phases, where one model raindrop is selected and provides guidance for the raindrops to find better solutions. The Baldwin learning strategy is applied to guarantee that fitter raindrops with higher probability are the likely winners selected. In this way, it provides a more reliable searching direction towards the global optimal solution.

In this work, a Balwin-teaching-learning-based artificial raindrop algorithm (BTLARA) is proposed to estimate the UAV route planning by combining hybrid teaching-learning-based optimization with artificial raindrop algorithm. This approach is capable of balancing the exploration and exploitation. Meanwhile, it can obtain higher-quality navigation trace and convergence and better avoid threat routes as compared with other algorithms.

The rest of the present paper is organized as follows. Section 2 introduces the BTLARA. The experimental settings and results are given in Section 3. Section 4 applies BTLARA to UAV route planning. The major conclusions of this paper are presented in Section 5.

2. Baldwin-Teaching-Learning-Based Artificial Raindrop Algorithm

This paper proposes the Baldwin-teaching-learning-based artificial raindrop algorithm (BTLARA), an improved version of ARA.

2.1. Artificial Raindrop Algorithm. Artificial raindrop algorithm is a heuristic algorithm based on population. As an algorithm inspired by the rainfall process, it is thus divided into the following six stages: generation, falling, collision, flowing of raindrop, and vapors replacement. In the process of optimization, the location information of water vapor or raindrop is assessed by altitude. The raindrop pool records the location with lower altitude. Figure 1 illustrates the procedure of raindrop generation where the grey circles and blue circles correspond to vapors and raindrop, respectively.

In ARA, a vapor corresponds to a feasible solution, and altitude is chosen as fitness function and used to compute the fitness measurement for vapor or raindrop. The population is made up of some vapors and evolved by five operators, containing five main raindrop operators. The specific definition of the main raindrop operators is given in Table 1.

Table 2 presents the ARA’s pseudocode.

2.2. Teaching-Learning-Based Optimization. The teaching-learning-based optimization algorithm introduced by Rao [9] is an optimization technique based on population, which is analogous to the teaching and learning process between the teacher and students. It has two steps, i.e., the teacher phase and the learner phase. Figure 2 shows the distribution of scores the learners obtain in both classes. Curve A indicates the original distribution of scores with the average value $M_A$. Since the most experienced in the class is the teacher, the best learner is analogous to a teacher. $T_A$ is the teacher in the class with the mean $M_A$. Teacher $T_A$ will try to make the average $M_A$ move towards his own level based on his own ability, thus rising the learners’ level to a new average $M_B$. Though efforts are made by teacher $T_A$ to teach the learners, they can only acquire a certain amount of knowledge according to the teaching quality and their own ability or quality. The learner’s quality is judged from the mean value of their population. Teacher $T_A$ increases the students’ quality from $M_A$ to $M_B$, during which the students will get a new teacher who has superior quality, which is $T_B$ for example. Therefore, a second curve B with the newly obtained teacher $T_B$ will be generated.

2.2.1. Teacher Phase. In this phase, each student learns from the teacher $X_{teacher}$, the best individual among all. Specifically, $X_i (i = 1, 2, ..., N$, where $N$ is the number of learners) is a vector of learner which includes $X_{i}$. It consists of various subjects $X_i (x_{i1}, x_{i2}, ..., x_{iD})$ where $D$ denotes the number of dimensions analogous to different subjects) and can be written as

$$X_{i,new} = X_i + r(X_{teacher} - TX_{mean}),$$ (1)

where $X_{i,new}$ is a newly generated individual from $X_i$, $X_{mean}$ and $X_{teacher}$ are the current average value of all individuals and the best individual among the current populations, respectively. $r$ is a random number within the range of $[0, 1]$, and $TF$ is the teaching factor and it is either 1 or 2:

$$T_F = \text{round}[1 + \text{rand}(0, 1)].$$ (2)

2.2.2. Learner Phase. In this phase, each learner learns new things from another randomly selected learner to improve his or her knowledge if the selected learner has better knowledge than him. The updating learning equation for the learner $X_i$ is

$$X_{i,new} = \begin{cases} X_i + r(X_i - X_j), & \text{if } f(X_i) < f(X_j); \\ X_j + r(X_j - X_i), & \text{if } f(X_j) < f(X_i), \end{cases}$$ (3)

where $X_i$ and $X_j$ are learners, and $f(X_i)$ is the sum of all the scores of subjects for the learner $i^{th}$.

