symmetry

MDPI

Article

The Gaussian Mutational Barebone Dragonfly Algorithm: From Design to Analysis

Li Yuan [1], Fangjun Kuang [2], Siyang Zhang [2,*] and Huiling Chen [3,*]

[1] School of Artificial Intelligence, Beijing Institute of Economics and Management, Beijing 100102, China; yuanli@biem.edu.cn
[2] School of Information Engineering, Wenzhou Business College, Wenzhou 325035, China; kfj@wzbc.edu.cn
[3] College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China
* Correspondence: zsy@wzbc.edu.cn (S.Z.); chenhuiling.jlu@gmail.com (H.C.)

Abstract: The dragonfly algorithm is a swarm intelligence optimization algorithm based on simulating the swarming behavior of dragonfly individuals. An efficient algorithm must have a symmetry of information between the participating entities. An improved dragonfly algorithm is proposed in this paper to further improve the global searching ability and the convergence speed of DA. The improved DA is named GGBDA, which adds Gaussian mutation and Gaussian barebone on the basis of DA. Gaussian mutation can randomly update the individual positions to avoid the algorithm falling into a local optimal solution. Gaussian barebone can quicken the convergent speed and strengthen local exploitation capacities. Enhancing algorithm efficiency relative to the symmetric concept is a critical challenge in the field of engineering design. To verify the superiorities of GGBDA, this paper sets 30 benchmark functions, which are taken from CEC2014 and 4 engineering design problems to compare GGBDA with other algorithms. The experimental result show that the Gaussian mutation and Gaussian barebone can effectively improve the performance of DA. The proposed GGBDA, similar to the DA, presents improvements in global optimization competence, search accuracy, and convergence performance.

Keywords: dragonfly algorithm; swarm intelligence; Gaussian mutation; Gaussian barebone; engineering design problem

Citation: Yuan, L.; Kuang, F.; Zhang, S.; Chen, H. The Gaussian Mutational Barebone Dragonfly Algorithm: From Design to Analysis. *Symmetry* **2022**, *14*, 331. https://doi.org/10.3390/sym14020331

Academic Editor: Jan Awrejcewicz

Received: 5 January 2022
Accepted: 1 February 2022
Published: 6 February 2022

1. Introduction

The swarm intelligence optimization algorithm (SIOA) mainly simulates biological individuals' group behavior, such as cooperation and competition, to obtain the optimal solution to complex problems. Moreover, SIOA has the benefit of an uncomplicated structure, few parameters, and uncomplicated implementations [1]. To date, varies of SIOA had been proposed by domestic and foreign scholars, namely the whale optimization algorithm (WOA) [2,3]; differential evolution [4] (DE); genetic algorithm (GA) [5]; ant colony optimization (ACO) [6,7]; particle swarm optimization (PSO) [8,9]; firefly algorithm (FA) [10]; fruit fly optimization algorithm (FOA) [11–13]; slime mould algorithm (SMA) [14]; moth flame optimization (MFO) [15–17]; grey wolf optimizer (GWO) [18,19]; bat algorithm (BA) [20,21]; grasshopper optimization algorithm (GOA) [22,23]; Harris hawks optimization (HHO) [24]; colony predation algorithm (CPA) [25]; hunger games search (HGS) [26]; Runge–Kutta optimizer (RUN) [27] and weighted mean of vectors (INFO) [28].

SIOA found its application in many fields, namely expensive optimization problems [29,30]; performance optimization [31]; object tracking [32,33]; multi-objective or many optimization problems [34–36]; traveling salesman problem [37]; neural network training [38]; scheduling problems [39]; big data optimization problems [40]; fault diagnosis of rolling bearings [41]; evolving deep convolutional neural networks [42]; gate resource allocation [43,44], and combination optimization problems [45]. The dragonfly algorithm (DA) is a population-based heuristic search algorithm that was first proposed by Mirjalili, S. [46] in 2015 and has since gained widespread adoption. It has a high level of performance and a broad range of applications in real life. Many applications, including parameter optimization [47], feature selection [48], load balancing [49], modeling [50], and others [51], have been effectively implemented using it. Many trials with complicated, high-dimensional, and multi-modal functions, on the other hand, demonstrated that DA had some drawbacks in some situations. For example, the DA lacks internal memory, has a poor convergence time, and is prone to falling into the local optimum when running in the background. As a result, several researchers are putting forth an attempt to increase the DA.

1.1. Related Works

When it comes to solving the challenge of numerical optimization, Sree Ranjini and colleagues [52] suggested a new memory-based hybrid DA (HMDA). The drawback of the DA was remedied by combining the advantages of the DA and the PSO together. Moreover, N. S. et al. [53] integrated the crow search algorithm (CSA) with the D-Crow optimization algorithm, presented a D-Crow optimization method, and applied this algorithm to optimize the configuration of virtual machines migrating. A method combining the dynamic analysis and the pattern search algorithm was presented by Khadanga and colleagues [54] to improve the performance and optimize the controller settings, in order to improve the control efficiency of the frequency of Microgrid. Using a trained multi-layer perceptron, Ghanem et al. [55] developed a novel hybridized metaheuristic method with improved properties in terms of attaining the best optimal value, convergence speed, avoiding local minima, and accuracy compared to previous algorithms. They created a hybrid algorithm by combining the artificial bee colony (ABC) algorithm with the distributed algorithm (DA). Shilaja and colleagues [56] used a combination of the enhanced grey wolf optimization and dynamic programming to handle the nonlinearity problems. Furthermore, it has been demonstrated to be more efficient than the conventional method. Using a dragonfly-based clustering method, CAVDO, Aadil et al. [57] proposed a solution to difficulties associated with the Internet of vehicles, such as scalability, dynamic topology changes, and finding the shortest path for routing. For the DA to be more random, Aci and colleagues [58] used the Brownian motion, which they found to be more effective. Furthermore, the results of the experiments revealed that the new DA had superior properties when compared to the old algorithm. Bao and colleagues [59] proposed a new DA that was changed using opposition-based learning. It also had a faster convergence time and a more balanced exploration–exploitation ratio, according to the results of the studies. Li et al. [60] improved the performance of DA by incorporating the adaptive learning factor and differential evolution (DE) approach into the algorithm. Sayed and colleagues [61] proposed a novel chaotic DA (CDA). In order to increase the DA, the researchers included chaotic maps in the searching iterations of the algorithm. Conforming to the experimental findings, CDA outperformed the control group in classification performance and was capable of identifying more suitable feature subsets.

Mafarja et al. [62] collected eight transfer functions (s-type function and v-type function) in BDA for evaluation, and proposed the time-varying s-type BDA, which made the algorithm have a high probability of changing the element position in the early optimization period, but with a low probability in the late optimization period. Hariharan et al. [63] proposed an improved binary dragonfly optimization algorithm (IBDFO) to solve the dimension problem and combined it with a feature extraction based on a wavelet packet to improve the accuracy of identifying the type of infant crying. Zhang et al. [64] used the DA to improve the prediction accuracy of the support vector machine (SVM) to obtain the optimal combination of parameters, and proposed the DA-SVM model to realize the short-term load prediction of the micro grid. Yuan et al. [65] tended to obtain an algorithm with better exploration capability as they combined the DA with the Coulomb force search strategy (CFSS). The resultant algorithm gained both a high accuracy and a remarkably improved convergence rate. Zhang et al. [66] quantized dragonfly behaviors to improve the search efficiency of the DA to obtain a quantized dragonfly algorithm (QDA). Furthermore, they put forward a new electric load forecasting model, based on the complete ensemble empirical mode decomposition adaptive noise, QDA, and support vector regression model, to accurately forecast the electric load. Suresh et al. [67] adopted the DA as the optimization algorithm to solve static economic dispatch incorporating solar energy. Based on the modified dragonfly algorithm (MDA) and bat search algorithm (BSA), Sureshkumar et al. [68] put forward a new method that adopted the MDABSA technique to control power flow more efficiently. In this method, MDA was used to develop the control signals of the voltage source. Xie et al. [69] adopted the DA to create a cancer classification algorithm. Furthermore, the comparative experiments proved it had a higher classification accuracy on cancer datasets. Xu et al. [70] adopted the DA and DE for color image segmentation. In this method, the DA was used for global search, and DE was used for local search.

1.2. Needs for Research

However, despite the fact that the literature discussed above made significant advances to the DA, it is not optimal enough to stabilize the algorithm's exploration and exploitation capabilities. With the goal of further improving the exploration and exploitation exactness of the DA, as well as avoiding falling into the local optimum, this work proposes an upgraded DA that incorporates Gaussian mutation and Gaussian barebone to further improve these aspects. With the use of Gaussian mutation, we were able to update the dragonfly's unique location while also improving the global search capabilities. Additionally, the Gaussian barebone was used to increase the local exploitation capabilities as well as the speed with which the searches could be conducted. The results of the simulations demonstrated that the algorithm's accomplishments were superior to those of the original DA, and that its global optimization capabilities, search accuracy, and convergence performance were all greatly enhanced as a consequence. In summary, the innovations and contributions of this paper are as follows.

- An improved dragonfly algorithm (GGBDA) is proposed in this paper to further improve the global searching ability and the convergence speed of DA.
- GGBDA achieves a great improvement in the ability of exploitation and exploration.
- The performance of GGBDA is verified by comparison with some excellent algorithms.
- GGBDA is applied to optimize the engineering optimization problems.

The following is a summary of the rest of this article. Section 2 introduces the DA; Section 3 describes the enhanced DA based on Gaussian mutation and Gaussian barebone; Section 4 presents the experimental findings of the benchmark functions; and Section 5 concludes the paper and provides an overview of the previous work as well as a forecast for future work.

2. Materials and Methods

2.1. Dragonfly Algorithm (DA)

DA was inspired by two states of idealized behaviors of dragonflies in nature. There are three principles in the core mathematical backgrounds of this method.

Separation aims to prevent search individuals from collisions with others in a static state within a partial range. The following is the calculation function:

$$S_i = -\sum_{j=1}^{N} X - X_j \tag{1}$$

where X is the position agents, X_j is j-th neighboring individual's position, and N is neighboring individuals' number.

Alignment is aimed at matching velocity between individuals within a partial range. The following is the calculation function:

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \tag{2}$$

where V_j is the j-th velocity of the neighboring individual.

Cohesion is aimed at making individuals move closer towards the center of swarm aggregation. The following is the calculation function:

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \tag{3}$$

where X is the current individual's position, N is neighborhoods' number, and X_j is j-th neighboring individual's position.

The following is the attraction towards a food source:

$$F_i = X^+ - X \tag{4}$$

where X is the current individual's position and X^+ is he food source's position.

The following is the distraction outwards an enemy source:

$$E_i = X^- + X \tag{5}$$

where X is the current individual's position and X^- is the enemy's position.

Step (ΔX) and position(X) are prerequisites to update and record the location of agents in the search domain. The step vector can be considered as the velocity vector in PSO. It is the direction of the agents' motion. The following is the calculation function of the position vector:

$$\Delta X_{t+1} = sS_i + aA_i + cC_i + fF_i + eE_i + w\Delta_t \tag{6}$$

where S_i, A_i, C_i, F_i, E_i indicates the separation, alignment, cohesion, food source and an enemy of the i-th individual's position. s, a, c, f, e represent the weights, w is the inertia weight, i is the i-th individual, and t is the number of the current iteration. The following is the calculation function of the position vector:

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{7}$$

Search agents have some deficiencies in terms of random behavior and exploration ability, and they also lack adjacent solutions. Therefore, Levy flight-based patterns are used to update the position of agents. The following is the function to update location:

$$X_{t+1} = X_t + Levy(d) \times X_t \tag{8}$$

where t is the current iteration number and d is the dimension of the position vector.

2.2. Gaussian Mutation

To improve the performance of DA, this paper used the Gaussian mutation to update the individual position of the dragonfly. Gaussian mutation has applied to many optimizers [3,16,71,72]. The following is the mutation function of the Gaussian mutation:

$$temp = X_j * (1 + k) \tag{9}$$

where X is the position agents, temp is a temporary individual position, X_j is j-th neighboring individual position, N is neighboring individuals' number, and k is a random number between 0 and 1.

After updating the individual position of the dragonfly with this mutation function, whether the result of the Gaussian mutation is better than the previous result needs to be verified. If the temporary individual position can obtain a better result, it will be used as the new individual position of the dragonfly. With the population iterates, the DA may fall into local optimum. The Gaussian mutation has randomness, thereby quickening the scouting speed, avoiding slipping into the local optimum effectively, improving the global optimization capacity, and eventually obtaining the global optimum or a satisfactory solution.

