Improvement of Meta-heuristic Algorithms from Hybridization and Novel Selection Operator

by

Zhentao Tang

A dissertation

submitted to the Graduate School of Science and Engineering for Education

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Engineering

Supervisor: Prof. Zheng Tang

Associate Supervisors: Assoc. Prof. Shangce Gao

University of Toyama

Gofuku 3190, Toyama-shi, Toyama 930-8555 Japan

2023

(Submitted January 30, 2023)

Acknowledgements

In an instant, the three-year doctoral student life was coming to an end and graduation was about to be ushered in. In the past three years of life and study, I have learned a lot, and I have a lot of emotions. The little bit of three years seems to be vivid, and I have grown a lot. In particular, I met a group of like-minded students, worked hard and worked together; and the teachers who taught carefully and achieved excellent results, with their guidance and help, I learned a lot and successfully completed the learning tasks. At the same time, it also laid the foundation for better adaptation to society. Here, I would like to express my most sincere gratitude to the teachers and classmates who helped me and cared for me, and thank you all.

First, I would like to thank my mentor, Professor Zheng Tang, for the care and encouragement to me in my studies and life, and for guiding me in my research. From the first day of my enrollment, Teacher Tang taught me that I must be rigorous and pragmatic in scientific research, read it extensively, practice diligently, do more experiments, find inspiration in the literature, and verify ideas in the experiments. Cultivation made me develop good research habits. At the same time, he also taught us a lot of experience in entering society. Here I would like to express my thanks and blessings to Teacher Tang. Teacher Tang's rigorous scientific research attitude and meticulous attention to detail have always influenced me, and it will be one of my most valuable assets in life.

Secondly, I would like to thank Associate Professor Shangce Gao for his academic guidance and help. Teacher Tang and Teacher Gao together led our scientific research

team to make continuous progress. Teacher Gao's excellent scientific research ability and interpersonal skills have benefited me greatly. It's worth my constant learning. From the topic selection of the thesis to the completion of the thesis, Teacher Gao has been careful to guide, spent a lot of energy, let me successfully complete the experiment, help me to modify the thesis over and over, so that I successfully completed the writing of the thesis. In the meantime, I want to thank the classmates in my laboratory for their help in my study and life; thank you for giving me a rich and wonderful life; thank you for bringing me good memories.

Finally, I want to thank my parents and my wife in particular. Thank you for their silent support and selfless dedication, which is a strong backing on my life path. Allows me to study with confidence and complete my studies.

Abstract

There are many existing methods for algorithm improvement, and this paper focuses on two methods: hybridization and selection operator. In hybridization, spherical search (SS) is the one of newest proposed meta-heuristic algorithms. SS performs search effectively in exploration, but due to the lack of local exploitation ability, it converges slowly and can't exploit the small region around the current promising solution. This paper proposes a novel optimization algorithm, namely SSGSA, which is inherited from the SS and gravitational search algorithm (GSA) to combine the effective exploration and exploitation of each algorithm, respectively. To evaluate the effectiveness of SSGSA, we compared it with the original SS, original GSA, particle swarm optimization, and whale optimization algorithm on the IEEE CEC 2017 benchmark function suit. Experimental results show that the proposed new method outperforms its competitors in terms of convergence speed and solution accuracy. In selection operator, wind driven optimization (WDO) is a meta-heuristic algorithm based on swarm intelligence. The original selection method makes it easy to converge prematurely and trap in local optima. Maintaining population diversity can solve this problem well. Therefore, we introduce a new fitness-distance balance-based selection strategy to replace the original selection method, and add chaotic local search with selecting chaotic map based on memory to further improve the search performance of the algorithm. A chaotic wind driven optimization with fitness-distance balance strategy is proposed, called CFDBWDO. In the experimental section, we find the optimal parameter settings for the proposed CFDBWDO. In order to verify the effect

of the CFDBWDO, we conduct comparative experiments on the CEC 2017 benchmark functions. The experimental results denote that the proposed CFDBWDO has superior performance. Compared with WDO, CFDBWDO can gradually converge in function optimization. We further verify the practicality of the proposed CFDBWDO with six real-world optimization problems, and the obtained results are all better than other algorithms.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nature can be regarded as a huge ecosystem with a large number of subsystems [1–3]. These self-contained environments have existed on the earth for millions of years. Through millions of years of accumulation and evolution, nature has become the largest information resource library on the earth, bringing unparalleled inspiration and strength to many scientists engaged in scientific research [4, 5]. In the past forty years, researchers have drawn inspiration from various biological systems and the behavioral characteristics of various organisms in biological systems, and proposed many biological heuristic models, and used these models to develop many versatile optimization methods, including meta-heuristic algorithms [6, 7]. Since the early 1970s, many proven effective evolutionary methods [8–10] have been proposed. For example, ant colony optimization [11–13] is derived from the simulation of ants flocking process. Genetic algorithm [14] is derived from the simulation of biological evolution process in nature. Particle swarm optimization [15, 16] is derived from the simulation of birds flocking process. Other representative ones include gravitational search algorithm [17, 18], immune algorithm [19, 20], simulated annealing [21], and grey wolf optimizer [22], etc. Meta-heuristic algorithm has excellent global search ability, strong adaptability and robustness. It has been successfully applied to many fields, such as optimization computing [23–28], artificial intelligence [29, 30], pattern recognition [31, 32], image processing [33, 34], constrained optimization [35–37], engi-

neering [38, 39] and biology [40, 41].

Spherical search (SS) [42] is a new meta-heuristic proposed by Abhishek et al. in 2019. This algorithm has the advantages of few parameters and can maintain the diversity of solutions during the optimization process, so it has become a new hot spot in computational intelligence research. With the continuous in-depth study of the basic theory and application of the SS, many shortcomings of the algorithm have also been gradually discovered: it has a slower convergence speed in the later iterations, and it is not easy to find the optimal solution such that the solution accuracy is low. It is of great practical significance for improving the performance of the algorithm and expanding the application field of the algorithm. The issue of balancing the exploration and exploitation is a very challenging problem, but a well balance can improve the accuracy and convergence speed of an algorithm. Motivated by this, we aim to hybridize the spherical search with one of the very effective search algorithms, gravitational search [17, 43, 44], which has shown great capacity in exploration search ability [45–48].