2.3. Baldwin Learning Strategy. The Baldwin learning [10] generates a new learner $X_{i,new}$ through evaluating the difference between two learners $X_i$ and $X_{mean}$ and adding the weighted difference between the two to the learner $X_i$. The
### Table 1: Specific definition of the main raindrop operators.

| Name                          | Detailed definition                                                                 |
|-------------------------------|--------------------------------------------------------------------------------------|
| Raindrop generation operator  | $\phi^G R \left( \text{Pop} \left( t \right) \right) = \left( \sum_{i=1}^{N} \frac{1}{N} \text{Vapor}_i \left( t \right) \right) / \sum_{i=1}^{N} \frac{1}{N} \text{Vapor}_i \left( t \right) / N \ldots $$ \sum_{i=1}^{N} \frac{1}{N} \text{Vapor}_i \left( t \right) / N$ |
| Raindrop descent operator     | $\phi^D R \left( \text{Raindrop} \left( t \right) \right) = \text{Raindrop}_{k2} \left( t \right) + \Phi \ast \left( \text{Raindrop}_{k3} \left( t \right) - \text{Raindrop}_{k4} \left( t \right) \right)$, $k \in \{1, 2, \ldots, N\}$; $\Phi \in (-1, 1)$ |
| Raindrop collision operator   | $\phi^C R \left( \text{New_Raindrop} \left( t \right) \cup \text{Pop} \left( t \right) \right)$ |
| Vapor flowing operator        | $\phi^F R \left( \text{Small_Raindrop} \left( t \right) \right) = \text{Small_Raindrop}_i \left( t \right) + d \left( t, \lambda \right)$ |
| Vapor replacement operator    | $\phi^R V \left( \text{Pop} \left( t \right) \cup \text{Small_Raindrop} \left( t \right) \right)$ |

### Table 2: The pseudocode of ARA.

**Input:** $N$, the population size; $D$, the dimensions of optimization problem; $\tau$, the flowing step parameters; $\text{RP}$, the raindrop pool; $\text{Max\_Flow\_Number}$, maximum number of flowing; $\text{Max\_FES}$, maximum number of function evaluations.

```plaintext
01 $t = 0$;
02 Generate an initial population: $\text{Pop} \left( 0 \right) = \{ \text{Vapor}_1 \left( 0 \right), \text{Vapor}_2 \left( 0 \right), \ldots, \text{Vapor}_N \left( 0 \right) \}$ by uniformly and randomly sampling from the feasible solution space;
03 Evaluate the objective function values $f \left( \text{Vapor}_1 \left( 0 \right) \right)$, $f \left( \text{Vapor}_2 \left( 0 \right) \right)$, $\ldots$, $f \left( \text{Vapor}_N \left( 0 \right) \right)$;
04 $\text{FES} = N$;
05 Find the best position best (0) of the initial population;
06 $\text{RP} = \text{gbest} \left( 0 \right)$;
07 While $\text{FES} \leq \text{Max\_FES}$ do
08 $\text{Raindrop} \left( t \right) = \left( \left( 1 / N \right) \sum_{i=1}^{N} \text{Vapor}_i \left( t \right) \right), \left( \left( 1 / N \right) \sum_{i=1}^{N} \text{Vapor}_i \left( t \right) \right), \ldots, \left( \left( 1 / N \right) \sum_{i=1}^{N} \text{Vapor}_i \left( t \right) \right)$;
09 Trail = $\text{Raindrop} \left( t \right)$;
10 Randomly chosen indexes: $r_1$, $r_2$, $r_3$, and $r_4 \in \{1, 2, \ldots, D\}$; Trail$$_{1} = \text{Raindrop}_{k2} \left( t \right) + \Phi \ast \left( \text{Raindrop}_{k3} \left( t \right) - \text{Raindrop}_{k4} \left( t \right) \right)$;
11 If $f \left( \text{Trail} \right) < f \left( \text{Raindrop} \left( t \right) \right)$
12 New_Raindrop $\left( t \right) = \text{Trail}$;
13 Else
14
```

**Figure 1:** The simulation scene graph of artificial raindrop algorithm.
The process of Baldwinian learning is shown in Figure 3. It can be observed that the learners close to the teacher should be added to encourage, because these learners are more likely to become winners. Therefore, the Baldwinian learning operator simulates the learning mechanism in the learner phase by applying information from within the population to change the search space.