2.3. Gaussian Barebone Mechanism

The speed of scouting for the optimal solution is a significant indicator of the performance of the algorithm. However, in the iteration, the scouting speed of the DA is dissatisfactory; thereby, this paper employed a Gaussian barebone to improve it. The Gaussian barebone mechanism hast been shown great potential in other optimizers [71,72]. The Gaussian barebone mechanism could help the DA scout the global optimum faster and more effectively by gathering individuals into a food source. There are two methods to gather individuals. The first method calculates the middle position between the food source and individual's position and the distance between them. Then, it generates a random position where the values of each dimension are normally distributed based on the two calculated variables. The second method obtains the distances for each dimension of two random individuals. Additionally, it uses them and the position of the food source to calculate a new position. The following is the function:

$$V_{i,j} = \begin{cases} normal(mu + sigma), rand() < CR \\ FP_j + k * \left(X_{k1,j} - X_{k2,j}\right), rand() \geq CR \end{cases} \tag{10}$$

where CR is a freely settable parameter; *rand* is a random number between 0 and 1; $V_{i,j}$ is a new temporary position; mu is the middle position between the food source's position and X_j; sigma is the distance between the j-th dimension of the i-th neighboring individual and the j-th dimension of food source; the normrnd function generates random numbers that follow a normal distribution with the mu parameter representing the mean value and the sigma parameter representing standard deviation; FP_j is the j-th dimension of food source; k is a random number; and $X_{k1,j}$ and $X_{k2,j}$ are j-th dimension of two random individuals in the population.

3. Proposed Method

The DA lacks internal memory, has a slow convergence speed, and quickly falls into the local optimum. As a result of these defects, this paper puts forward a new DA improved by Gaussian mutation and a Gaussian barebone named GGBDA. It uses the Gaussian barebone to gather individuals to food to quicken the speed of scouting the optimal solution and strengthen local exploitation capacities. It can update the individuals' positions based on the position of the food source. However, Gaussian barebone could make the population fall into local optimums. Therefore, this paper employs the Gaussian mutation to improve the global search capacities, search accuracy, and convergence performance by preventing it from trapping into local optimums.

The Gaussian mutation is mainly used to randomly update individuals' positions to escape the local optimums based on the Gaussian mutation function. The flowchart of the improved DA is shown in Figure 1. And The pseudocode of GGBDA is shown in Algorithm 1.

Algorithm 1. Pseudocode of GGBDA

Begin
  Initialize the dragonflies' population $X_i(i = 1, 2, \ldots, n)$
  Initialize the step vectors $\Delta X_i(i = 1, 2, \ldots, n)$
  while the end condition is not satisfied
    Calculate the population fitness of all the dragonflies
    Update the food source and enemy
    Update w, s, a, c, f, and e
    Calculate S, A, C, F, and E by Equations (1)–(5)
    Update the neighboring radius
    if a dragonfly has at least one neighboring dragonfly
      Update the velocity and vector by Equation (6)
      Update the position vector by Equation (7)
    else
      Update the position vector by Equation (8)
    end if
    Check and correct the new position according to the boundaries of the variables
    Update with the Gaussian mutation and Gaussian barebone
  end while
End

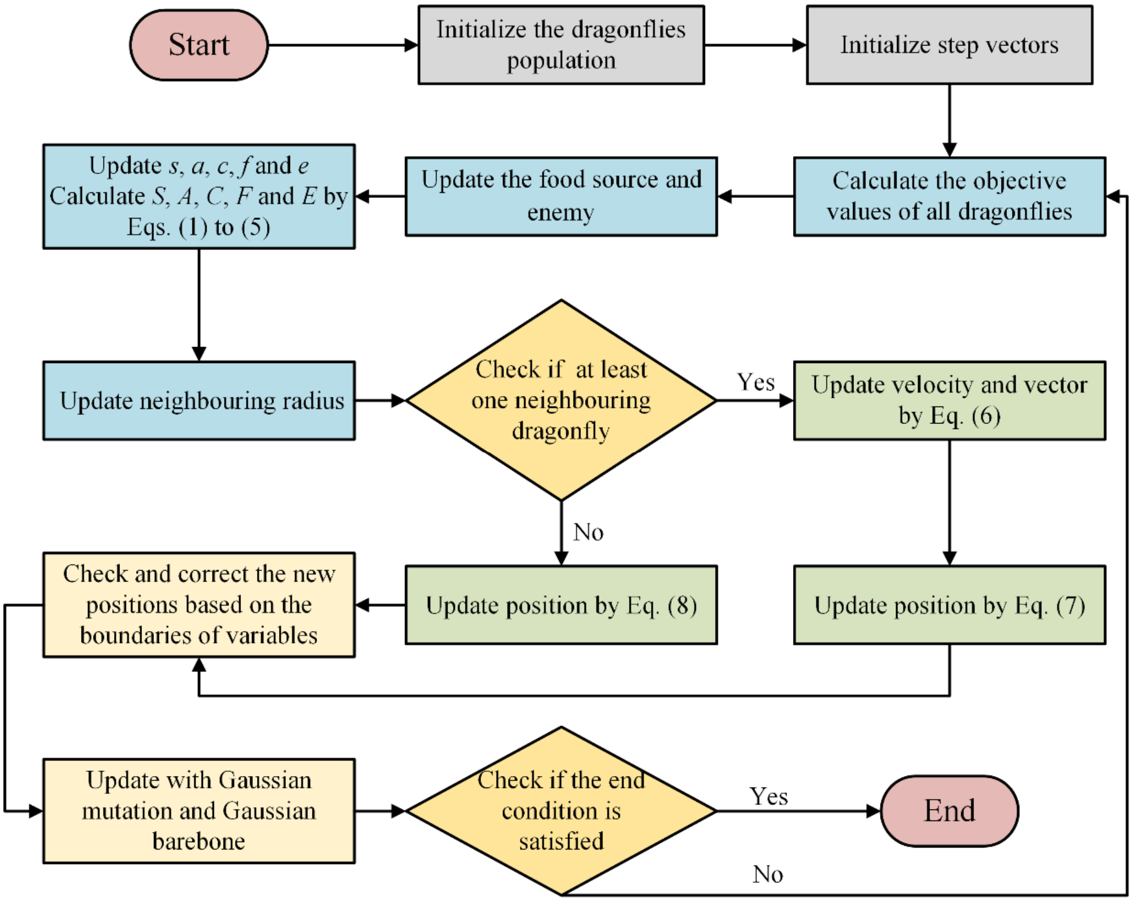

Figure 1. Flowchart of GGBDA.

4. Experimental Results

In this part, the GGBDA was evaluated on CEC2014 benchmarks and practical engineering problems. To obtain unbiased results, all the experiments were carried out in the same environments, and the maximum number of iterations and the population size were set to 500 and 30, respectively. Each algorithm was run 30 times independently on each function to decrease the weight of unpredictability. Regarding the parameters that affect the algorithms involved in the comparison, we adopted the same values as in the original paper. In this paper, the average value and standard deviation of the experimental results of the optimization function were used to evaluate and analyze the potential of related technologies. To show the experimental result intuitively, the best values of each function are shown in bold.

4.1. Benchmark Functions

To compare the proposed algorithm and other algorithms, this experiment used 30 classical functions, including unimodal functions, multi-modal functions, hybrid functions, and composition functions.

These 30 functions are all taken from CEC2014 [73]. Thirty different types of benchmarks can more comprehensively estimate the performance of the proposed algorithm. The details of the thirty benchmarks are listed in Table 1.

Table 1. Description of the 30 benchmark functions.

ID	Function Equation	Search Range	Optimum Value
	CEC 2014 Unimodal Functions		
F1	Rotated High Conditioned Elliptic Function	[−100,100]	$f_1\{X_{min}\} = 100$
F2	Rotated Bent Cigar Function	[−100,100]	$f_2\{X_{min}\} = 200$
F3	Rotated Discus Function	[−100,100]	$f_3\{X_{min}\} = 300$
	CEC 2014 Simple Multi-Modal Functions		
F4	Shifted and Rotated Rosenbrock Function	[−100,100]	$f_4\{X_{min}\} = 400$
F5	Shifted and Rotated Ackley Function	[−100,100]	$f_5\{X_{min}\} = 500$
F6	Shifted and Rotated Weierstrass Function	[−100,100]	$f_6\{X_{min}\} = 600$
F7	Shifted and Rotated Griewank Function	[−100,100]	$f_7\{X_{min}\} = 700$
F8	Shifted Rastrigin Function	[−100,100]	$f_8\{X_{min}\} = 800$
F9	Shifted and Rotated Rastrigin Function	[−100,100]	$f_9\{X_{min}\} = 900$
F10	Shifted Schwefel Function	[−100,100]	$f_{10}\{X_{min}\} = 1000$
F11	Shifted and Rotated Schwefel Function	[−100,100]	$f_{11}\{X_{min}\} = 1100$
F12	Shifted and Rotated Katsuura Function	[−100,100]	$f_{12}\{X_{min}\} = 1200$
F13	Shifted and Rotated HappyCat Function	[−100,100]	$f_{13}\{X_{min}\} = 1300$
F14	Shifted and Rotated HGBat Function	[−100,100]	$f_{14}\{X_{min}\} = 1400$
F15	Shifted and Rotated Expanded Griewank Plus Rosenbrock Function	[−100,100]	$f_{15}\{X_{min}\} = 1500$
F16	Shifted and Rotated Expanded Scaffer F6 Function	[−100,100]	$f_{16}\{X_{min}\} = 1600$
	CEC 2014 Hybrid Functions		
F17	Hybrid Function 1 (N = 3)	[−100,100]	$f_{17}\{X_{min}\} = 1700$
F18	Hybrid Function 2 (N = 3)	[−100,100]	$f_{18}\{X_{min}\} = 1800$
F19	Hybrid Function 3 (N = 4)	[−100,100]	$f_{19}\{X_{min}\} = 1900$
F20	Hybrid Function 4 (N = 4)	[−100,100]	$f_{20}\{X_{min}\} = 2000$
F21	Hybrid Function 5 (N = 5)	[−100,100]	$f_{21}\{X_{min}\} = 2100$
F22	Hybrid Function 6 (N = 5)	[−100,100]	$f_{22}\{X_{min}\} = 2200$

Table 1. *Cont.*

ID	Function Equation	Search Range	Optimum Value
CEC 2014 Composition Functions			
F23	Composition Function 1 (N = 5)	[−100,100]	$f_{23}\{X_{min}\} = 2300$
F24	Composition Function 2 (N = 3)	[−100,100]	$f_{24}\{X_{min}\} = 2400$
F25	Composition Function 3 (N = 3)	[−100,100]	$f_{25}\{X_{min}\} = 2500$
F26	Composition Function 4 (N = 5)	[−100,100]	$f_{26}\{X_{min}\} = 2600$
F27	Composition Function 5 (N = 5)	[−100,100]	$f_{27}\{X_{min}\} = 2700$
F28	Composition Function 6 (N = 5)	[−100,100]	$f_{28}\{X_{min}\} = 2800$
F29	Composition Function 7 (N = 3)	[−100,100]	$f_{29}\{X_{min}\} = 2900$
F30	Composition Function 8 (N = 3)	[−100,100]	$f_{30}\{X_{min}\} = 3000$

4.2. Comparison with Classical Algorithms

In order to validate the effectiveness of the improved GGBDA, there are some representative algorithms employed for comparison: OBSCA [74], m_SCA [75], SCADE [76], ASCA_PSO [77], ACWOA [78], MFO [15], SCA [79], FA [80], and DA.

In the experimental part, the parameter values of the compared algorithms were set, as shown in Table 2. To ensure the fairness of the experiments as far as possible, the experimental environment of algorithms stayed the same. The experimentations used 30D classical functions for comparing the proposed method and other rivals. Table 3 recorded the experimental results on 30D. Each algorithm ran independently 30 times. The average (Ave) and standard deviation (Std) of the optimal solutions obtained are shown in these tables. "AVR" expresses the average of the algorithm's ranking results on all functions. In this experiment, the maximum number of iterations and the population size (Pop) were set to 1000 and 30. Each algorithm was performed in every function with 30 dimensions for the test of scalabilities, respectively. The symbol "+/=/−" refers to whether the performance of GGBDA is greater, equal, or worse than other algorithms compared.

Table 2. Parameter settings of the algorithms in the experiment.