Inherited from spherical and gravitational search dynamics, the proposed SSGSA is expected to possess a well-balance between the global exploitation search capacity from SS and local exploration ability from gravitational search algorithm (GSA), thus being able to find better solutions within reasonable search time. The two search dynamics are performed in a parallel manner: in the early search phase, SS is responsible for the whole search space exploitation within a relative wide search area, aiming to find the most promising area for the latter's search. Along with the iteration of evolution, the whole population is divided into two subpopulations: one inherited from SS perform global exploitation, while the other inherited from GSA utilizes the gravitational attraction to make all individual more toward the current global optimal one, thus enabling the search to achieve a good balance between exploitation and

exploration. Experimental results based on IEEE CEC 2017 benchmark optimization functions show the effectiveness of fast convergence speed of the proposed method.

The contribution of this improvement method can be summarized as follows: 1) To our best knowledge, we are for the first time to consider an algorithm inherited search dynamics from both spherical search and gravitational search, and successfully utilize both characteristics of each search dynamic, thus achieving an efficient search ability. 2) This study reveals that the co-evolution of two carefully selected search algorithms can perform better than each single one, which gives more insights into the key problems of how to integrate different search dynamics to improve the search performance of optimization.

Wind driven optimization (WDO) [49] is a meta-heuristic algorithm similar to particle swarm optimization and is derived from simulating the process that the air particles continuously move due to different air pressures until air pressure balance [50]. WDO has the advantages of simple structure, few parameters, easy to understand and program. Since the algorithm was proposed, more and more scholars have paid attention by applying it on many scenarios and improved it from many aspects. In [49], wind driven optimization was tested on the benchmark functions and applied to the design for the double-sided Artificial Magnetic Conducting (AMC) surface for the first time. It simply proves the effectiveness and practicality of WDO. In [51], WDO was applied to the stub-loaded inverted-F antenna synthesis. It indicates that WDO performs well on electromagnetics optimization problems. In [52], WDO was applied to the design of high-impedance metasurfaces. It demonstrates that WDO is a useful tool for designing high-impedance metasurfaces. In [53], it was utilized to solve unimodal and multimodal functions, and applied to three electromagnetic optimization problems. The results show that WDO is better than some common evolutionary algorithms in electromagnetics on many cases. In [54], the collision avoidance technol-

ogy and multi-region concept in meteorology were introduced into WDO to maintain population diversity. Accordingly, a new multi-region and anti-collision wind driven optimization variant was presented, and the improved algorithm was applied to dynamic optimization problems. Comparison between it and the existing methods on moving peak benchmark illustrates that its search ability needs to be further improved. In [55], wind driven optimization with levy flights (WDOLE) was employed on global continuous optimization problem. Although WDOLE performs better than WDO on benchmark functions, it is criticized that just comparing WDO is not enough to verify the superior performance of WDOLE. In [56], it was shown that WDO had the worst performance in quadrature mirror filter bank design among the existing technologies. In [57], a hybrid wind driven optimization and differential evolution was presented. Although the algorithm can find better solutions, it still has the problem of premature convergence from the perspective of the convergence graphs, and the algorithm does not optimize real-world problems and lacks practicality. In [58], covariance matrix adaptive evolutionary strategy (CMAES) was utilized to update parameters, and an adaptive wind driven optimization algorithm (AWDO) was proposed. Although AWDO has no parameter settings, the faster convergence speed will cause the algorithm to easily fall into local optima. In [59], Pareto was embedded into an adaptive wind driven optimization (MO-AWDO) method to handle multi-objective optimization problems. Experimental results exhibit that MO-AWDO can surpass some well-known multi-objective algorithms. In [60], WDO was applied to the design of switched reluctance motors, which is a multi-objective optimization problem. In [61], WDO was applied to the estimation of solar photovoltaic parameters. The results obtained by WDO display better accuracy. In [62], AWDO based on Chenlo's model was applied to the extracting the parameters of photovoltaic cell model. It is an efficient and reliable method. In [63], a hybrid wind driven optimization and

cuckoo search was used to solve hyperspectral band selection problems. However, its initialization strategy should be further improved. Following the chronological order of the literature, we list some references on the improvement and application of WDO, and provide the comments of the cited papers. To sum up, WDO is effective in the field of electromagnetic and solar photovoltaic parameter estimation. In the improvement of WDO, most of them use hybrid algorithm to improve exploration and exploitation abilities, and it has not changed the selection strategy and working mechanism of WDO.

The No-Free-Lunch theorem [64, 65] proves that no algorithm performs best for various optimization problems, which motivates us to improve WDO and solve various optimization problems more efficiently. In WDO, air particles are easily attracted by a local optimal point and fall into local optima. Therefore, if the selection strategy of WDO is not changed, it is difficult to solve this defect. Changing the original selection strategy to enrich the population diversity and improve the possibility of different solutions is a way to jump out of the local optimization.

Motivated by this, in this study we try to change the selection method and working mechanism of the algorithm to improve its search performance. To be specific, although WDO has few parameters and is easily implemented, when solving optimization problems, the convergence speed of the WDO algorithm is too fast to converge to the local optimal solution prematurely, which makes the entire population lack diversity and misses better solutions. Maintaining the diversity of the population is one of the most important means to avoid premature convergence. Thus, a novel strategy fitness-distance balance is introduced in this work to maintain population diversity. In order to further improve the performance of the algorithm, taking advantage of the randomicity and ergodicity of chaos [45], chaotic local search (CLS) [66, 67] is also incorporated sophisticately as well, which can well balance the exploration and

exploitation abilities. Additionally, computational time complexity analysis shows that CFDBWDO is computationally efficient. Finally, 29 IEEE CEC2017 benchmark functions are utilized to find the best parameter settings of CFDBWDO and examine the superior performance of CFDBWDO. Six real-world optimization problems are optimized to indicate its practicality.

The main contributions of the CFDBWDO are as follows: (1) The WDO algorithm is prone to fall into local optima. The fitness distance balance strategy is applied to the WDO, which increases the diversity of the population, jumps out of the local optima, explores better solutions, and improves the exploration ability of WDO. (2) Chaotic local search with memory-based selection can make fully use of the performance of each chaotic map to adaptively choose the best chaotic map for chaotic local search, which enhances the exploitation ability of WDO, and well balances the exploration and exploitation abilities to further improve its performance. (3) Aiming at the disadvantages of the original WDO selection strategy, we replace the original selection strategy with a new selection strategy FDB, and propose the CFDBWDO algorithm. Extensive experimental results based on performance comparison with other state-of-the-art algorithms together with the statistical results show the superiority and practicality of the proposed CFDBWDO.