![Figure 2: Schematic diagram of TLBO algorithm.](image)

![Figure 3: The Baldwin learning mechanism.](image)

The updating equation of the learner phase is

\[
X_{i,\text{new}} = \begin{cases} 
X_i + s \ast (X_l - X_m), & \text{if } r \leq p_t, \\
X_i, & \text{else,} 
\end{cases}
\]  

(4)

where \(l, m \in [1, 2, \ldots, N], i \neq l \neq m\) and \(f(X_l) > f(X_m)\), and \(s > 0\) is the strength of Baldwin learning. \(p_t \in (0, 1]\) represents the possibility of Baldwin learning, and \(r\) is a vector, the
Mathematical Problems in Engineering

3. Experimental Studies

3.1. Test Benchmark Functions. In order to assess the performance of the proposed BTLARA, six contest benchmark functions are adopted in the experiments. Table 3 lists the detailed definitions of these functions [11]. Figure 5 shows the plots of six benchmark functions.

3.2. Experimental Platform and Termination Criterion. The experiments were all performed on the computer equipped with a Celeron 2.26 GHz CPU, a 2 GB memory, and a Windows XP operating system with MATLAB 2012. To reduce statistical errors, each experiment was repeated 25 times for all the six test functions of 30 dimensions. In addition, 300,000 function evaluations (FEs) [12] were adopted as the stopping criterion.

3.3. Performance Metric. The average values \( F_{\text{mean}} \) of the function’s error \( f(X) - f(X^*) \) are recorded to assess the performance of the algorithms. The \( f(X) \) and \( f(X^*) \) here are the best fitness value for solution and the real global optimization value of the test problem, respectively. The standard deviation (SD) indicates the robust of the optimization algorithms on \( F_1 \)–\( F_6 \) test functions on 30 dimensions. The results obtained by the algorithms for the same problem are compared by statistical analysis, so as to investigate the differences between the algorithms’ performance in optimization. Thus, Wilcoxon’s rank sum test [13] at the significance level of 0.05 is adopted. The statistical tool judges whether the average values \( F_{\text{mean}} \) of two solutions obtained any two algorithms differ statistically.

3.4. Population Size Influence on BTLARA Performance. The population is now set from 10 to 90 in increments of 20 with the other parameters, the same as previously. Table 5 gives the results for different population sizes, where the figures in bold are the best results. As shown in Table 5, the mean value \( F_{\text{mean}} \) of BTLARA \( (N = 30) \) is higher than those of other algorithms on \( F_3, F_2, F_3, F_4, \) and \( F_5 \), but similar on \( F_6 \). It thus can be concluded that, on the whole, BTLARA \( (N = 30) \) outperforms others. This may be ascribed to the premature convergence resulted from the smaller population size, because a larger population size will significantly decrease the probability of finding the right searching direction.

3.5. Comparison with the Original ARA, TLBO, and BTLARA Algorithm. The BTLARA is compared to oral version of TLBO and ARA as well as the teaching-learning-based artificial raindrop algorithm (TLARA) which has no Baldwin learning strategy for convenient implementation. The parameters for oral version of TLBO and ARA are obtained from the corresponding references. The parameter for the TLARA is same as that for BTLARA. Each algorithm runs independently 25 times. Table 6 presents the statistical results of \( F_{\text{mean}} \) and SD, with the experimental results listing in the last three rows. Figure 6 shows the evolution plots of TLBO, ARA, TLARA, and BTLARA. Moreover, the correlation between the average errors of the functions is analyzed by the semilogarithmic convergence plots. The values in bold are the best results.