Algorithms	Pop	Maximum Iterations	Others
GGBDA	30	1000	$w \in [0.9\ 0.2]$; $s = 0.1$; $a = 0.1$; $c = 0.7$; $f = 1$; $e = 1$
OBSCA	30	1000	$a = 2$
m_SCA	30	1000	$a = 2$
SCADE	30	1000	$a = 2$; $CR = 0.8$; $LSF = 0.8$; $USF = 0.2$
ASCA_PSO	30	1000	$M = 4$; $N = 9$; $Vmax = 6$; $wMax = 0.9$; $wMin = 0.2$; $c_1 = 2$; $c_2 = 2$;
ACWOA	30	1000	$B = 1$
MFO	30	1000	$B = 1$
SCA	30	1000	$a = 2$
FA	30	1000	$alpha = 0.5$; $betamin = 0.2$; $gamma = 1$;
DA	30	1000	$w \in [0.9\ 0.2]$; $s = 0.1$; $a = 0.1$; $c = 0.7$; $f = 1$; $e = 1$
GGBDA	30	1000	$w \in [0.9\ 0.2]$; $s = 0.1$; $a = 0.1$; $c = 0.7$; $f = 1$; $e = 1$

4.2.1. Results on 30D Functions

F1–F7 do not have local optimal solutions. They are very suitable for measuring the exploration competence of the algorithm. In F2, F3, and F6, the results of GGBDA are far superior to all the others. Furthermore, in the rest functions, the results of GGBDA are better than most comparison algorithms. The results of F1–F7 show that GGBDA has an advantage over other algorithms in the ability to explore in the unimodal locality.

Table 3. Experimental results of the 30 dimensions (30Ds).

	F1		F2		F3	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	3.3428×10^{7}	2.3615×10^{7}	5.7799×10^{7}	1.32174×10^{7}	2.2502×10^{3}	1.0237×10^{3}
OBSCA	3.8095×10^{8}	1.2188×10^{8}	2.4577×10^{10}	3.9982×10^{9}	5.1744×10^{4}	7.3043×10^{3}
m_SCA	7.2766×10^{7}	3.9039×10^{7}	6.4809×10^{9}	2.7501×10^{9}	2.6967×10^{4}	7.4237×10^{3}
SCADE	4.3235×10^{8}	1.0258×10^{8}	2.9383×10^{10}	4.9065×10^{9}	5.3542×10^{4}	6.3130×10^{3}
ASCA_PSO	1.5733×10^{7}	7.8447×10^{6}	5.7234×10^{8}	7.6338×10^{8}	2.0200×10^{4}	5.3347×10^{3}
ACWOA	1.3598×10^{8}	5.9536×10^{7}	7.6372×10^{9}	3.3593×10^{9}	5.1123×10^{4}	8.7487×10^{3}
MFO	7.0131×10^{7}	8.4361×10^{7}	1.3759×10^{10}	7.4030×10^{9}	9.8036×10^{4}	6.1005×10^{4}
SCA	2.2033×10^{8}	7.5726×10^{7}	1.6600×10^{10}	3.2678×10^{9}	3.5442×10^{4}	6.2559×10^{3}
FA	2.5375×10^{8}	5.0283×10^{7}	1.5600×10^{10}	2.0292×10^{9}	6.3396×10^{4}	9.7529×10^{3}
DA	8.11892×10^{8}	4.3376×10^{8}	2.2171×10^{10}	2.2383×10^{10}	5.9107×10^{4}	1.5850×10^{4}
	F4		F5		F6	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	5.9527×10^{2}	8.7643×10^{1}	5.2093×10^{2}	5.5872×10^{-2}	6.2033×10^{2}	4.0175×10^{0}
OBSCA	2.4186×10^{3}	8.0598×10^{2}	5.2097×10^{2}	4.9147×10^{-2}	6.3202×10^{2}	1.7351×10^{0}
m_SCA	7.5730×10^{2}	1.0198×10^{2}	5.2061×10^{2}	1.4096×10^{-1}	6.2114×10^{2}	3.2807×10^{0}
SCADE	2.4370×10^{3}	5.6808×10^{2}	5.2094×10^{2}	6.3764×10^{-2}	6.3419×10^{2}	2.3689×10^{0}
ASCA_PSO	5.7201×10^{2}	1.5123×10^{2}	5.2094×10^{2}	4.1898×10^{-2}	6.2512×10^{2}	3.2965×10^{0}
ACWOA	1.0827×10^{3}	2.3891×10^{2}	5.2083×10^{2}	1.2246×10^{-1}	6.3454×10^{2}	3.1803×10^{0}
MFO	1.4154×10^{3}	1.1476×10^{3}	5.2026×10^{2}	2.0197×10^{-1}	6.2398×10^{2}	3.0738×10^{0}
SCA	1.4155×10^{3}	3.0882×10^{2}	5.2093×10^{2}	4.4333×10^{-2}	6.3375×10^{2}	2.6530×10^{0}
FA	1.5386×10^{3}	1.7232×10^{2}	5.2095×10^{2}	5.1811×10^{-2}	6.3392×10^{2}	6.4751×10^{-1}
DA	7.2148×10^{3}	5.0944×10^{3}	5.2096×10^{2}	3.8523×10^{-2}	6.3831×10^{2}	3.8669×10^{0}
	F7		F8		F9	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	7.0154×10^{2}	1.4093×10^{-1}	8.8953×10^{2}	1.4755×10^{1}	1.0718×10^{3}	3.5377×10^{1}
OBSCA	9.1188×10^{2}	3.2095×10^{1}	1.0564×10^{3}	1.5937×10^{1}	1.2007×10^{3}	1.8331×10^{1}
m_SCA	7.5112×10^{2}	2.7312×10^{1}	9.4797×10^{2}	2.0587×10^{1}	1.0570×10^{3}	2.4289×10^{1}
SCADE	8.9697×10^{2}	3.1487×10^{1}	1.0680×10^{3}	1.3258×10^{1}	1.2072×10^{3}	1.7261×10^{1}
ASCA_PSO	7.1122×10^{2}	1.5224×10^{1}	9.5707×10^{2}	2.6319×10^{1}	1.1114×10^{3}	3.7255×10^{1}
ACWOA	7.2872×10^{2}	1.6207×10^{1}	9.9483×10^{2}	2.5768×10^{1}	1.1277×10^{3}	2.1651×10^{1}
MFO	8.1621×10^{2}	7.0326×10^{1}	9.3286×10^{2}	3.1243×10^{1}	1.1154×10^{3}	4.2025×10^{1}
SCA	8.3820×10^{2}	2.6572×10^{1}	1.0372×10^{3}	1.6583×10^{1}	1.1745×10^{3}	1.5443×10^{1}
FA	8.3255×10^{2}	9.9991×10^{0}	1.0236×10^{3}	1.5241×10^{1}	1.1575×10^{3}	8.8945×10^{0}
DA	1.0796×10^{3}	2.5224×10^{2}	1.0603×10^{3}	8.6112×10^{1}	1.1875×10^{3}	4.3320×10^{1}

Table 3. *Cont.*

	F10		F11		F12	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	2.1632×10^{3}	4.0569×10^{2}	4.3103×10^{3}	5.8058×10^{2}	1.2016×10^{3}	6.4654×10^{-1}
OBSCA	6.1914×10^{3}	3.3800×10^{2}	7.3712×10^{3}	3.8870×10^{2}	1.2022×10^{3}	3.8380×10^{-1}
m_SCA	4.2173×10^{3}	6.7303×10^{2}	4.6926×10^{3}	5.6709×10^{2}	1.2007×10^{3}	2.8914×10^{-1}
SCADE	7.3873×10^{3}	2.0852×10^{2}	8.2043×10^{3}	2.8866×10^{2}	1.2026×10^{3}	2.9637×10^{-1}
ASCA_PSO	5.3236×10^{3}	6.1947×10^{2}	6.0330×10^{3}	1.0051×10^{3}	1.2024×10^{3}	3.2840×10^{-1}
ACWOA	4.3616×10^{3}	9.4361×10^{2}	6.5284×10^{3}	8.8174×10^{2}	1.2018×10^{3}	5.3507×10^{-1}
MFO	4.2961×10^{3}	1.0010×10^{3}	5.2553×10^{3}	5.8399×10^{2}	1.2004×10^{3}	1.6921×10^{-1}
SCA	6.9536×10^{3}	5.2169×10^{2}	8.1744×10^{3}	2.6469×10^{2}	1.2024×10^{3}	2.8490×10^{-1}
FA	7.5877×10^{3}	2.4931×10^{2}	7.8979×10^{3}	2.2794×10^{2}	1.2024×10^{3}	3.1798×10^{-1}
DA	7.8983×10^{3}	8.8564×10^{2}	8.2497×10^{3}	7.2246×10^{2}	1.2024×10^{3}	3.9165×10^{-1}
	F13		F14		F15	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	1.3006×10^{3}	1.1023×10^{-1}	1.4003×10^{3}	4.9476×10^{-2}	1.5246×10^{3}	3.8702×10^{0}
OBSCA	1.3037×10^{3}	4.2284×10^{-1}	1.4669×10^{3}	1.1727×10^{1}	1.7547×10^{4}	9.8027×10^{3}
m_SCA	1.3007×10^{3}	3.3452×10^{-1}	1.4142×10^{3}	1.1462×10^{1}	2.2627×10^{3}	8.4352×10^{2}
SCADE	1.3038×10^{3}	2.5871×10^{-1}	1.4902×10^{3}	1.1514×10^{1}	2.0450×10^{4}	8.8527×10^{3}
ASCA_PSO	1.3006×10^{3}	1.4205×10^{-1}	1.4035×10^{3}	7.1583×10^{0}	1.5545×10^{3}	1.2124×10^{2}
ACWOA	1.3017×10^{3}	1.0761×10^{0}	1.4166×10^{3}	1.0655×10^{1}	1.9949×10^{3}	5.8404×10^{2}
MFO	1.3019×10^{3}	1.2975×10^{0}	1.4267×10^{3}	1.5955×10^{1}	3.3650×10^{5}	8.2577×10^{5}
SCA	1.3029×10^{3}	3.7934×10^{-1}	1.4443×10^{3}	9.4586×10^{0}	5.0147×10^{3}	3.4034×10^{3}
FA	1.3029×10^{3}	1.9248×10^{-1}	1.4403×10^{3}	4.8273×10^{0}	1.5752×10^{4}	4.4028×10^{3}
DA	1.3068×10^{3}	1.9095×10^{0}	1.5637×10^{3}	8.3347×10^{1}	2.4757×10^{4}	7.1463×10^{4}
	F16		F17		F18	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	1.6122×10^{3}	3.8012×10^{-1}	2.1700×10^{6}	2.7205×10^{6}	1.5189×10^{4}	5.2280×10^{4}
OBSCA	1.6130×10^{3}	1.4281×10^{-1}	1.1486×10^{7}	5.1039×10^{6}	1.9793×10^{8}	1.4800×10^{8}
m_SCA	1.6115×10^{3}	5.1409×10^{-1}	1.5833×10^{6}	1.7905×10^{6}	3.4874×10^{7}	4.7812×10^{7}
SCADE	1.6127×10^{3}	1.9941×10^{-1}	1.4197×10^{7}	6.7951×10^{6}	1.6517×10^{8}	1.1211×10^{8}
ASCA_PSO	1.6126×10^{3}	3.3022×10^{-1}	1.2265×10^{6}	1.0213×10^{6}	3.6646×10^{6}	1.0393×10^{6}
ACWOA	1.6123×10^{3}	4.6588×10^{-1}	1.6366×10^{7}	1.4017×10^{7}	4.6377×10^{7}	3.8096×10^{7}
MFO	1.6128×10^{3}	4.8526×10^{-1}	4.0035×10^{6}	5.0310×10^{6}	3.9147×10^{7}	1.0322×10^{8}
SCA	1.6127×10^{3}	2.8567×10^{-1}	6.9907×10^{6}	3.6926×10^{6}	1.6756×10^{8}	8.8211×10^{7}
FA	1.6129×10^{3}	2.3262×10^{-1}	6.7491×10^{6}	2.2624×10^{6}	2.6476×10^{8}	7.8340×10^{7}
DA	1.6129×10^{3}	2.4315×10^{-1}	8.5018×10^{7}	4.2101×10^{7}	4.0928×10^{9}	1.8915×10^{9}