This paper is organized as follows: after the introduction, section 2 describes original SS and WDO. Section 3 introduces the proposed SSGSA and CFDBWDO. Section 4 conducts several comparative experiments and analyses. Finally, section 5 draws research conclusions and indicates future studies.

Chapter 2

Meta-heuristic Algorithms

2.1 SS

In SS, the search direction $\overline{z}_i$ of the i-th solution is calculated first, which is generated by the positions of three random individuals:

$$\overline{z}_i^{(k)} = (\overline{x}_t^{(k)} + r_1^{(k)} - r_2^{(k)}) - \overline{x}_i^{(k)} \tag{2.1}$$

where $\overline{x}_t^{(k)}$ is the initial target point, $r_1^{(k)}$ and $r_2^{(k)}$ are two random solutions in the population.

In order to balance exploration and exploitation, in SS, the search direction $\overline{z}_i$ of the top half of the population with a better solution is calculated using the towards-rand:

$$\overline{z}_i^{(k)} = (\overline{x}_{p_i}^{(k)} + r_1^{(k)} - r_2^{(k)}) - \overline{x}_i^{(k)} \tag{2.2}$$

where $\overline{x}_{p_i}^{(k)}$,$r_1^{(k)}$ and $r_2^{(k)}$ are randomly selected and $p_i \neq r_1 \neq r_2 \neq i$.

For the remaining half of the population, can use towards-best to calculate $\overline{z}_i$:

$$\overline{z}_i^{(k)} = (\overline{x}_{pbest_i}^{(k)} + r_1^{(k)} - r_2^{(k)}) - \overline{x}_i^{(k)} \tag{2.3}$$

where $\overline{x}_{pbest_i}^{(k)}$ is the randomly selected individual in the top p solutions. Therefore,

the search direction makes a set of better solutions diversified and enhances the exploration ability. It makes inferior solutions work hard to improve the quality of the solutions and enhance the exploitation ability of the algorithm.

Finally, use step-size control vector c_i and projection matrix P_i to calculate trial solutions $\overline{y}_i$:

$$\overline{y}_i^{(k)} = \overline{x}_i^{(k)} + c_i^{(k)} P_i^{(k)} \overline{z}_i^{(k)} \tag{2.4}$$

where P_i is projection matrix that determines the value of $\overline{y}_i$ on the spherical boundary. Different P_i produces different $\overline{y}_i$, and the trajectory of $\overline{y}_i$ produces spherical boundary. $P = A' diag(\overline{b}_i) A$, where A is orthogonal matrix and $diag(\overline{b}_i)$ is binary diagonal matrix ($0 < rank(diag(\overline{b}_i)) < D$). c_i is step-size control vector and $0.5 < c_i < 0.7$.

Algorithm 1: SS

```
/* Initialization */
while nFES ≤ FES do
  for i = 1 to N do
    if i < 0.5 * N then
      z̄_i ← TowardsRand(i) /* Better exploration */
    else
      z̄_i ← TowardsBest(i) /* Better exploitation */
    end if
    ȳ_i^(k) = x̄_i^(k) + c_i^(k) P_i^(k) z̄_i^(k)
    nFES ← nFES + 1
    x̄_i ← Selection(x̄_i, ȳ_i)
  end for
end while
```

2.2 WDO

According to the knowledge of atmospheric dynamics, the composition and structure of the atmosphere are very complex, and any tiny part (air cluster) can be treated as a "particle", called air particle. The WDO algorithm takes this abstract air particle as the research object, assuming that it is in a hydrostatic equilibrium state and satisfies

the ideal gas state equation. In addition, the horizontal movement in the atmospheric movement is stronger than the vertical movement, so the WDO algorithm only studies the horizontal movement of the wind.

When analyzing the movement of air particles [68,69], we take the air particle per unit volume as the research object, and then use Newton's second law and the ideal gas state equation to obtain:

$$\rho \boldsymbol{a} = \sum \boldsymbol{F}_i \tag{2.5}$$

$$P = \rho RT \tag{2.6}$$

where ρ is the air density. $\boldsymbol{a}$ represents acceleration vector. $\boldsymbol{F}_i$ is the force acting on air particle. P is pressure value. R is ideal gas constant. T is temperature.

$\boldsymbol{F}_i$ is the force on the air particle, four main forces are considered: (1) Pressure gradient force ($\boldsymbol{F}_{PG}$) is generated due to the different pressures in different positions, which makes the air particles move along the high-pressure position to the low-pressure position [70, 71], and the direction is from the high-pressure position to the low-pressure position. (2) Friction force ($\boldsymbol{F}_F$), its direction is opposite to the direction of $\boldsymbol{F}_{PG}$, here is a simplified friction force expression. (3) Gravitational force ($\boldsymbol{F}_G$) points vertically to the center of the earth. In the three-dimensional coordinate system of physics, it is the force that moves the air particle to the origin of the coordinate system. Similarly, the problem is mapped to the N dimensions, and gravitational force represents the force pointing to the center of the coordinate system. (4) Coriolis force ($\boldsymbol{F}_C$) is produced by the earth rotation. In WDO, it represents that the velocity and position of the current dimension are affected by motions in other

dimensions. Their simplified expressions are as follows:

$$\boldsymbol{F}_{\mathrm{PG}} = -\nabla P \delta V \tag{2.7}$$

$$\boldsymbol{F}_{\mathrm{F}} = -\rho \alpha \boldsymbol{u} \tag{2.8}$$

$$\boldsymbol{F}_{\mathrm{G}} = \rho \delta V \boldsymbol{g} \tag{2.9}$$

$$\boldsymbol{F}_{\mathrm{C}} = -2\Omega \times \boldsymbol{u} \tag{2.10}$$

where $-\nabla P$ represents air pressure gradient, the minus sign indicates the direction of gradient drop. δV represents finite volume of air. α is the coefficient of friction. $\boldsymbol{u}$ is velocity vector of air particle. $\boldsymbol{g}$ is gravitational acceleration. Ω is earth rotation.