The following is to compare BTLARA with the oral version of TLBO and ARA as well as the TLARA. The statistical mean value \( (F_{\text{mean}}) \) in Table 6 shows that BTLARA is superior to other algorithms in terms of the overall performance. BTLARA outperforms TLBO, ARA, and TLARA in six, five, and five test functions out of six test functions, respectively. It is also shown that BTLARA is superior to the other three algorithms on functions \( F_1, F_2, F_3, F_5, \) and \( F_6 \) except \( F_3 \), on which BTLARA performs the same as the TLARA. Thus, it can also be concluded that BTLARA is the best as compared with the oral version of TLBO, ARA, and TLARA.

The possible reasons for the better performance of BTLARA are that TLBO expands the search space and Baldwin learning strategy directly achieves more accurate solution, which contributes to identifying a better solution. Thus, exploration and exploitation are balanced better in the BTLARA. Thus, conclusion can be drawn that BTLARA has the best performance in accuracy as compared with other approaches.

3.6. Comparison with Other Well-Established TLBO Algorithms. In this subsection, comparison is made between BTLARA and some well-established TLBO algorithms, including bird mating optimizer with teaching-learning-based optimization (TLBMO) [14], hybrid algorithm based on harmony search and teaching-learning-based optimization (HSTLBO) [15], teaching-learning-based optimization and differential evolution (TLBODE) [16], and teaching and peer-learning particle swarm optimization (TPLPSO) [17]. The parameters of the other four TLBO algorithms are obtained from corresponding references. Each of the algorithms runs 25 times. Table 7 lists all the statistical results of \( F_{\text{mean}} \) and SD. The experimental results are presented in the last three rows, with the best ones in bold.

BTLARA is compared with some other well-established TLBO algorithms including TLBMO, HSTLBO, TLBODE, and TPLPSO. The statistical mean value \( (F_{\text{mean}}) \) in Table 7 shows that BTLARA has the best overall performance. It outperforms TLBMO, HSTLBO, TLBODE, and TPLPSO algorithms on six, six, and four test functions out of six test functions, respectively. In addition, BTLARA outperforms the other three algorithms on functions \( F_3, F_4, F_3, \) and \( F_5 \), except for \( F_1 \) and \( F_6 \), on which BTLARA performs the same as the TLARA in terms of \( F_{\text{mean}} \). Thus, it can be concluded that BTLARA has the best overall performance as compared with the other well-established TLBO algorithms including TLBMO, HSTLBO, TLBODE, and TPLPSO.
Baldwin learning strategy directly achieves a more accurate solution, which are conducive to identifying a better solution. As a result, BTLARA can better balance exploration with exploitation. It is thus summarized that BTLARA has