Table 3. *Cont.*

	F19		F20		F21	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	1.9217×10^3	8.2441×10^0	2.2795×10^3	6.9312×10^1	1.9235×10^5	2.8749×10^5
OBSCA	2.0091×10^3	1.1149×10^1	3.0362×10^4	1.2377×10^4	2.3649×10^6	1.5032×10^6
m_SCA	1.9453×10^3	2.5699×10^1	1.0286×10^4	4.6386×10^3	4.6439×10^5	4.6037×10^5
SCADE	2.0209×10^3	1.7879×10^1	2.7828×10^4	1.2075×10^4	2.7903×10^6	1.0593×10^6
ASCA_PSO	1.9258×10^3	2.5713×10^1	6.0026×10^3	2.2111×10^3	3.2508×10^5	2.5701×10^5
ACWOA	2.0062×10^3	3.5162×10^1	4.0828×10^4	1.8916×10^4	5.1240×10^6	4.8145×10^6
MFO	1.9722×10^3	6.5003×10^1	6.7453×10^4	3.5593×10^4	7.3786×10^5	1.1693×10^6
SCA	1.9950×10^3	2.2940×10^1	1.7570×10^4	5.4464×10^3	1.3486×10^6	6.6249×10^5
FA	2.0029×10^3	1.1339×10^1	2.1545×10^4	8.6661×10^3	1.8937×10^6	6.2858×10^5
DA	2.2044×10^3	1.3085×10^2	7.3418×10^4	4.0133×10^4	2.4506×10^7	2.0130×10^7
	F22		F23		F24	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	2.6667×10^3	1.3749×10^2	2.5001×10^3	8.2205×10^{-2}	2.6001×10^3	4.2571×10^{-2}
OBSCA	3.0956×10^3	1.6521×10^2	2.6865×10^3	1.6694×10^1	2.6000×10^3	2.6232×10^{-4}
m_SCA	2.6529×10^3	1.6213×10^2	2.6396×10^3	1.0453×10^1	2.6000×10^3	6.3375×10^{-4}
SCADE	3.1130×10^3	1.5936×10^2	2.5000×10^3	0.0000×10^0	2.6000×10^3	1.0671×10^{-7}
ASCA_PSO	2.7768×10^3	1.7913×10^2	2.6237×10^3	3.9400×10^0	2.6366×10^3	8.2081×10^0
ACWOA	3.0574×10^3	2.1215×10^2	2.5367×10^3	7.4780×10^1	2.6000×10^3	8.5021×10^{-6}
MFO	2.9977×10^3	2.5111×10^2	2.6671×10^3	4.5312×10^1	2.6827×10^3	3.0780×10^1
SCA	2.9493×10^3	1.4065×10^2	2.6668×10^3	1.2152×10^1	2.6001×10^3	5.8342×10^{-2}
FA	2.9399×10^3	1.0040×10^2	2.7354×10^3	1.4354×10^1	2.7065×10^3	4.4005×10^0
DA	1.3035×10^4	1.1958×10^4	2.8764×10^3	2.2534×10^2	2.6261×10^3	5.0498×10^0
	F25		F26		F27	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	2.7000×10^3	1.3977×10^{-3}	2.7006×10^3	1.8063×10^{-1}	2.9000×10^3	1.8895×10^{-3}
OBSCA	2.7000×10^3	1.0817×10^{-3}	2.7039×10^3	4.7598×10^{-1}	3.2360×10^3	4.5158×10^1
m_SCA	2.7134×10^3	2.6641×10^0	2.7008×10^3	3.4050×10^{-1}	3.1926×10^3	1.5161×10^2
SCADE	2.7000×10^3	0.0000×10^0	2.7037×10^3	6.1565×10^{-1}	3.1829×10^3	2.6437×10^2
ASCA_PSO	2.7125×10^3	5.1192×10^0	2.7006×10^3	1.2849×10^{-1}	3.5114×10^3	2.3638×10^2
ACWOA	2.7000×10^3	0.0000×10^0	2.7471×10^3	5.0332×10^1	3.6882×10^3	3.2535×10^2
MFO	2.7190×10^3	1.0042×10^1	2.7023×10^3	1.5257×10^0	3.6672×10^3	1.8397×10^2
SCA	2.7242×10^3	1.1442×10^1	2.7023×10^3	5.9638×10^{-1}	3.4473×10^3	3.1999×10^2
FA	2.7342×10^3	4.0567×10^0	2.7023×10^3	2.8881×10^{-1}	3.8003×10^3	2.8675×10^1
DA	2.7109×10^3	4.7079×10^0	2.7740×10^3	4.0242×10^1	4.2646×10^3	2.7086×10^2

Table 3. *Cont.*

	F28		F29		F30	
	Ave	Std	Ave	Std	Ave	Std
GGBDA	3.0000×10^3	2.8946×10^{-2}	3.1087×10^3	5.4266×10^0	3.5861×10^3	6.0874×10^2
OBSCA	5.3347×10^3	3.2181×10^2	1.8861×10^7	1.0186×10^7	4.5744×10^5	1.5327×10^5
m_SCA	3.9404×10^3	2.3055×10^2	1.6245×10^6	4.3077×10^6	4.6418×10^4	2.2798×10^4
SCADE	5.2213×10^3	5.2511×10^2	1.5436×10^7	7.9392×10^6	4.1012×10^5	1.8490×10^5
ASCA_PSO	4.4056×10^3	3.2811×10^2	5.1473×10^6	6.2012×10^6	4.1476×10^4	3.1316×10^4
ACWOA	4.2050×10^3	1.1911×10^3	2.1367×10^7	1.7150×10^7	3.9650×10^5	2.1202×10^5
MFO	3.9192×10^3	1.3812×10^2	2.6412×10^6	3.4748×10^6	5.7740×10^4	4.9279×10^4
SCA	4.7438×10^3	2.5806×10^2	1.1250×10^7	6.3057×10^6	2.4763×10^5	7.9063×10^4
FA	4.2782×10^3	1.8313×10^2	3.2845×10^6	1.2612×10^6	1.7562×10^5	4.0487×10^4
DA	8.5874×10^3	1.1741×10^3	2.7820×10^8	2.7534×10^8	4.1499×10^6	2.4981×10^6
	Overall Rank					
	Rank	+/=/−				
GGBDA	1	~				
OBSCA	9	28/0/2				
m_SCA	2	23/0/7				
SCADE	8	27/0/3				
ASCA_PSO	3	24/0/6				
ACWOA	5	26/0/4				
MFO	4	28/0/2				
SCA	6	28/0/2				
FA	7	30/0/0				
DA	10	30/0/0				

F8–F13 represents the multi-modal functions that have numerous local optimal solutions. They are very suitable for evaluating the local optimal prevention of the search ability of the algorithm. For F10 and F11, the results of GGBDA are near to the global optimal solution. However, the other comparison algorithm is easy to fall into the non-global optimal solution to different degrees. For the rest functions, GGBDA still obtains results that are better than most other algorithms. In conclusion, the experimental result verifies the global exploration ability of GGBDA.

From the convergence in Figure 2, we can estimate and evaluate the convergence performance of the algorithm. In F3, F10, F18, F20, F27, F28, F29, and F30, the convergence of GGBDA is better than other comparison algorithms in the early iterations. From the convergence of F6 and F11, GGBDA does not obtain the best adaptive in the early iteration but in the later iteration. In summary, the symbol "+/=/−" shows that GGBDA ranks first with the avg far lower than the second SCA, and the performance is even better than OBSCA, m_SCA, SCADE, ASCA_PSO, ACWOA, MFO, SCA, FA, and DA.

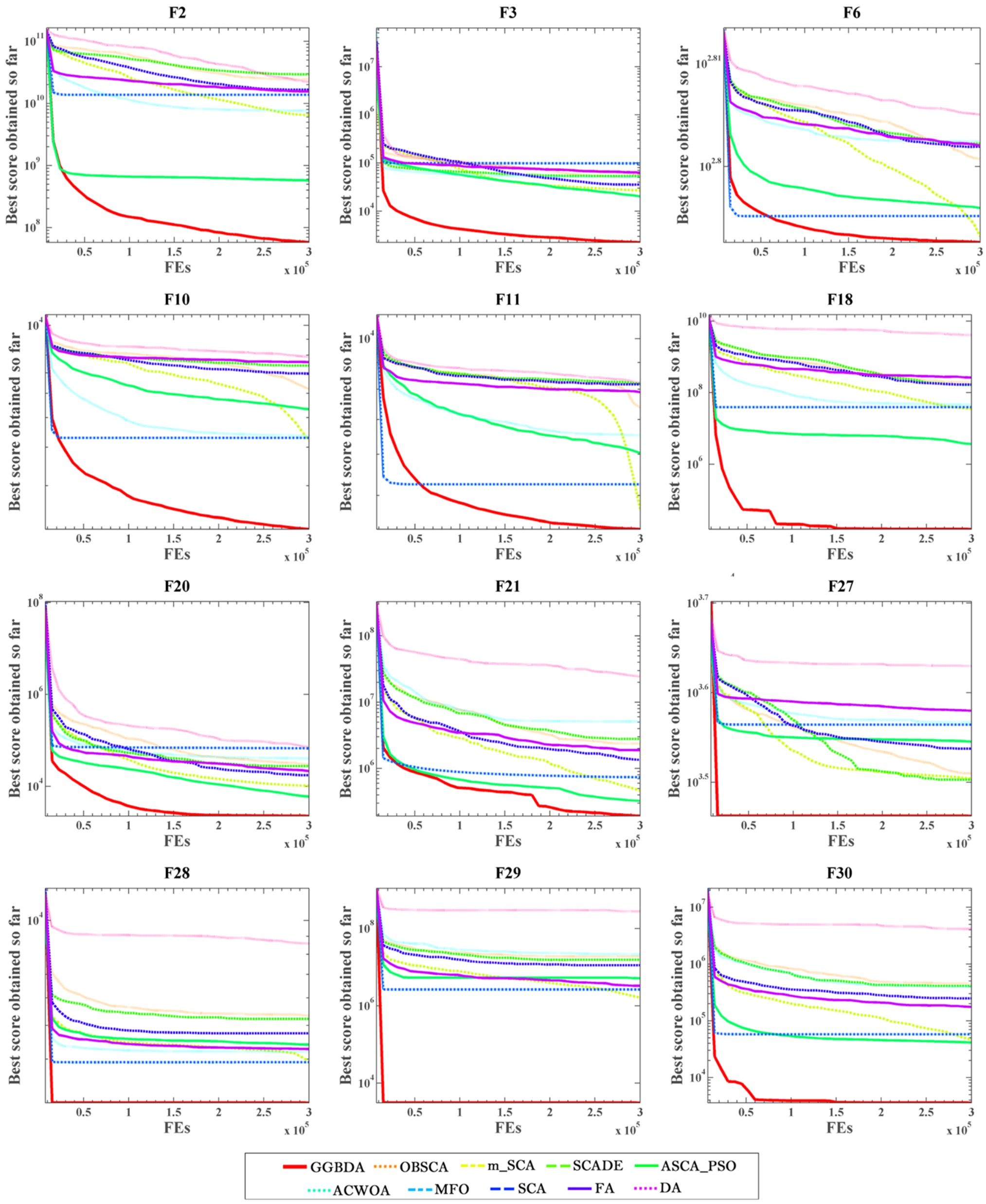

Figure 2. Convergence graph of the 12 benchmarks.

4.2.2. Balance Analysis

In this section, we conduct a qualitative analysis of GGBDA on the 30 functions of CEC14. The original DA was selected for comparison with GGBDA. Figure 3 shows the results of the feasibility analysis of GGBDA and the DA. There are five columns in the figure. The first column (a) is the location distribution of the GGBDA search history on the three-dimensional plane. The second column (b) is the location distribution of the GGBDA search history on the two-dimensional plane. The third column (c) is the trajectory of the first dimension of GGBDA during the iteration. The fourth column (d) shows the change of the average fitness of GGBDA during the iteration. The fifth column (e) shows the convergence curves of GGBDA and DA. In Figure 3b, the red dot represents the location of the optimal solution, and the black dot represents the search location of GGBDA. In the selected 5 function images, the black dots are denser in the area around the red dots, which shows that GGBDA has developed the area in which the optimal solution is located. In Figure 3c, we can see that the first-dimensional trajectory of GGBDA fluctuates greatly in the early period. Early volatility indicates that the algorithm has conducted extensive searches. The average fitness change of GGBDA in the whole iterative process is shown in Figure 3d. We can see that the average fitness of GGBDA dropped to a lower level in the mid-term. This shows that GGBDA has a good convergence speed. In Figure 3e, we can clearly see that the convergence curve of GGBDA is lower than that of DA, which shows that GGBDA can obtain a better solution.

The balance analysis and diversity analysis are carried out on the same functions. Figure 4 shows the results of the balanced analysis of GGBDA and DA. In Figure 4, there are three curves in each graph. As shown in the Figure, the blue curve and red curve represent exploitation and exploration, respectively. The larger value of the curve means that the corresponding behavior is dominant in the algorithm. The green curve indicates incremental–decremental. The curve can more intuitively reflect the changing trends of the two behaviors of the algorithm. When the value of the curve increases, it means that the exploration activity is dominant. Instead, exploitative behavior predominates. When the curve drops to a negative value, the curve will be set to zero. Comparing the curves of the two algorithms shows that both algorithms were dominated by exploration behavior in the early stage. This is because the swarm intelligence optimization algorithm performs a global search first, at the beginning. However, the difference between the two algorithm curves is also very obvious. The DA spends more time on exploration behavior than GGBDA. The exploration behavior of DA almost accounts for half of the entire iteration process. However, the exploitative behavior of GGBDA quickly became dominant, indicating that it spent more time exploiting the target area. This is the impact of the two mechanisms added to GGBDA on its balance.