Bring the above four equations into the right side of Eq. (2.5), and combine the acceleration equation: $\boldsymbol{a} = \frac{\boldsymbol{\Delta u}}{\Delta t}$, Eq. (2.5) can be changed to:

$$\rho \frac{\boldsymbol{\Delta u}}{\Delta t} = \rho \delta V \boldsymbol{g} + (-\nabla P \delta V) + (-\rho \alpha \boldsymbol{u}) + (-2\Omega \times \boldsymbol{u}) \tag{2.11}$$

Since the research object is the air particle per unit volume, take $\delta V = 1$. For the sake of simplicity, take $\Delta t = 1$ and substitute into Eq. (2.11) to obtain:

$$\rho \boldsymbol{\Delta u} = \rho \boldsymbol{g} - \nabla P - \rho \alpha \boldsymbol{u} + (-2\Omega \times \boldsymbol{u}) \tag{2.12}$$

Substituting Eq. (2.6) into Eq. (2.12):

$$\frac{P_{cur}}{RT} \boldsymbol{\Delta u} = \frac{P_{cur}}{RT} \boldsymbol{g} - \nabla P - \frac{P_{cur}}{RT} \alpha \boldsymbol{u} + (-2\Omega \times \boldsymbol{u}) \tag{2.13}$$

where P_{cur} is the pressure value at the current position. Dividing both sides of Eq. (2.13) by $\frac{P_{cur}}{RT}$:

$$\boldsymbol{\Delta u} = \boldsymbol{g} - \nabla P \frac{RT}{P_{cur}} - \alpha \boldsymbol{u} + \left(-\frac{2\Omega \times \boldsymbol{u} RT}{P_{cur}} \right) \tag{2.14}$$

Because $\boldsymbol{\Delta u} = \boldsymbol{u_{new}} - \boldsymbol{u_{cur}}$, $\boldsymbol{u_{new}}$ is the updated velocity vector of air particle, $\boldsymbol{u_{cur}}$ is the current velocity vector of air particle, Eq. (2.14) can be written as:

$$\boldsymbol{u_{new}} = \boldsymbol{g} - \nabla P \frac{RT}{P_{cur}} + (1-\alpha)\boldsymbol{u_{cur}} + \left(-\frac{2\Omega \times \boldsymbol{u}RT}{P_{cur}}\right) \tag{2.15}$$

The direction of the gravitational force is from the current position of air particle to the center of the coordinate system, and the coordinate of the center of the coordinate system is 0, so the vector $\boldsymbol{g}$ can be expressed as $\boldsymbol{g} = |\boldsymbol{g}|\,(0 - \boldsymbol{x_{cur}})$, where $\boldsymbol{x_{cur}}$ is the current position of air particle. Similarly, the direction of the pressure gradient $-\nabla P$ is from the current position to the optimal position, then $-\nabla P$ can be expressed as $-\nabla P = |P_{opt} - P_{cur}|\,(x_{opt} - x_{cur})$, where P_{opt} is the current optimal pressure value. $\boldsymbol{x_{opt}}$ is the optimal position. Due to the coriolis force ($\boldsymbol{F}_C$), $u_{\text{cur}}^{\text{other dim}}$ indicates that the velocity of the current dimension is affected by other dimensions, that is, the velocity in coriolis force is $u_{\text{cur}}^{\text{other dim}}$. And $c = -2|\Omega|RT$ [72], where c is a constant. So Eq. (2.15) can be changed to:

$$\begin{aligned} \boldsymbol{u_{new}} = &- g\boldsymbol{x_{cur}} + |P_{opt} - P_{cur}|\,(x_{opt} - x_{cur})\,\frac{RT}{P_{cur}} \\ &+ (1-\alpha)\boldsymbol{u_{cur}} + \left(\frac{-cu_{\text{cur}}^{\text{other dim}}}{P_{cur}}\right) \end{aligned} \tag{2.16}$$

The pressure values P_{opt} and P_{cur} in Eq. (2.16) may make the updated velocity become very large and lack of operability. The pressure values can be replaced by i, where i is the rank of the air particle based on the pressure value, so the velocity update equation is:

$$\begin{aligned} \boldsymbol{u_{new}} = &- g\boldsymbol{x_{cur}} + RT\left|1 - \frac{1}{i}\right|(x_{opt} - x_{cur}) \\ &+ (1-\alpha)\boldsymbol{u_{cur}} + \frac{cu_{\text{cur}}^{\text{other dim}}}{i} \end{aligned} \tag{2.17}$$

Then the position of air particle is updated as follows:

$$\boldsymbol{x_{new}} = \boldsymbol{x_{cur}} + (\boldsymbol{u_{new}} \times \Delta t) \tag{2.18}$$

where $\boldsymbol{x_{new}}$ is the updated position, assuming the time interval $\Delta t = 1$.

In order to prevent the moving step of air particle from being too large, the position is restricted between the upper and lower boundaries of the problem, and the updated velocity is limited to the range of $[-u_{\max}, u_{\max}]$ by Eq. (2.19):

$$u^*_{\text{new}} = \begin{cases} u_{\max} & if \quad u_{\text{new}} > u_{\max} \\ -u_{\max} & if \quad u_{\text{new}} < -u_{\max} \end{cases} \tag{2.19}$$

where u^*_{new} denotes the adjusted velocity limited to maximal velocity, and $u_{\max}$ is the maximal velocity.

WDO derives the velocity and position equations of air particles by applying Newton's second law and the ideal gas equation of state. It has fewer adjustable parameters and is easy to implement. It has the characteristics of strong global search ability, fast convergence speed and high optimization efficiency.

Some basic information to help understand the proposed algorithm in this paper is described below:

(1) The population is composed of N individuals, each individual X_i is expressed as $X_i = \left(x_i^1, x_i^2, \ldots, x_i^d\right)$, $i \in \{1, 2, \ldots, N\}$, where x_i^d denotes the position of the ith individual in the dth dimension.

(2) In order to enhance the robustness for dealing with optimization problems, 12 different chaotic maps are used (i.e., $J = 12$) in this paper. The 12 chaotic maps are: Logistic map, Piecewise linear chaotic map (PWLCM), Singer map, Sine map, Gaussian map, Tent map, Bernoulli map, Chebyshev map, Circle map,

Algorithm 2: WDO

Input: Parameters N, d, T_{max}, α, g, RT, c, $u_{\max}$
Output: Optimal solution
Initialization: Generate N air particles randomly;
Fitness calculation: Calculate fitness of air particle;
while $k < T_{max}$ *(maximum iteration count)* **do**
  for i = *1 to* N **do**
    Rank the air particles based on pressure values (fitness values);
    Update velocity of air particle by Eq. (2.17) and check velocity limits by Eq. (2.19);
    Update position of air particle by Eq. (2.18) and check boundaries;
    Calculate fitness of air particle;
    if $f(X_i) < f(X_{opt})$ **then**
      $X_{opt} = X_i$;
    end
  end
  $k = k + 1$;
end

Cubic map, Sinusoidal map and Iterative chaotic map with infinite collapses (ICMIC). The determination equations of these chaotic maps are in reference [73].