### Table 3: The pseudocode of BTLARA.

| Line | Description |
|------|-------------|
| 01   | \( t = 0; \) |
| 02   | Generate an initial population: \( \text{Pop} (0) = \{ \text{Vapor}_1 (0), \text{Vapor}_2 (0), \ldots, \text{Vapor}_N (0) \} \) by uniformly and randomly sampling from the feasible solution space; |
| 03   | Evaluate the objective function values \( f (\text{Vapor}_1 (0)), f (\text{Vapor}_2 (0)), \ldots, f (\text{Vapor}_N (0)) \); |
| 04   | \( \text{FES} = N; \) |
| 05   | Find the best position \( \text{gbest} (0) \) of the initial population; |
| 06   | \( \text{RP} = \text{gbest} (0); \) |
| 07   | \( \text{mbest} = \text{mean} (f (\text{Vapor}_1 (0)), f (\text{Vapor}_2 (0)), \ldots, f (\text{Vapor}_N (0))); \) |
| 08   | While \( \text{FES} \leq \text{Max } \_\text{FES}, \) do |
| 09   | \( \text{Raindrop}(t) = \frac{(1/ N) \sum_{i=1}^{N} \text{Vapor}_i(t)}{(1/ N) \sum_{i=1}^{N} \text{Vapor}_2(t), \ldots, (1/ N) \sum_{i=1}^{N} \text{Vapor}_D(t)}; \) |
| 10   | \( \text{Trail} = \text{Raindrop}(t); \) |
| 11   | Randomly chosen indexes: \( r_1, r_2, r_3, r_4 \in \{1, 2, \ldots, D\}; \) |
| 12   | \( \text{Trail}_1 = \text{Raindrop}(t) + \Phi \ast (\text{Raindrop}_1(t) - \text{Raindrop}_4(t)); \) |
| 13   | If \( f (\text{Trail}) < f (\text{Raindrop}(t)) \) |
| 14   | New_Raindrop(t) = Trail; |
| 15   | Else |
| 16   | New_Raindrop(t) = Raindrop(t); |
| 17   | End if |
| 18   | \( \text{FES} = \text{FES} + 2; \) |
| 19   | For \( i = 1:N, \) do |
| 20   | Randomly chosen indexes: \( k \in \{1, 2, \ldots, N\}; \) |
| 21   | For \( j = 1:D, \) do |
| 22   | \( c = \text{sign} (a_j - 0.5) \ast \log(\beta_j); \) |
| 23   | If \( c < P \) |
| 24   | Small_Raindrop_i(t) = New_Raindrop_i(t) + c \ast (New_Raindrop_i(t) - \text{Vapor}_i(t)); \) |
| 25   | Else |
| 26   | \( \text{TF} = \text{round} (1 + \text{rand}) \) |
| 27   | Small_Raindrop_i(t) = New_Raindrop_i(t) + \text{rand} \ast (\text{gbest}(t) - \text{TF} \ast \text{mbest}(t)); \) |
| 28   | If \( \text{rand} < P_i \) |
| 29   | Small_Raindrop_i(t) = New_Raindrop_i(t) + \text{s} \ast (Small_Raindrop_i(t) - Small_Raindrop_i(m)); \) |
| 30   | Else |
| 31   | Small_Raindrop_i(t) = New_Raindrop_i(t) |
| 32   | End for |
| 33   | \( \text{FES} = \text{FES} + 1; \) |
| 34   | End for |
| 35   | For \( i = 1:N, \) do |
| 36   | \( \text{Flow}_\text{number} = 0; \) |
| 37   | While \( \text{Flow}_\text{number} \leq \text{Max } \_\text{Flow}_\text{Number}, \) do |
| 38   | \( \lambda = \text{round} (\text{rand}) + 1; \) |
| 39   | Choose \( \text{RP}_{k_1}(t) \) and \( \text{RP}_{k_2}(t) \) from \( \text{RP} \) by the tournament selection procedure; |
| 40   | New_Small_Raindrop_i(t) = Small_Raindrop_i(t) + \hat{d} (t, \lambda); |
| 41   | \( \text{FES} = \text{FES} + 1; \) |
| 42   | New_Small_Raindrop_i(t) = Small_Raindrop_i(t) + \hat{d} (t, \lambda); |
| 43   | Small_Raindrop_i(t) = New_Small_Raindrop_i(t); |
| 44   | \( \text{Flow}_\text{number} = \text{Flow}_\text{number} + 1; \) |
| 45   | Else |
| 46   | Break; |
| 47   | End if |
| 48   | End while |
| 49   | End for |
| 50   | Update raindrop pool \( \text{RP}; \) |
| 51   | Update population \( \text{Pop} (t+1) = \text{Select} (\text{Pop} (t); \text{Small}_\text{Raindrop}(t)); \) |
| 52   | \( t = t + 1; \) |
| 53   | End while |
| 54   | Output: the individual with the smallest objective function value in the population. |
**Table 4**: Definitions and optimum of benchmark functions for evaluating optimization algorithms [11].

| Name | Detailed definition | Optimum |
|------|---------------------|---------|
| \(F_1\) | Shifted Schwefel’s problem 1.2 (first version for a second revision Schwefel’s function) | 0 |
| \(F_2\) | Shifted rotated high conditioned elliptic function | 0 |
| \(F_3\) | Shifted Schwefel’s problem 1.2 with noise in fitness | 0 |
| \(F_4\) | Schwefel’s problem 2.6 with global optimum on bounds | 0 |
| \(F_5\) | Shifted Rastrigin’s function | 0 |
| \(F_6\) | Shifted rotated Rastrigin’s function | 0 |
Figure 5: Plots of benchmark functions F1–F6 ($D = 2$).
Table 5: Results of BTLARA based on different populations over 25 independent runs on F1–F6 test functions on 30 dimensions with 300,000 FEs.