Figure 5 is the result of the diversity analysis of GGBDA and DA. In Figure 5, the ordinate represents the population diversity. We can see that the diversity of the two algorithms is very high at the beginning. This is because the initial population of the algorithm is randomly generated. Then, in the iterative process, the algorithm continues to narrow the search range so that the diversity of the population will reduce, although the diversity curves of the two algorithms almost reached the lowest in the iteration. However, the descent process of the two algorithms is very different. We can clearly observe that the DA maintained a high diversity in the early stage. The diversity curve of the DA dropped to its lowest value very quickly in the mid-term. This change was completed in a concise time.

In contrast, the curve of GGBDA declined more gently. GGBDA only declines rapidly at the initial stage, and then the rate of decline slows down. This is obvious for F2 and F14. This shows that the two added mechanisms have an impact on the diversity of the DA. Owing to the strong search capability, the proposed GGBDA can also be applied to other optimization problems, such as fault detection [81]; metabolomic data processing [82,83]; urban road planning [84]; multivariate time series analysis [85]; gene signature identification [86]; drug target discovery [87]; drug discovery [88]; pharmacoinformatics data mining [89]; service ecosystem [90,91]; information retrieval services [92–94]; kayak cycle phase segmentation [95]; covert communication system [96–98]; location-based services [99,100]; and human motion capture [101].

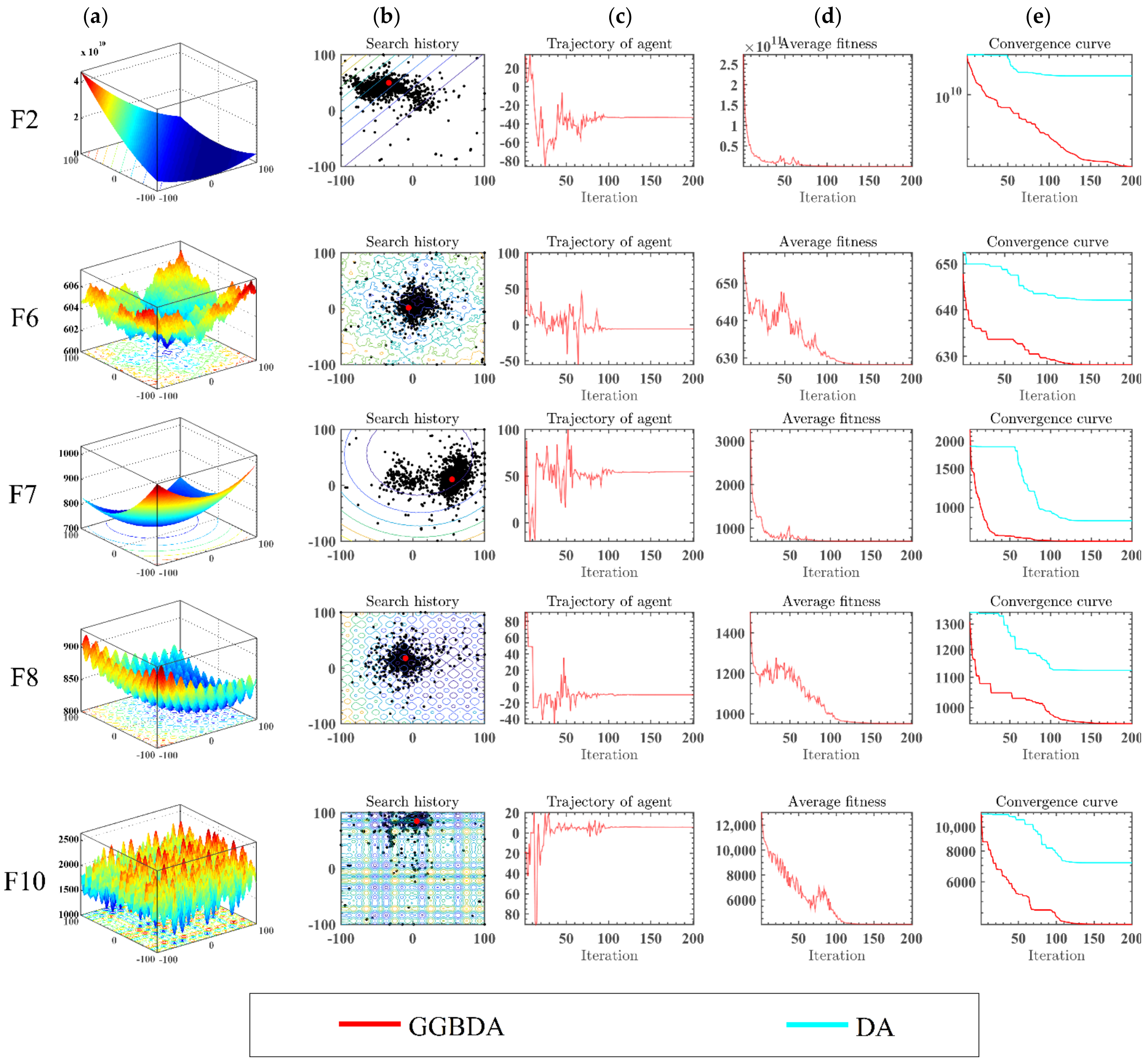

Figure 3. (**a**) Three-dimensional location distribution of GGBDA, (**b**) two-dimensional location distribution of GGBDA, (**c**) trajectory of GGBDA in the first dimension, (**d**) average fitness of GGBDA, and (**e**) convergence curves of GGBDA and DA.

4.3. Real-World Problems

4.3.1. Pressure Vessel Design (PVD) Problem

The PVD problem is a common engineering design problem. There are four constraints and four parameters in the PVD problem. The main aim is to obtain a pressure vessel that meets the conditions with relatively minimal costs.

The formula of this problem is listed below.

Consider:

$$X = [x_1\ x_2\ x_3\ x_4] = [T_s\ T_h\ R\ L]$$

Range of parameters:

$$\begin{aligned} 0 &\leq x_1 \leq 99 \\ 0 &\leq x_2 \leq 99 \\ 10 &\leq x_3 \leq 200 \\ 10 &\leq x_4 \leq 200 \end{aligned}$$

Minimize:

$$f\left(\vec{x}\right) = 0.6224x_1x_3x_4 + 1.7781x_3x_1^2 + 3.1661x_4x_1^2 + 19.84x_3x_1^2$$

Subject to:

$$\begin{aligned} g_1(X) &= -x_1 + 0.0193x_3 \leq 0 \\ g_2(X) &= -x_3 + 0.00954x_3 \leq 0 \\ g_3(X) &= -\pi x_4x_3^2 - \tfrac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\ g_4(X) &= x_4 - 240 \leq 0 \end{aligned}$$

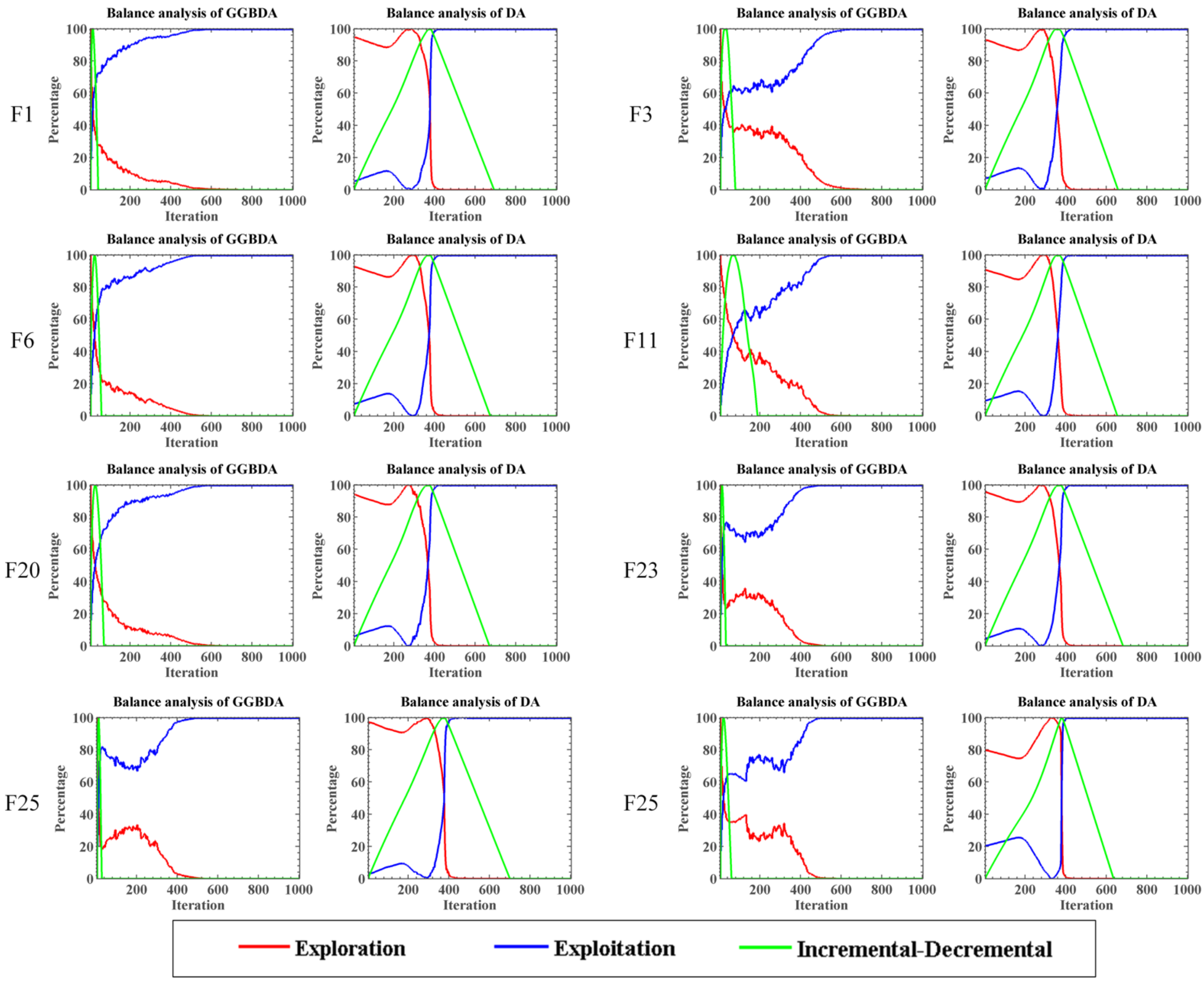

Figure 4. Balance analysis of GGBDA and DA.

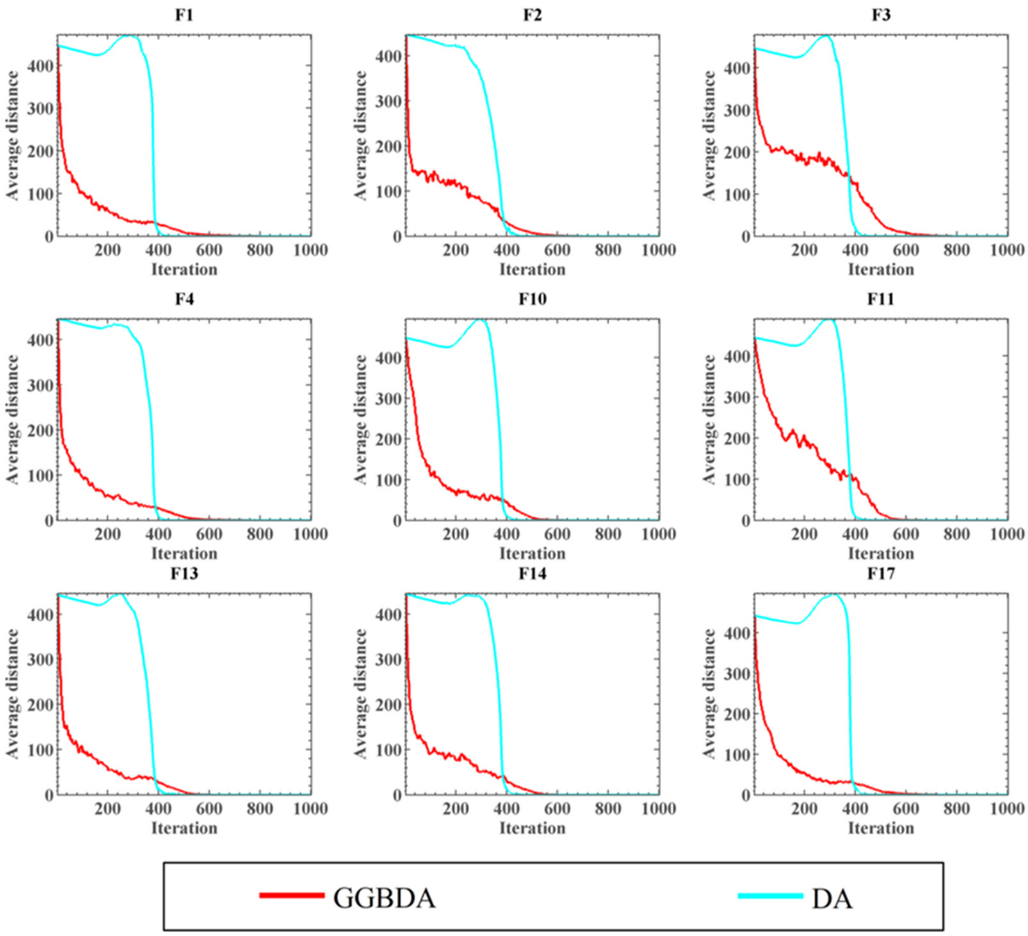

Figure 5. Diversity analysis of GGBDA and DA.