(3) Existing selection strategies include: random selection, greedy selection, probabilistic selection, adaptive selection, fitness-distance balance selection and combined selection, where fitness-distance balance selection is the latest selection method, and the combined selection includes at least two of the other selection strategies.

Chapter 3

Improvement of Meta-heuristic Algorithms from Hybridization and Novel Selection Operator

3.1 Proposed SSGSA

Gravitational search algorithm (GSA) is a heuristic optimization algorithm proposed by Esmat et al. in 2009. It is derived from a swarm intelligence algorithm that simulates the gravitation in physics. The theory is to treat search particles as a group of objects moving in space. These objects are attracted by gravitational interaction, and the movement of objects follows the law of dynamics. A particle with a larger fitness has a larger inertial mass, so the universal gravitation will cause the objects to move toward the most massive object, thereby gradually approaching the optimal solution. GSA has strong local exploitation ability and convergence speed. With the deepening of the theoretical research of GSA, its application becomes more and more extensive. In this study, the strong exploitation ability of GSA is incorporated to SS to improve the quality of solution.

The proposed SSGSA is illustrated in Algorithm 2, where we set the maximum iteration to be 3000, the population size to be 100. In the first 1500 iterations, the SS algorithm is used to perform the search, aiming to find a promising search area,

Algorithm 3: SSGSA

```
/* Initialization */
while nFES < FES do
  for i = 1 to N do
    if nFES> 1/2FES then
      if i < 0.5 * N then
        /* Update The towards − rand Part Of The Population Using SS */
      else
        /* Update The towards − best Part Of The Population Using GSA */
      end if
    else
      /* Update The Population Using SS */
    end if
    nFES ← nFES + 1
    x̄_i ← Selection(x̄_i, ȳ_i)
  end for
end while
```

while both search dynamics are utilized in the last 1500 iterations. In the early search stage, SSGSA is performed via Eqs. (1)-(4). The first half of the population, i.e., $N/2$ better solutions in the population are calculated by towards-rand in SS, and the last $N/2$ worse solutions are updated using GSA instead of the original towards-best via the following evolution:

Its mass $M_i(t)$ is defined as:

$$M_i(t) = \frac{m_i(t)}{\sum_{l=1}^{N} m_l(t)} \tag{3.1}$$

$$m_i(t) = \frac{f_i(t) - f_w(t)}{f_b(t) - f_w(t)} \tag{3.2}$$

where $f_i(t)$ is the fitness of the ith individual at iteration t, $f_w(t)$ and $f_b(t)$ represent the worst and best fitness values in the current population, respectively. The gravitational force $F_{ij}^d(t)$ is generated by X_i and X_j based on mass, formulated as:

$$F_{ij}^d(t) = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon}(x_j^d(t) - x_i^d(t)) \tag{3.3}$$

where $R_{ij}(t)$ is Euclidean distance between X_i and X_j. ε is a small value. $G(t)$ is a gravitational constant with time, defined as

$$G(t) = G_0 \times e^{-\alpha \frac{t}{T}} \tag{3.4}$$

where G_0 is an initial value, α is a constant, t is current iteration index and T is maximum iteration count. The total force that acts on X_i in the dth dimension is obtained as

$$F_i^d(t) = \sum_{j \in K_{best}, j \neq i} rand_j \cdot F_{ij}^d(t) \tag{3.5}$$

where K_{best} is a set of the first K best individuals in the current population, and K is linearly decreased with t. $rand_j$ is a random value in the interval [0,1]. Therefore, the acceleration $a_i^d(t)$ in the dth dimension is

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{3.6}$$

Finally, the velocity and position of X_i at iteration $t+1$ are performed as

$$v_i^d(t+1) = rand_i \cdot v_i^d(t) + a_i^d(t) \tag{3.7}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{3.8}$$

where $rand_i$ is a random variable in the interval [0,1].

Although the towards-best of the SS algorithm and GSA have good local exploitation ability, it can be seen in the following experimental part that GSA can find the optimal solution better than the towards-best. Therefore, the quality of the SSGSA solution can be improved.

3.2 Proposed CFDBWDO

3.2.1 Motivation

WDO is a swarm intelligence-based [74, 75] meta-heuristic algorithm. Swarm intelligence algorithms have two behaviors, exploration and exploitation [76, 77]. Exploration is to randomly generate an individual in the search space to explore a promising solution that is not neighbor to the current best solution, which helps to jump out of the current local optima. The exploitation is to search promising solutions in a small region near the current optimal solution, which conducts local search in a promising region and accelerates the convergence of the algorithm. The challenging issue that how to balance exploration and exploitation is the focus of improving the algorithm. WDO is derived from the simulation of air movement. Due to the different air pressures in different positions, pressure gradient force ($\boldsymbol{F}_{PG}$) is generated, which makes the air particles in the high pressure position move to the low pressure position until the air pressure in each position is equal. The existence of pressure gradient force enables air particles to find the optimal solution of optimization problem, but the pressure gradient force usually guides particles to find the local optimal solution, which makes the population fall into the local optima. This shows that the pressure gradient force cannot make the particles find a better solution, that is, the pressure gradient force in WDO can not help particles get rid of local optima, which easily causes WDO to converge prematurely. Due to the intrinsic search mechanism of WDO, the particles can not explore the entire search space, which limits the exploration and exploitation abilities of WDO and makes it easier to obtain local optimal solutions. Therefore, on the one hand, in order for particles to skip the local optimal solution and explore a better solution, it is necessary to maintain population diversity and avoid falling into the local optima. This paper introduces a new selection method fitness distance balance strategy to replace the original pressure

selection. On the other hand, to balance the exploration and exploitation abilities of WDO and further optimize the performance of the algorithm, chaotic local search is incorporated sophisticately. Accordingly, CFDBWDO is proposed based on these two improvements.