| Function | Result | N = 10 | N = 30 | N = 50 | N = 70 | N = 90 |
|----------|--------|--------|--------|--------|--------|--------|
| F1       | $F_{\text{mean}}$ | 1.98E-14 | 5.30E-15 | 5.49E-15 | 8.60E-15 | 9.93E-15 |
|          | SD     | 2.06E-09 | 2.59E-09 | 6.04E-09 | 7.15E-09 | 6.33E-09 |
| F2       | $F_{\text{mean}}$ | 9.44E+05 | 4.95E+05 | 5.18E+05 | 5.18E+05 | 8.22E+05 |
|          | SD     | 6.66E+05 | 5.28E+05 | 4.77E+05 | 4.53E+05 | 3.78E+05 |
| F3       | $F_{\text{mean}}$ | 5.31E-03 | 4.18E-04 | 4.19E-04 | 5.38E-04 | 8.55E-04 |
|          | SD     | 7.00E-02 | 6.19E-02 | 7.66E-02 | 4.38E-02 | 6.00E-02 |
| F4       | $F_{\text{mean}}$ | 1.09E+02 | 1.00E+02 | 1.01E+02 | 2.05E+02 | 2.26E+02 |
|          | SD     | 6.05E+01 | 5.00E+01 | 3.22E+01 | 3.08E+01 | 4.88E+01 |
| F5       | $F_{\text{mean}}$ | 2.53E-11 | 1.85E-12 | 3.62E-12 | 3.65E-12 | 8.87E-11 |
|          | SD     | 2.54E+01 | 1.53E+01 | 1.44E+01 | 2.51E+01 | 1.63E+01 |
| F6       | $F_{\text{mean}}$ | 5.36E+01 | 5.36E+01 | 5.36E+01 | 5.36E+01 | 5.36E+01 |
|          | SD     | 3.11E-01 | 6.28E-01 | 4.78E-01 | 6.42E-01 | 2.25E-01 |

Table 6: Results of four algorithms for 25 independent runs on 6 test functions of 30 dimensions.

| Function | Result | TLBO | ARA | TLARA | BTLARA |
|----------|--------|------|-----|-------|--------|
| F1       | $F_{\text{mean}}$ | 5.30E-10 | 5.30E-12 | 5.30E-13 | 5.30E-15 |
|          | SD     | 9.09E-09 | 2.23E-08 | 8.29E-10 | 1.05E-09 |
| F2       | $F_{\text{mean}}$ | 1.25E+06 | 4.95E+05 | 5.52E+05 | 4.95E+05 |
|          | SD     | 6.28E+05 | 5.17E+05 | 5.28E+05 | 5.28E+05 |
| F3       | $F_{\text{mean}}$ | 1.55E+01 | 5.25E+02 | 9.88E-02 | 4.18E-04 |
|          | SD     | 2.39E+02 | 2.11E-01 | 3.42E-03 | 2.39E-02 |
| F4       | $F_{\text{mean}}$ | 5.00E+03 | 5.00E+03 | 5.08E+02 | 1.00E+02 |
|          | SD     | 8.42E+01 | 6.22E+01 | 7.44E+01 | 5.00E+01 |
| F5       | $F_{\text{mean}}$ | 1.08E+02 | 1.14E+01 | 1.03E+01 | 1.85E-12 |
|          | SD     | 1.65E+01 | 3.61E+01 | 2.97E+01 | 1.53E+01 |
| F6       | $F_{\text{mean}}$ | 1.35E+02 | 1.01E+02 | 9.85E+01 | 5.36E+01 |
|          | SD     | 3.27E-01 | 7.93E-01 | 2.73E-00 | 6.28E-01 |

- , +, and $\approx$ mean that the performance of the corresponding algorithm is much worse than, way better than, and similar to that of BTLARA, respectively.

Figure 6: Continued.
the best performance in accuracy as compared with the others including TLBMO, HSTLBO, TLBODE, and TPLPSO.

4. Application of BTLARA to UAV Route Planning

4.1. Route Planning Problem Model of UAV. Route planning of UAV is a novel low altitude penetration technique in mission planning system which aims to avoid the invading threat and find an optimal or suboptimal flight route to break through the enemy threat environments [18, 19]. Figure 7 shows that the UAV problem is regarded as a D-dimensional function optimization problem.