Table 4 shows the results GGBDA for the optimization for the PVD problem, compared with other peers in the literature. The results show that the optimal value obtained by the GGBDA was 6059.7298, which was better than CPSO, WOA, and Branch-bound. Moreover, GGBDA has a similar effect with MFO, HPSO, and BA.

Table 4. Comparison results of the PVD problem between GGBDA and other approaches.

Algorithm	Optimum Variables				Optimum Cost
	T_s	T_h	R	L	
GGBDA	0.8125	0.4375	42.0983	176.6380	6059.7298
MFO [15]	0.8125	0.4375	42.0984	176.6366	6059.7143
BA [102]	0.8125	0.4375	42.0984	176.6366	6059.7143
HPSO [103]	0.8125	0.4375	42.0984	176.6366	6059.7143
CSS [104]	0.8125	0.4375	42.1036	176.5727	6059.0888
CPSO [105]	0.8125	0.4375	42.0912	176.7465	6061.0777
ACO [106]	0.8125	0.4375	42.1036	176.5727	6059.0888
GWO [18]	0.8125	0.4345	42.0892	176.7587	6051.5639
WOA [2]	0.8125	0.4375	42.0983	176.6390	6059.7410
MDDE [107]	0.8125	0.4375	42.0984	176.6360	6059.7017
Branch-bound [108]	1.1250	0.6250	47.7000	117.7010	8129.1036

4.3.2. Hydrostatic Thrust Bearings Design (HTBD) Problem

The goal of the HTBD problem is to minimize power loss. At the same time, the design needs to meet some constraints. There are four design variables: bearing step radius (R), recess radius (R_0), oil viscosity (μ), and flow rate (Q). The mathematical model of this problem is shown as below.

Minimize:

$$\mathrm{f(x)} = \frac{\mathrm{QP_0}}{0.7} + \mathrm{E}_f$$

Subject to:

$$\mathrm{g_1(x)} = \frac{\pi \mathrm{P_0}}{2} \times \frac{\mathrm{R^2 - R_0^2}}{\ln(\mathrm{R/R_0})} - \mathrm{W_s} \geq 0$$

$$\mathrm{g}_2(x) = P_{max} - P_0 \geq 0$$

$$\mathrm{g}_3(x) = \Delta T_{max} - \Delta T \geq 0$$

$$\mathrm{g}_4 = h - h_{min} \geq 0$$

$$\mathrm{g}_5(x) = R - R_0 \geq 0$$

$$\mathrm{g}_6(x) = 0.001 - \gamma / g P_0 (Q/2\pi R h) \geq 0$$

$$\mathrm{g}_7(x) = 5000 - \frac{W}{\pi\left(R^2 - R_0^2\right)} \geq 0$$

where

$$\mathrm{P_0} = \frac{6\mu Q}{\pi h^3} \ln\left(\frac{R}{R_0}\right)$$

$$\mathrm{E}_f = 9336 Q \gamma C \Delta T$$

$$\Delta T = 2\left(10^{\mathrm{P}} - 560\right)$$

$$P = \frac{\log\left(\log\left(8.122 \times 10^6 + 0.8\right)\right) - C_1}{n}$$

$$h = \left(\frac{2\pi N}{60}\right)^2 \frac{2\pi\mu}{E_f} \left(\frac{R^4}{4} - \frac{R_0^4}{4}\right)$$

$$\mathrm{C_1} = 10.04$$

$$\mathrm{n} = -3.55,\ \mathrm{Pmax} = 1000,\ \mathrm{Ws} = 101000$$

$$\Delta \mathrm{T}_{max} = 50$$

$$h_{min} = 0.001$$

$$\mathrm{g} = 386.4,\ \mathrm{N} = 750$$

$$5 \leq D_e, D_i \leq 15$$

$$0.01 \leq t \leq 6$$

$$0.05 \leq h \leq 0.5$$

Table 5 shows the results of the HTBD problem. It can be seen that the optimal value of GGBDA is 19,508.76, which is better than PSO, SQP, and GASO. Moreover, GGBDA has almost the same effect as TNE and TLBO.

Table 5. Comparison results of the hydrostatic thrust bearing problem between GGBDA and other approaches.

Algorithm	Optimum Variables				Optimum Cost
	R	R0	μ	Q	
GGBDA	5.956071	5.389334	5.36×10^{-6}	2.271766	19,508.7584
PSO [8]	5.956868	5.389175	5.4021×10^{-6}	2.301546	19,586.5788
NDE [109]	5.955781	5.389013	5.3586×10^{-6}	2.269656	19,506.0090
TLBO [110]	5.955781	5.389013	5.3586×10^{-6}	2.269656	19,505.3132
SQP [111]	5.955800	5.389040	8.6332×10^{-6}	8.000010	26,114.5450
GASO [112]	6.271000	12.90100	5.6050×10^{-6}	2.938000	23,403.4320

4.3.3. Welded Beam Design (WBD) Problem

WBD problem aims to minimize the cost of welded beams subject to the four constraints of shear stress (τ), bending stress (θ), buckling load (P_c), and deflection (δ). The variables in this problem are composed of welding seam thickness (h), welding joint length (l), beam width (t), and beam thickness (b). The mathematical model of this problem is listed as below.

Consider:

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

Minimize:

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

Subject to:

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

Variable range:

$$0.1 \leq x_1 \leq 2$$
$$0.1 \leq x_2 \leq 10$$
$$0.1 \leq x_3 \leq 10$$
$$0.1 \leq x_4 \leq 2$$

where

$$\tau\left(\vec{x}\right) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}$$

$$\tau'' = \frac{MR}{J}$$

$$\tau'' = \frac{MR}{J}$$

$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$

$$\sigma\left(\vec{x}\right) = \frac{6PL}{x_4{x_3}^2}$$

$$\delta\left(\vec{x}\right) = \frac{6PL^3}{Ex_3^2x_4}$$

$$P_C\left(\vec{x}\right) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$$P = 60001b$$

$$L = 14 \in \delta_{max} = 0.25 \in$$

$$E = 30 \times 1^6 psi$$

$$G = 12 \times 10^6 psi \quad \tau_{max} = 13600psi$$

$$\sigma_{max} = 30000psi$$

The results of the WBD problem are shown in Table 6. The optimal value of GGBDA is 1.724527, which is the lowest among all the algorithms. It can be seen that GGBDA has a better effect than other peers in the experiment.

Table 6. Comparison results of the WBD problem between GGBDA and other approaches.

Algorithm	Optimal Values for Variables				Optimum Cost
	h	***l***	***t***	***b***	
GGBDA	0.187156	3.615020	9.056672	0.206464	1.724527
RO [113]	0.203687	3.528467	9.004233	0.207241	1.735344
SSA [114]	0.205700	3.471400	9.036600	0.205700	1.724910
CDE [115]	0.203137	3.542998	9.033498	0.206179	1.733462
GWO [18]	0.205700	3.478400	9.036800	0.205800	1.726240
GSA [116]	0.182129	3.856979	10.00000	0.202376	1.879950
NDE [109]	0.205729	3.470488	9.903662	0.205729	1.724852

4.3.4. Tension–Compression String Design (TCSD) Problem

The TCSD problem is to design a tension–compression spring with the minimum weight and meets the constraints. The three variables in the problem are the wire diameter (d), mean coil diameter (D), and the number of active coils (N). The mathematical model of this problem is listed as below.

Consider:

$$\vec{x} = [x_1\ x_2\ x_3] = [d\ D\ N]$$

Objective function:

$$Minimize\ f(x) = x_1^2 x_2 x_3 + 2x_1^2 x_2$$

Subject to:

$$h_1\left(\vec{x}\right) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$$

$$h_2\left(\vec{x}\right) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5180 x_1^2} - 1 \leq 0$$

$$h_3\left(\vec{x}\right) = 1 - \frac{140.45 x_1}{x_2^3 x_3} \leq 0$$

$$h_4\left(\vec{x}\right) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

Variable ranges:

$$0.05 \leq x_1 \leq 2.00$$

$$0.25 \leq x_2 \leq 1.30,$$

$$2.00 \leq x_3 \leq 15.0$$

Table 7 shows the results of the TCSD problem. The optimal values of GGBDA and NDE are both 0.012665, which is the lowest among the algorithms. It can be seen that GGBDA still has a good effect on the TCSD problem.

Table 7. Comparison results of the TCSD problem between GGBDA and other approaches.

Algorithm	Optimal Values for Variables			Optimum Cost
	d	***D***	***N***	
GGBDA	0.051652	0.355837	11.34081	0.012665
GA [117]	0.051480	0.351661	11.63220	0.012705
RO [113]	0.051370	0.349096	11.76279	0.012679
IHS [118]	0.051154	0.349871	12.07643	0.012671
ES [119]	0.051989	0.363965	10.89052	0.012681
GSA [116]	0.050276	0.323680	13.52541	0.012702
WOA [2]	0.051207	12.00430	0.345215	0.012676
PSO [8]	0.015728	11.24454	0.357644	0.012675
NDE [109]	0.051689	0.356718	11.28896	0.012665

5. Conclusions

The purpose of this research was to propose an enhanced DA that anticipates engineering design problems more efficiently and precisely. The Gaussian mutation and the Gaussian barebone are embedded into the DA, termed as GGBDA. The Gaussian mutation was used to prevent slipping into local optimal situations and to update the individual locations in a random manner. To further enhance local exploitation capacities, Gaussian barebone was used in conjunction with the improvement of Gaussian mutation, the global searching ability, and the convergence efficiency of GGBDA to accelerate the convergent speed and strengthen local exploitation capacities. This study compared the performance of GGBDA with other competitive peers on 30 benchmarks and 4 engineering design issues. The experimental findings demonstrate that GGBDA outperforms DA and other competing algorithms in terms of solution accuracy and convergence speed.

GGBDA's performance and time cost will be improved in future developments. For example, we will address GGBDA's design issues. GGBDA may also be used to anticipate and optimize the parameters for energy optimization, image segmentation, and parameter optimization of machine learning methods.

Author Contributions: Conceptualization, L.Y., F.K. and H.C.; Methodology, H.C. and S.Z.; software, S.Z.; validation, H.C., L.Y., F.K. and S.Z.; formal analysis, L.Y. and F.K.; investigation, S.Z.; resources, H.C.; data curation, S.Z.; writing—original draft preparation, S.Z.; writing—review and editing, H.C., S.Z., L.Y. and F.K.; visualization, S.Z., L.Y. and F.K.; supervision, L.Y. and F.K.; project administration, S.Z.; funding acquisition, H.C., L.Y., F.K. and S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Wenzhou Science and Technology Bureau (ZG2020030) and the Humanities and Social Science Research Planning Fund Project of the Ministry of Education (20YJA790090).