3.2.2 Chaotic local search (CLS) based on memory selection

The chaotic local search (CLS) is to use the chaotic map to search a promising solution locally around the current best individual to update the best individual. The mathematical equations of CLS are as follows:

$$X_{opt'}(k) = X_{opt}(k) + r\left(U_b - L_b\right)\left(z^j(k) - 0.5\right) \tag{3.9}$$

$$X_{opt}(k) = \begin{cases} X_{opt'}(k), & \text{if } f\left(X_{opt'}(k)\right) < f\left(X_{opt}(k)\right) \\ X_{opt}(k), & \text{otherwise} \end{cases} \tag{3.10}$$

where $X_{opt}(k)$ is the position of global best individual at the current iteration k, expressed as $X_{opt} = \left(x^1_{opt}, x^2_{opt}, \ldots, x^d_{opt}\right)$. $X_{opt'}(k)$ is position of a new individual generated by chaotic local search. r is the chaotic search radius within $(0, 1)$, where $r = 0.0001$. U_b and L_b denote the upper and lower bound vectors of the search space, respectively. If the value of $X_{opt'}(k)$ exceeds the boundary value, it is adjusted to the closest boundary value. $z^j(k)$ represents the chaotic sequence at iteration k, and it is generated by the jth chaotic map randomly chosen from 12 chaotic maps (i.e., $j \in \{1, 2, \ldots, 12\}$).

In Eq. (3.10), if the fitness of the updated $X_{opt'}(k)$ is better than the current best fitness of $X_{opt}(k)$, $X_{opt'}(k)$ will replace $X_{opt}(k)$, otherwise $X_{opt}(k)$ remains unchanged.

Then we set a learning iteration count (L) to 50. If the updated best individual ($X_{opt'}(k)$) replaces the original best individual ($X_{opt}(k)$), the memory will record a success, otherwise record a failure. The recording mechanisms are formulated as

follows:

$$\alpha_{k,j} = \begin{cases} \alpha_{k-1,j} + 1, & \text{if } f(X^j_{opt'}(k)) < f(X_{opt}(k)) \& k \leq L \\ \alpha_{k-1,j}, & \text{if } f(X^j_{opt'}(k)) \geq f(X_{opt}(k)) \& k \leq L \\ \alpha_{k-1,j} + 1 - (\alpha_{k-L,j} & \\ -\alpha_{k-L-1,j}), & \text{if } f(X^j_{opt'}(k)) < f(X_{opt}(k)) \& k > L \\ \alpha_{k-1,j} - (\alpha_{k-L,j} & \\ -\alpha_{k-L-1,j}), & \text{otherwise} \end{cases} \tag{3.11}$$

$$\beta_{k,j} = \begin{cases} \beta_{k-1,j}, & \text{if } f(X^j_{opt'}(k)) < f(X_{opt}(k)) \& k \leq L \\ \beta_{k-1,j} + 1, & \text{if } f(X^j_{opt'}(k)) \geq f(X_{opt}(k)) \& k \leq L \\ \beta_{k-1,j} - (\beta_{k-L,j} & \\ -\beta_{k-L-1,j}), & \text{if } f(X^j_{opt'}(k)) < f(X_{opt}(k)) \& k > L \\ \beta_{k-1,j} + 1 - (\beta_{k-L,j} & \\ -\beta_{k-L-1,j}), & \text{otherwise} \end{cases} \tag{3.12}$$

where $X^j_{opt'}(k)$ denotes position of an individual generated by jth chaotic map. $\alpha_{k,j}$ represents the number of successes. $\beta_{k,j}$ represents the number of failures of the jth chaotic map at iteration k. let $\alpha_{0,j} = 0$, $\beta_{0,j} = 0$. If $f(X^j_{opt'}(k))$ is less than $f(X_{opt}(k))$, $\alpha_{k-1,j}$ is added with 1, otherwise $\beta_{k-1,j}$ is added with 1. Since the memory space is only L, when k is greater than L, it is necessary to subtract $\alpha_{k-L,j}$ from $\alpha_{k,j}$ and subtract $\beta_{k-L,j}$ from $\beta_{k,j}$ to make space for the latest individual ($\alpha_{k+1,j}$ and $\beta_{k+1,j}$), so as to ensure that the memory only records the success and failure times of the last L iterations.

The selection method of J chaotic maps is described as follows. In L learning iterations, each chaotic map will be chosen with the same probability $1/J$, and the memory will record the number of successes and failures. After the learning iterations, the selection probabilities for chaotic maps can be calculated using the number of successes and failures, shown as follows:

$$S_{k,j} = \frac{\alpha_{k,j}}{\alpha_{k,j} + \beta_{k,j}} + \varepsilon \tag{3.13}$$