4.2. Simulation Results and Comparisons Analysis. The threat points of the UAV (unmanned aerial vehicle) and the coordinates of the starting point and the target point are presented in Table 8. The parameters for oral version of TLBO, ARA, and GWO (grey wolf optimization) [20]
algorithms are obtained from the corresponding references. The parameters for the TLARA are the same as those of BTLARA. Then, the typical algorithms such as BTLARA, oral version of TLBO, ARA, TLARA, and GWO algorithms can be applied to the UAV path planning. Figure 8 presents the simulation results. It can be observed from Figure 8 that BTLARA generates a longer optimal route map than the other four algorithms. In addition, it can be also observed that the UAV route obtained by the TLARA, GWO, and BTLARA has avoided all the threat sources and successfully reached the task end. The comparison of these algorithms shows that the BTLARA is capable of obtaining higher-quality navigation trace and convergence and avoiding the threat routes. Furthermore, it converges faster than other algorithms such as TLBO, ARA, TLARA, and GWO.

| Function | Result \( F_{\text{mean}} \) | TLBMO | HSTLBO | TLBODE | TPLPSO | BTLARA |
|----------|--------------------------|-------|--------|--------|--------|--------|
| \( F_1 \) | \( 1.50 \times 10^{-9} \) | \( 8.00 \times 10^{-14} \) | \( 5.30 \times 10^{-11} \) | \( 5.42 \times 10^{-15} \) | \( 5.30 \times 10^{-15} \) |
| \( F_2 \) | \( 1.07 \times 10^{-6} \) | \( 3.35 \times 10^{-6} \) | \( 2.83 \times 10^{-6} \) | \( 5.87 \times 10^{-5} \) | \( 4.95 \times 10^{-5} \) |
| \( F_3 \) | \( 1.33 \times 10^{-2} \) | \( 1.16 \times 10^{-3} \) | \( 9.16 \times 10^{-4} \) | \( 5.20 \times 10^{-4} \) | \( 4.18 \times 10^{-4} \) |
| \( F_4 \) | \( 3.66 \times 10^{-2} \) | \( 6.38 \times 10^{-3} \) | \( 2.77 \times 10^{-5} \) | \( 1.52 \times 10^{-6} \) | \( 2.39 \times 10^{-2} \) |
| \( F_5 \) | \( 1.59 \times 10^{-3} \) | \( 1.37 \times 10^{-2} \) | \( 1.44 \times 10^{-2} \) | \( 1.08 \times 10^{-2} \) | \( 1.00 \times 10^{-2} \) |
| \( F_6 \) | \( 2.34 \times 10^{-11} \) | \( 2.88 \times 10^{-12} \) | \( 1.29 \times 10^{-10} \) | \( 3.00 \times 10^{-12} \) | \( 1.85 \times 10^{-12} \) |

- - , + , and = respectively mean that the performance of the corresponding algorithm is much worse than, significantly better than, and similar to that of BTLARA.

| Start-point | Target-point | (10, 10) | (55, 100) |
|-------------|--------------|----------|-----------|
| Threat-point | Number | Threat-center\threat-radius\threat-level | Number | Threat-center\threat-radius\threat-level | Number | Threat-center\threat-radius\threat-level |
| 1 | (12,40) | 10 \( \times \) 10 | 2 | (32, 68) | 8 \( \times \) 1 | 3 | (36,26) | 12 \( \times \) 2 |
| 4 | (45,50) | 10 \( \times \) 2 | 5 | (55, 80) | 9 \( \times \) 3 |

\( k = 0.5 \)
Figure 8: Continued.
5. Conclusions

In this work, a BTLARA consisting of an artificial raindrop algorithm frame and a Balwin-teaching-learning-based optimization strategy is put forward to address such optimization problems as UAV route planning. The introduction of the ARA framework ensures a better control of swarm exploration/exploitation searches. The teaching-learning-based optimization (TLBO) algorithm accelerates the convergence speed, and the Baldwinian learning operator can simulate the learning mechanism in the learner phase by utilizing the information from within the population to change the search space. The simulation results have demonstrated that the route planned by the proposed method is much smoother, shorter, and more stable than that by the oral version of TLBO, ARA, GWO, and TLARA.
algorithms. It should be pointed out that the research on the BTLARA algorithm is at its initial stage; there are many problems need to be addressed in the further studies.