Data Availability Statement: The data involved in this study are all public data, which can be downloaded through public channels.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhan, Z.H.; Shi, L.; Tan, K.C.; Zhang, J. A survey on evolutionary computation for complex continuous optimization. *Artif. Intell. Rev.* **2021**, *55*, 59–110. [CrossRef]
2. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
3. Luo, J.; Chen, H.; Heidari, A.A.; Xu, Y.; Zhang, Q.; Li, C. Multi-strategy boosted mutative whale-inspired optimization approaches. *Appl. Math. Model.* **2019**, *73*, 109–123. [CrossRef]
4. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
5. Booker, L.B.; Goldberg, D.E.; Holland, J.H. Classifier systems and genetic algorithms. *Artif. Intell.* **1989**, *40*, 235–282. [CrossRef]
6. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **1996**, *26*, 29–41. [CrossRef]
7. Deng, W.; Xu, J.; Zhao, H. An Improved Ant Colony Optimization Algorithm Based on Hybrid Strategies for Scheduling Problem. *IEEE Access* **2019**, *7*, 20281–20292. [CrossRef]
8. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks—Conference Proceedings, Perth, WA, Australia, 27 November–1 December 1995; IEEE: Piscataway, NJ, USA, 1995.
9. Zhang, X.; Hu, W.; Xie, N.; Bao, H.; Maybank, S. A Robust Tracking System for Low Frame Rate Video. *Int. J. Comput. Vis.* **2015**, *115*, 279–304. [CrossRef]
10. Yang, X.-S. Firefly Algorithms for Multimodal Optimization. In *Stochastic Algorithms: Foundations and Applications*; SAGA 2009. Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 5792.
11. Pan, W.T. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. *Knowl.-Based Syst.* **2012**, *26*, 69–74. [CrossRef]
12. Shen, L.; Chen, H.; Yu, Z.; Kang, W.; Zhang, B.; Li, H.; Yang, B.; Liu, D. Evolving support vector machines using fruit fly optimization for medical data classification. *Knowl.-Based Syst.* **2016**, *96*, 61–75. [CrossRef]
13. Zhang, Y.; Liu, R.; Asghar Heidari, A.; Wang, X.; Chen, Y.; Wang, M.; Chen, H. Towards Augmented Kernel Extreme Learning Models for Bankruptcy Prediction: Algorithmic Behavior and Comprehensive Analysis. *Neurocomputing* **2020**, *430*, 185–212. [CrossRef]
14. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [CrossRef]
15. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [CrossRef]
16. Xu, Y.; Chen, H.; Luo, J.; Zhang, Q.; Jiao, S.; Zhang, X. Enhanced Moth-flame optimizer with mutation strategy for global optimization. *Inf. Sci.* **2019**, *492*, 181–203. [CrossRef]
17. Xu, Y.; Chen, H.; Heidari, A.A.; Luo, J.; Zhang, Q.; Zhao, X.; Li, C. An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks. *Expert Syst. Appl.* **2019**, *129*, 135–155. [CrossRef]
18. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
19. Zhao, X.; Zhang, X.; Cai, Z.; Tian, X.; Wang, X.; Huang, Y.; Chen, H.; Hu, L. Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients. *Comput. Biol. Chem.* **2019**, *78*, 481–490. [CrossRef]
20. Yang, X.S. A new metaheuristic Bat-inspired Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; (Studies in Computational Intelligence); Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
21. Yu, H.; Zhao, N.; Wang, P.; Chen, H.; Li, C. Chaos-enhanced synchronized bat optimizer. *Appl. Math. Model.* **2020**, *77*, 1201–1215. [CrossRef]
22. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper Optimisation Algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]

23. Luo, J.; Chen, H.; Zhang, Q.; Xu, Y.; Huang, H.; Zhao, X. An improved grasshopper optimization algorithm with application to financial stress prediction. *Appl. Math. Model.* **2018**, *64*, 654–668. [CrossRef]
24. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
25. Tu, J.; Chen, H.; Wang, M.; Gandomi, A.H. The Colony Predation Algorithm. *J. Bionic Eng.* **2021**, *18*, 674–710. [CrossRef]
26. Yang, Y.; Chen, H.; Heidari, A.A.; Gandomi, A.H. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst. Appl.* **2021**, *177*, 114864. [CrossRef]
27. Ahmadianfar, I.; Asghar Heidari, A.; Gandomi, A.H.; Chu, X.; Chen, H. RUN Beyond the Metaphor: An Efficient Optimization Algorithm Based on Runge Kutta Method. *Expert Syst. Appl.* **2021**, *181*, 115079. [CrossRef]
28. Ahmadianfar, I.; Asghar Heidari, A.; Noshadian, S.; Chen, H.; Gandomi, A.H. INFO: An Efficient Optimization Algorithm based on Weighted Mean of Vectors. *Expert Syst. Appl.* **2022**, *194*, 116516. [CrossRef]
29. Wu, S.-H.; Zhan, Z.-H.; Zhang, J. SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems. *IEEE Trans. Evol. Comput.* **2021**, *25*, 478–491. [CrossRef]
30. Li, J.-Y.; Zhan, Z.-H.; Wang, C.; Jin, H.; Zhang, J. Boosting data-driven evolutionary algorithm with localized data generation. *IEEE Trans. Evol. Comput.* **2020**, *24*, 923–937. [CrossRef]
31. Ying, C.; Ying, C.; Ban, C. A performance optimization strategy based on degree of parallelism and allocation fitness. *EURASIP J. Wirel. Commun. Netw.* **2018**, *2018*, 1–8. [CrossRef]
32. Hu, K.; Ye, J.; Fan, E.; Shen, S.; Huang, L.; Pi, J. A novel object tracking algorithm by fusing color and depth information based on single valued neutrosophic cross-entropy. *J. Intell. Fuzzy Syst.* **2017**, *32*, 1775–1786. [CrossRef]
33. Hu, K.; He, W.; Ye, J.; Zhao, L.; Peng, H.; Pi, J. Online Visual Tracking of Weighted Multiple Instance Learning via Neutrosophic Similarity-Based Objectness Estimation. *Symmetry* **2019**, *11*, 832. [CrossRef]
34. Zhang, W.; Hou, W.; Li, C.; Yang, W.; Gen, M. Multidirection Update-Based Multiobjective Particle Swarm Optimization for Mixed No-Idle Flow-Shop Scheduling Problem. *Complex Syst. Modeling Simul.* **2021**, *1*, 176–197. [CrossRef]
35. Liu, X.-F.; Zhan, Z.-H.; Gao, Y.; Zhang, J.; Kwong, S.; Zhang, J. Coevolutionary particle swarm optimization with bottleneck objective learning strategy for many-objective optimization. *IEEE Trans. Evol. Comput.* **2018**, *23*, 587–602. [CrossRef]
36. Deng, W.; Zhang, X.; Zhou, Y.; Liu, Y.; Deng, W.; Chen, H.; Zhao, H. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* **2021**, *585*, 441–453. [CrossRef]
37. Lai, X.; Zhou, Y. Analysis of multiobjective evolutionary algorithms on the biobjective traveling salesman problem (1, 2). *Multimed. Tools Appl.* **2020**, *79*, 30839–30860. [CrossRef]
38. Yang, Z.; Li, K.; Guo, Y.; Ma, H.; Zheng, M. Compact real-valued teaching-learning based optimization with the applications to neural network training. *Knowl. Based Syst.* **2018**, *159*, 51–62. [CrossRef]
39. Han, X.; Han, Y.; Chen, Q.; Li, J.; Sang, H.; Liu, Y.; Pan, Q.; Nojima, Y. Distributed Flow Shop Scheduling with Sequence-Dependent Setup Times Using an Improved Iterated Greedy Algorithm. *Complex Syst. Modeling Simul.* **2021**, *1*, 198–217. [CrossRef]
40. Yi, J.-H.; Deb, S.; Dong, J.; Alavi, A.H.; Wang, G.-G. An improved NSGA-III algorithm with adaptive mutation operator for Big Data optimization problems. *Future Gener. Comput. Syst.* **2018**, *88*, 571–585. [CrossRef]
41. Deng, W.; Liu, H.; Xu, J.; Zhao, H.; Song, Y.J. An improved quantum-inspired differential evolution algorithm for deep belief network. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 7319–7327. [CrossRef]
42. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G. Evolving deep convolutional neural networks for image classification. *IEEE Trans. Evol. Comput.* **2019**, *24*, 394–407. [CrossRef]
43. Deng, W.; Xu, J.; Zhao, H.; Song, Y. A Novel Gate Resource Allocation Method Using Improved PSO-Based QEA. *IEEE Trans. Intell. Transp. Syst.* **2020**. [CrossRef]
44. Deng, W.; Xu, J.; Song, Y.; Zhao, H. An Effective Improved Co-evolution Ant Colony Optimization Algorithm with Multi-Strategies and Its Application. *Int. J. Bio-Inspired Comput.* **2020**, *16*, 158–170. [CrossRef]
45. Zhao, F.; Di, S.; Cao, J.; Tang, J. A novel cooperative multi-stage hyper-heuristic for combination optimization problems. *Complex Syst. Modeling Simul.* **2021**, *1*, 91–108. [CrossRef]
46. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [CrossRef]
47. Guha, D.; Roy, P.K.; Banerjee, S. Optimal tuning of 3 degree-of-freedom proportional-integral-derivative controller for hybrid distributed power system using dragonfly algorithm. *Comput. Electr. Eng.* **2018**, *72*, 137–153. [CrossRef]
48. Wu, J.; Zhu, Y.; Wang, Z.; Song, Z.; Liu, X.; Wang, W.; Zhang, Z.; Yu, Y.; Xu, Z.; Zhang, T.; et al. A novel ship classification approach for high resolution SAR images based on the BDA-KELM classification model. *Int. J. Remote Sens.* **2017**, *38*, 6457–6476. [CrossRef]
49. Ashok Kumar, C.; Vimala, R.; Aravind Britto, K.R.; Sathya Devi, S. FDLA: Fractional Dragonfly based Load balancing Algorithm in cluster cloud model. *Clust. Comput.* **2019**, *22*, 1401–1414. [CrossRef]
50. VeeraManickam, M.R.M.; Mohanapriya, M.; Pandey, B.K.; Akhade, S.; Kale, S.A.; Patil, R.; Vigneshwar, M. Map-Reduce framework based cluster architecture for academic student's performance prediction using cumulative dragonfly based neural network. *Clust. Comput.* **2019**, *22*, 1259–1275. [CrossRef]
51. Yu, C.; Cai, Z.; Ye, X.; Wang, M.; Zhao, X.; Liang, G.; Chen, H.; Li, C. Quantum-like mutation-induced dragonfly-inspired optimization approach. *Math. Comput. Simul.* **2020**, *178*, 259–289. [CrossRef]