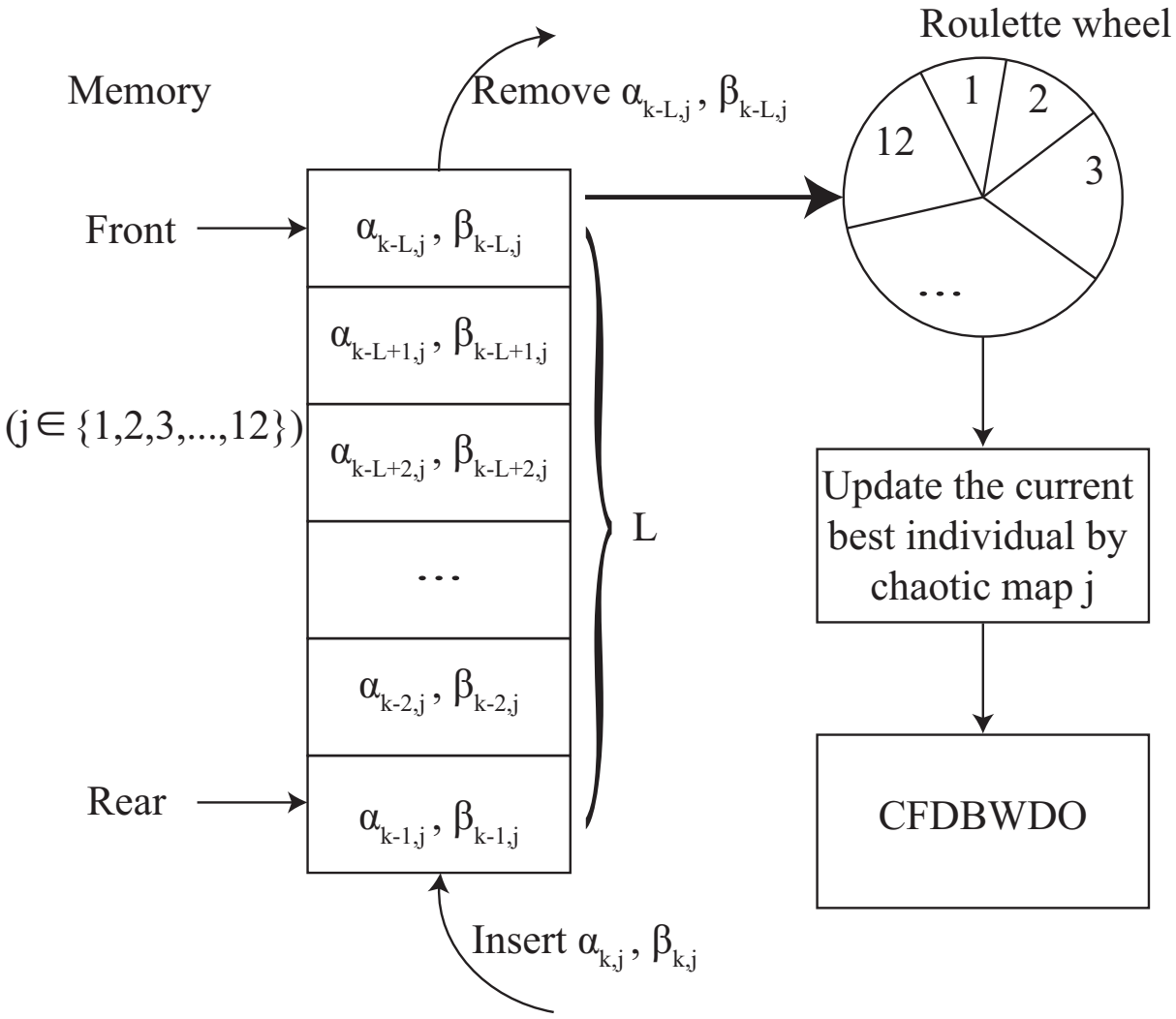

Figure 3.1: CLS with selecting chaotic map based on memory.

$$P_{k,j} = \frac{S_{k,j}}{\sum_{j=1}^{J} S_{k,j}} \tag{3.14}$$

where $S_{k,j}$ is the success rate of the jth chaotic map at iteration k, $P_{k,j}$ is the selection probability of the jth chaotic map. ε is a constant to prevent the null success rate, where $\varepsilon = 0.01$. The higher the success rate, the greater the probability of being selected through roulette wheel, which not only improves the efficiency of choosing chaotic maps, but also makes full use of the advantages of each chaotic map. Therefore, it can adaptively choose the best chaotic map to implement the local search, as

illustrated in Fig. 3.1.

3.2.3 Fitness-distance balance (FDB) selection strategy

Since the WDO algorithm easily converges to the local optimal solution by pressure selection and misses other better solutions. Population diversity is to explore many non-neighbor promising individuals in the search space, increase the possibility of different solutions, avoid falling into local solutions, and improve the solution quality. In this study, the fitness distance balance strategy [78,79] is used to replace the original elite selection method. The score is calculated for each individual and the individuals are ranked based on scores. The higher the score, the greater the contribution of the individual to solving the problem, the higher the probability of being selected, and the easier it is for the algorithm to generate a new optimal solution, thus skipping the local optima, effectively improving the quality of solutions. The following is a specific introduction to the fitness distance balance strategy.

First we calculate the fitness value of each individual with the following equation:

$$\forall_{i=1}^{N} F_i = \begin{cases} \text{if goal is minimization:} F_i = 1 - \text{ norm } G_i \\ \text{if goal is maximization:} F_i = \text{ norm } G_i \end{cases} \tag{3.15}$$

where F_i is fitness value, and $normG_i$ denotes normalized objective function value of the ith individual. The fitness vector F created for individuals is given in Eq. (3.16).

$$F = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix}_{N \times 1} \tag{3.16}$$

Then we use Eq. (3.17) to calculate the distance of each individual from the best

individual.

$$\begin{aligned} &\forall_{i=1}^{N}, i \neq opt, D_i \\ &= \sqrt{\left(x_i^1 - x_{opt}^1\right)^2 + \left(x_i^2 - x_{opt}^2\right)^2 + \cdots + \left(x_i^d - x_{opt}^d\right)^2} \end{aligned} \tag{3.17}$$

where D_i is the distance value of the ith individual (X_i) from the best individual (X_{opt}). The distance vector D created for individuals is given in Eq. (3.18).

$$D = \begin{bmatrix} d_1 \\ \vdots \\ d_N \end{bmatrix}_{N \times 1} \tag{3.18}$$

Finally, the fitness vector F and distance vector D are used to calculate the score.

$$\forall_{i=1}^{N} S_i = w * normF_i + (1 - w) * normD_i \tag{3.19}$$

$$S = \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}_{N \times 1} \tag{3.20}$$

where S_i is the score of the ith individual, w is the weight coefficient in the range of $[0.4, 0.6]$, where $w = 0.5$, $normF_i$ denotes the normalized fitness value, $normD_i$ denotes the normalized distance value of the ith individual, and S is the score vector of individuals.

This selection method takes into account the fitness and the distance from the current optimal solution to calculate the score for each individual. According to the score vector S, the population can be ranked, the individual with high score who contributes the most to the optimization problem is selected determinedly and consciously instead of random, greedy or probabilistic selection, even in the later stages of the iteration. It can also effectively maintain the diversity of the population and prevent premature convergence.

Algorithm 4: Steps of fitness-distance balance selection strategy

Input: Parameters N, d, w
Output: Score vector
for *i = 1 to N* **do**
    Calculate distance between X_i and X_{opt} by Eq. (3.17);
    Create the distance vector by Eq. (3.18);
end
for *i = 1 to N* **do**
    Normalize fitness and distance vectors within $[0, 1]$;
    Calculate the score of air particle by Eq. (3.19);
    Create the score vector by Eq. (3.20);
end
Rank the air particles based on score vector (S);

It is worth pointing out that FDB is utilized to enrich the diversity of the population for improving the exploration ability, while CLS is used to update the current best individual for improving the exploitation ability. Considering these, the exploration and exploitation abilities of WDO are expected to be improved, and the resultant chaotic wind driven optimization with fitness distance balance strategy (CFDBWDO) is proposed. In addition, velocity limitation and boundary checks are not implemented in CFDBWDO to simplify its implementation process. The flowchart of CFDBWDO is illustrated in Fig. 3.2.