The future work will focus on the practical application of the BTLARA algorithm in UAV path planning and extending two-dimensional UAV to three-dimensional UAV path planning.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (no. 11704302) and the Special Scientific Research Project of Education Department of Shaanxi Province (no.20JK0691).

References

[1] J. Kennedy and R. C. Eberhart, “Particle swarm optimization,” in Proceedings of the IEEE International Conference on Neural Networks, vol. 107, no. 12, pp. 1–4, IEEE Service Center Piscataway Nj N, Perth, Australia, December 1995.

[2] R. Storn and K. Price, “Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[3] Z. Sen and Z. Yongquan, “Grey Wolf optimizer with ranking-based mutation operator for IIR model identification,” Chinese Journal of Electronics, vol. 27, no. 5, pp. 1071–1079, 2018.

[4] Q. Jiang, L. Wang, and X. Hei, “Parameter identification of chaotic systems using artificial raindrop algorithm,” Journal of Computational Science, vol. 8, pp. 20–31, 2015.

[5] Q. Jiang, L. Wang, X. Hei, G. Yu, and Y. Lin, “The performance comparison of a new version of artificial raindrop algorithm on global numerical optimization,” Neurocomputing, vol. 179, no. 1, pp. 1–25, 2016.

[6] S. Mahata, S. K. Saha, R. Kar et al., “Artificial raindrop algorithm-based design of wideband IIR fractional order digital integrators,” Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 2017.

[7] Q. Jiang, L. Wang, Y. Lin, X. Hei, G. Yu, and X. Lu, “An efficient multi-objective artificial raindrop algorithm and its application to dynamic optimization problems in chemical processes,” Applied Soft Computing, vol. 58, pp. 354–377, 2017.

[8] R. V. Rao, V. J. Savsani, and D. P. Vakharia, “Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems,” Computer-Aided Design, vol. 43, no. 3, pp. 303–315, 2011.

[9] R. V. Rao and V. Patel, “An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems,” International Journal of Industrial Engineering Computations, vol. 3, no. 4, pp. 535–560, 2012.

[10] M. Gong, L. Jiao, and L. Zhang, “Baldwinian learning in clonal selection algorithm for optimization,” Information Sciences, vol. 180, no. 8, pp. 1218–1236, 2010.

[11] P. N. Suganthan, N. Hansen, J. J. Liang et al., “Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,” Natural Computing, pp. 341–357, 2005.

[12] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” IEEE Transactions on Evolutionary Computation, vol. 10, no. 3, pp. 281–295, 2006.

[13] W. Yong, “Differential evolution with composite trial vector generation strategies and control parameters,” IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 55–66, 2011.

[14] Q. Zhang and G. Yu, “A hybrid bird mating optimizer algorithm with teaching-learning-based optimization for global numerical optimization,” Statistics Optimisation and Information Computing, vol. 3, no. 1, 2015.

[15] S. Tuo, “HSTLBO: A hybrid algorithm based on Harmony Search and Teaching-Learning-Based Optimization for complex high-dimensional optimization problems,” PLoS ONE, vol. 12, no. 4, 2017.

[16] L. Wang, F. Zou, X. Hei et al., “A hybridization of teaching-learning-based optimization and differential evolution for chaotic time series prediction,” Neural Computing and Applications, vol. 25, no. 6, pp. 1407–1422, 2014.

[17] W. H. Lim and N. A. Mat Isa, “Teaching and peer-learning particle swarm optimization,” Applied Soft Computing, vol. 18, pp. 39–58, 2014.

[18] Y. E. Wen and H. D. Fan, “Research on mission planning system key techniques of UCAV,” Journal of Naval Aeronautical Engineering Institute, 2007.

[19] Z. Cheng, Y. Sun, and Y. Liu, “Path planning based on immune genetic algorithm for UAV,” in Proceedings of the International Conference on Electric Information and Control Engineering, pp. 590–593, IEEE, Wuhyan, China, April 2011.

[20] H. Rezaei, O. Bozorg-Haddad, and X. Chu, Grey Wolf Optimization (GWO) Algorithm, 2018.