52. Sree Ranjini, S.R.; Murugan, S. Memory based Hybrid Dragonfly Algorithm for numerical optimization problems. *Expert Syst. Appl.* **2017**, *83*, 63–78.
53. More, N.S.; Ingle, R.B. Energy-aware VM migration using dragonfly-crow optimization and support vector regression model in Cloud. *Int. J. Modeling Simul. Sci. Comput.* **2018**, *9*, 1850050. [CrossRef]
54. Khadanga, R.K.; Padhy, S.; Panda, S.; Kumar, A. Design and Analysis of Tilt Integral Derivative Controller for Frequency Control in an Islanded Microgrid: A Novel Hybrid Dragonfly and Pattern Search Algorithm Approach. *Arab. J. Sci. Eng.* **2018**, *43*, 3103–3114. [CrossRef]
55. Ghanem, W.A.H.M.; Jantan, A. A Cognitively Inspired Hybridization of Artificial Bee Colony and Dragonfly Algorithms for Training Multi-layer Perceptrons. *Cogn. Comput.* **2018**, *10*, 1096–1134. [CrossRef]
56. Shilaja, C.; Arunprasath, T. Internet of medical things-load optimization of power flow based on hybrid enhanced grey wolf optimization and dragonfly algorithm. *Future Gener. Comput. Syst.* **2019**, *98*, 319–330.
57. Aadil, F.; Ahsan, W.; Rehman, Z.U.; Shah, P.A.; Rho, S.; Mehmood, I. Clustering algorithm for internet of vehicles (IoV) based on dragonfly optimizer (CAVDO). *J. Supercomput.* **2018**, *74*, 4542–4567. [CrossRef]
58. Aci, C.I.; Gülcan, H. A modified dragonfly optimization algorithm for single- and multiobjective problems using brownian motion. *Comput. Intell. Neurosci.* **2019**, *2019*, 6871298. [CrossRef] [PubMed]
59. Bao, X.; Jia, H.; Lang, C. Dragonfly algorithm with Opposition-based learning for multilevel thresholding color image segmentation. *Symmetry* **2019**, *11*, 716. [CrossRef]
60. Li, L.L.; Zhao, X.; Tseng, M.L.; Tan, R.R. Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm. *J. Clean. Prod.* **2020**, *242*, 118447. [CrossRef]
61. Sayed, G.I.; Tharwat, A.; Hassanien, A.E. Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection. *Appl. Intell.* **2019**, *49*, 188–205. [CrossRef]
62. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Faris, H.; Fournier-Viger, P.; Li, X.; Mirjalili, S. Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl.-Based Syst.* **2018**, *161*, 185–204. [CrossRef]
63. Hariharan, M.; Sindhu, R.; Vijean, V.; Yazid, H.; Nadarajaw, T.; Yaacob, S.; Polat, K. Improved binary dragonfly optimization algorithm and wavelet packet based non-linear features for infant cry classification. *Comput. Methods Programs Biomed.* **2018**, *155*, 39–51. [CrossRef]
64. Zhang, A.; Zhang, P.; Feng, Y. Short-term load forecasting for microgrids based on DA-SVM. *COMPEL Int. J. Comput. Math. Electr. Electron. Eng.* **2018**, *38*, 68–80. [CrossRef]
65. Yuan, Y.; Lv, L.; Wang, X.; Song, X. Optimization of a frame structure using the Coulomb force search strategy-based dragonfly algorithm. *Eng. Optim.* **2019**, *52*, 915–931. [CrossRef]
66. Zhang, Z.; Hong, W.C. Electric load forecasting by complete ensemble empirical mode decomposition adaptive noise and support vector regression with quantum-based dragonfly algorithm. *Nonlinear Dyn.* **2019**, *98*, 1107–1136. [CrossRef]
67. Suresh, V.; Sreejith, S. Generation dispatch of combined solar thermal systems using dragonfly algorithm. *Computing* **2017**, *99*, 59–80. [CrossRef]
68. Sureshkumar, K.; Ponnusamy, V. Power flow management in micro grid through renewable energy sources using a hybrid modified dragonfly algorithm with bat search algorithm. *Energy* **2019**, *181*, 1166–1178. [CrossRef]
69. Xie, T.; Yao, J.; Zhou, Z. DA-based parameter optimization of combined kernel support vector machine for cancer diagnosis. *Processes* **2019**, *7*, 263. [CrossRef]
70. Xu, L.; Jia, H.; Lang, C.; Peng, X.; Sun, K. A Novel Method for Multilevel Color Image Segmentation Based on Dragonfly Algorithm and Differential Evolution. *IEEE Access* **2019**, *7*, 19502–19538. [CrossRef]
71. Zhang, Q.; Wang, Z.; Heidari, A.A.; Gui, W.; Shao, Q.; Chen, H.; Zaguia, A.; Turabieh, H.; Chen, M. Gaussian Barebone Salp Swarm Algorithm with Stochastic Fractal Search for medical image segmentation: A COVID-19 case study. *Comput. Biol. Med.* **2021**, *139*, 104941. [CrossRef]
72. Xia, J.; Zhang, H.; Li, R.; Wang, Z.; Cai, Z.; Gu, Z.; Chen, H.; Pan, Z. Adaptive Barebones Salp Swarm Algorithm with Quasi-oppositional Learning for Medical Diagnosis Systems: A Comprehensive Analysis. *J. Bionic Eng.* **2022**, *19*, 1–17. [CrossRef]
73. Liang, J.; Qu, B.Y.; Suganthan, P. *Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization*; Technical Report, 201311; Zhengzhou University: Zhengzhou, China; Nanyang Technological University: Singapore, 2013.
74. Abd Elaziz, M.; Oliva, D.; Xiong, S. An improved Opposition-Based Sine Cosine Algorithm for global optimization. *Expert Syst. Appl.* **2017**, *90*, 484–500. [CrossRef]
75. Qu, C.; Zeng, Z.; Dai, J.; Yi, Z.; He, W. A Modified Sine-Cosine Algorithm Based on Neighborhood Search and Greedy Levy Mutation. *Comput. Intell. Neurosci.* **2018**, *2018*, 4231647. [CrossRef] [PubMed]
76. Nenavath, H.; Jatoth, R.K. Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. *Appl. Soft Comput.* **2018**, *62*, 1019–1043. [CrossRef]
77. Issa, M.; Hassanien, A.E.; Oliva, D.; Helmi, A.; Ziedan, I.; Alzohairy, A. ASCA-PSO: Adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment. *Expert Syst. Appl.* **2018**, *99*, 56–70. [CrossRef]
78. Elhosseini, M.A.; Haikal, A.Y.; Badawy, M.; Khashan, N. Biped robot stability based on an A–C parametric Whale Optimization Algorithm. *J. Comput. Sci.* **2019**, *31*, 17–32. [CrossRef]
79. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]

80. Yang, X.-S. *Firefly Algorithms for Multimodal Optimization*; Springer: Berlin/Heidelberg, Germany, 2009.
81. Cui, H.; Guan, Y.; Chen, H.; Deng, W. A Novel Advancing Signal Processing Method Based on Coupled Multi-Stable Stochastic Resonance for Fault Detection. *Appl. Sci.* **2021**, *11*, 5385. [CrossRef]
82. Fu, J.; Zhang, Y.; Wang, Y.; Zhang, H.; Liu, J.; Tang, J.; Yang, Q.; Sun, H.; Qiu, W.; Ma, Y. Optimization of metabolomic data processing using NOREVA. *Nat. Protoc.* **2021**, *17*, 129–151. [CrossRef]
83. Li, B.; Tang, J.; Yang, Q.; Li, S.; Cui, X.; Li, Y.; Chen, Y.; Xue, W.; Li, X.; Zhu, F. NOREVA: Normalization and evaluation of MS-based metabolomics data. *Nucleic Acids Res.* **2017**, *45*, W162–W170. [CrossRef]
84. Ran, X.; Zhou, X.; Lei, M.; Tepsan, W.; Deng, W. A novel k-means clustering algorithm with a noise algorithm for capturing urban hotspots. *Appl. Sci.* **2021**, *11*, 11202. [CrossRef]
85. Wang, M.; Zhang, Q.; Chen, H.; Heidari, A.A.; Mafarja, M.; Turabieh, H. Evaluation of constraint in photovoltaic cells using ensemble multi-strategy shuffled frog leading algorithms. *Energy Convers. Manag.* **2021**, *244*, 114484. [CrossRef]
86. Yang, Q.; Li, B.; Tang, J.; Cui, X.; Wang, Y.; Li, X.; Hu, J.; Chen, Y.; Xue, W.; Lou, Y. Consistent gene signature of schizophrenia identified by a novel feature selection strategy from comprehensive sets of transcriptomic data. *Brief. Bioinform.* **2020**, *21*, 1058–1068. [CrossRef] [PubMed]
87. Li, Y.H.; Li, X.X.; Hong, J.J.; Wang, Y.X.; Fu, J.B.; Yang, H.; Yu, C.Y.; Li, F.C.; Hu, J.; Xue, W.W. Clinical trials, progression-speed differentiating features and swiftness rule of the innovative targets of first-in-class drugs. *Brief. Bioinform.* **2020**, *21*, 649–662. [CrossRef] [PubMed]
88. Zhu, F.; Qin, C.; Tao, L.; Liu, X.; Shi, Z.; Ma, X.; Jia, J.; Tan, Y.; Cui, C.; Lin, J. Clustered patterns of species origins of nature-derived drugs and clues for future bioprospecting. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 12943–12948. [CrossRef]
89. Yin, J.; Sun, W.; Li, F.; Hong, J.; Li, X.; Zhou, Y.; Lu, Y.; Liu, M.; Zhang, X.; Chen, N. VARIDT 1.0: Variability of drug transporter database. *Nucleic Acids Res.* **2020**, *48*, D1042–D1050. [CrossRef]
90. Xue, X.; Wang, S.F.; Zhan, L.J.; Feng, Z.Y.; Guo, Y.D. Social Learning Evolution (SLE): Computational Experiment-Based Modeling Framework of Social Manufacturing. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3343–3355. [CrossRef]
91. Xue, X.; Chen, Z.; Wang, S.; Feng, Z.; Duan, Y.; Zhou, Z. Value Entropy: A Systematic Evaluation Model of Service Ecosystem Evolution. *IEEE Trans. Serv. Comput.* **2020**. [CrossRef]
92. Wu, Z.; Li, R.; Xie, J.; Zhou, Z.; Guo, J.; Xu, X. A user sensitive subject protection approach for book search service. *J. Assoc. Inf. Sci. Technol.* **2020**, *71*, 183–195. [CrossRef]
93. Wu, Z.; Shen, S.; Lian, X.; Su, X.; Chen, E. A dummy-based user privacy protection approach for text information retrieval. *Knowl.-Based Syst.* **2020**, *195*, 105679. [CrossRef]
94. Wu, Z.; Shen, S.; Zhou, H.; Li, H.; Lu, C.; Zou, D. An effective approach for the protection of user commodity viewing privacy in e-commerce website. *Knowl.-Based Syst.* **2021**, *220*, 106952. [CrossRef]
95. Qiu, S.; Hao, Z.; Wang, Z.; Liu, L.; Liu, J.; Zhao, H.; Fortino, G. Sensor Combination Selection Strategy for Kayak Cycle Phase Segmentation Based on Body Sensor Networks. *IEEE Internet Things J.* **2021**, *in press*. [CrossRef]
96. Zhang, L.; Zou, Y.; Wang, W.; Jin, Z.; Su, Y.; Chen, H. Resource Allocation and Trust Computing for Blockchain-Enabled Edge Computing System. *Comput. Secur.* **2021**, *105*, 102249. [CrossRef]
97. Zhang, L.; Zhang, Z.; Wang, W.; Waqas, R.; Zhao, C.; Kim, S.; Chen, H. A Covert Communication Method Using Special Bitcoin Addresses Generated by Vanitygen. *Comput. Mater. Contin.* **2020**, *65*, 597–616.
98. Zhang, L.; Zhang, Z.; Wang, W.; Jin, Z.; Su, Y.; Chen, H. Research on a Covert Communication Model Realized by Using Smart Contracts in Blockchain Environment. *IEEE Syst. J.* **2021**, *in press*. [CrossRef]
99. Wu, Z.; Li, G.; Shen, S.; Cui, Z.; Lian, X.; Xu, G. Constructing dummy query sequences to protect location privacy and query privacy in location-based services. *World Wide Web* **2021**, *24*, 25–49. [CrossRef]
100. Wu, Z.; Wang, R.; Li, Q.; Lian, X.; Xu, G. A location privacy-preserving system based on query range cover-up for location-based services. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5244–5254. [CrossRef]
101. Qiu, S.; Zhao, H.; Jiang, N.; Wu, D.; Song, G.; Zhao, H.; Wang, Z. Sensor network oriented human motion capture via wearable intelligent system. *Int. J. Intell. Syst.* **2021**, *37*, 1646–1673. [CrossRef]
102. Gandomi, A.; Yang, X.-S.; Alavi, A.; Talatahari, S. Bat algorithm for constrained optimization tasks. *Neural Comput. Appl.* **2013**, *22*, 1239–1255. [CrossRef]
103. He, Q.; Wang, L. A Hybrid Particle Swarm Optimization with a Feasibility-based Rule for Constrained Optimization. *Appl. Math. Comput.* **2007**, *186*, 1407–1422. [CrossRef]
104. Kaveh, A.; Talatahari, S. A Novel Heuristic Optimization Method: Charged System Search. *Acta Mech.* **2010**, *213*, 267–289. [CrossRef]
105. He, Q.; Wang, L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng. Appl. Artif. Intell.* **2007**, *20*, 89–99. [CrossRef]
106. Kaveh, A.; Talatahari, S. An improved ant colony optimization for constrained engineering design problems. *Eng. Comput.* **2010**, *27*, 155–182. [CrossRef]
107. Mezura-Montes, E.; ACoello Coello, C.; Velázquez-Reyes, J.; Muñoz-Dávila, L. Multiple trial vectors in differential evolution for engineering design. *Eng. Optim.* **2007**, *39*, 567–589. [CrossRef]

108. Sandgren, E. Nonlinear integer and discrete programming in mechanical design. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Kissimmee, FL, USA, 25–28 September 1988; The American Society of Mechanical Engineers: New York, NY, USA, 1988; Volume 14.
109. Wagdy, A. A novel differential evolution algorithm for solving constrained engineering optimization problems. *J. Intell. Manuf.* **2018**, *29*, 659–692.
110. Rao, V.R.; Savsani, J.V.; Vakharia, P.D. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **2011**, *43*, 303–315. [CrossRef]
111. Kentli, A.; Sahbaz, M. Optimisation of Hydrostatic Thrust Bearing Using Sequential Quadratic Programming. *Oxid. Commun.* **2014**, *37*, 1144–1152.
112. He, S.; Prempain, E.; Wu, Q. An improved particle swarm optimizer for mechanical design optimization problems. *Eng. Optim.* **2004**, *36*, 585–605. [CrossRef]
113. Kaveh, A.; Khayatazad, M. A new meta-heuristic method: Ray Optimization. *Comput. Struct.* **2012**, *112–113*, 283–294. [CrossRef]
114. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
115. Huang, F.-Z.; Wang, L.; He, Q. An effective co-evolutionary differential evolution for constrained optimization. *Appl. Math. Comput.* **2007**, *186*, 340–356. [CrossRef]
116. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
117. Coello Coello, C.A. Use of a self-adaptive penalty approach for engineering optimization problems. *Comput. Ind.* **2000**, *41*, 113–127. [CrossRef]
118. Sandgren, E. Nonlinear Integer and Discrete Programming in Mechanical Design Optimization. *J. Mech. Des.* **1990**, *112*, 223–229. [CrossRef]
119. Mezura-Montes, E.; Coello, C.A.C. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *Int. J. Gen. Syst.* **2008**, *37*, 443–473. [CrossRef]