The computational time complexity of CFDBWDO is calculated to analyze the efficiency of CFDBWDO, where the population size is N, described as follows: (1) Initialization process needs $O(N)$. (2) CLS takes $O(N)$. (3) FDB costs $O(N^2)$. (4) Updating the velocity and position of air particles requires $O(N)$. (5) Evaluating each air particle expends $O(N)$. Thus, time complexity of CFDBWDO can be regarded as $O(N^2)$ which is the same as WDO's. Accordingly, CFDBWDO is computationally efficient.

3.2.4 Advantages of CFDBWDO

To summarize, the CFDBWDO proposed in this study has the following advantages:

Algorithm 5: CFDBWDO

Input: Parameters N, d, $T_{\max}$, α, g, RT, c, r, L, ε, J, w
Output: Optimal solution

Initialization: Generate N air particles randomly;
Fitness calculation: Calculate fitness of air particle and obtain the best position (X_{opt});
while *not met stopping criterion* **do**
  for $j = 1$ *to* J **do**
    if $k \leq L$ **then**
      Choose a chaotic map with the same probability $1/J$;
    else
      Choose a chaotic map with the selection probability $P_{k,j}$;
    end
  end
  Use CLS to update the current best air particle X_{opt} by Eq. (3.9) and Eq. (3.10);
  if $f(X_{opt'}) < f(X_{opt})$ **then**
    Record a success by Eq. (3.11);
  else
    Record a failure by Eq. (3.12);
  end
  Update chaos success rate and selection probability by Eqs. (3.13) and (3.14);
  Rank the air particles using FDB (in Algorithm 4);
  for $i = 1$ *to* N **do**
    Update velocity and position of air particle by Eqs. (2.17) and (2.18);
    Calculate fitness of air particle;
    if $f(X_i) < f(X_{opt})$ **then**
      $X_{opt} = X_i$;
    end
  end
  $k = k + 1$;
end

(1) Fitness distance balance strategy (FDB) is a novel selection strategy, aiming to improve the exploration performance of the algorithm. According to the scores, the individuals are ranked, which is different from other selection methods. FDB strategy calculates the score of each individual in the entire optimization process, which effectively maintains the diversity of the population and avoids premature convergence.

(2) At each iteration, the success or failure of a chaotic map is recorded, so that at

the current iteration the most promising chaotic map can be adaptively selected from 12 different chaotic maps with the roulette wheel according to current search performance, which not only fully utilizes the performance of different chaotic maps to enhance the robustness when dealing with different problems, but also improves the efficiency of choosing chaotic maps.

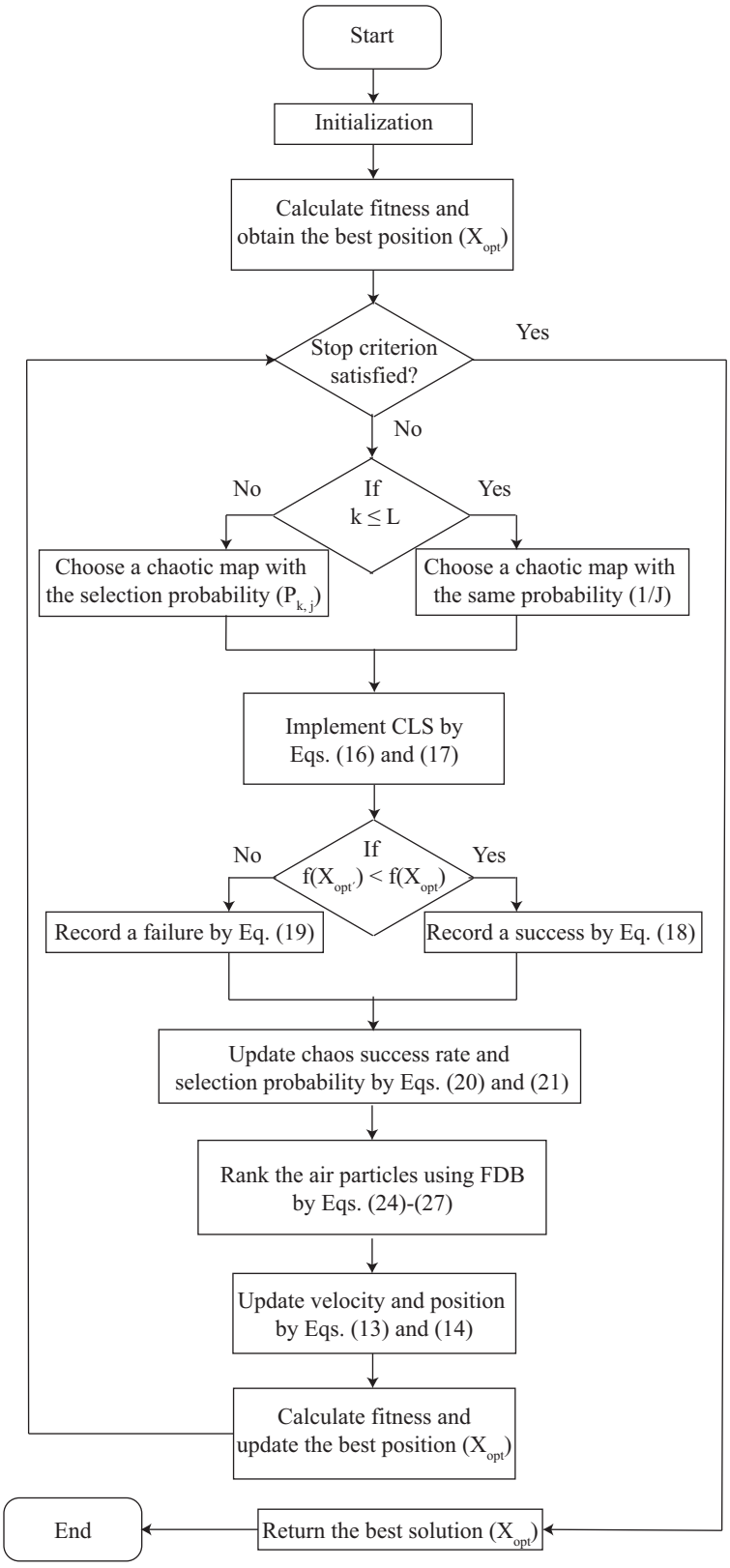

Figure 3.2: Flowchart of CFDBWDO.

Chapter 4

Experiments

4.1 Experimental setting

In this study, IEEE CEC 2017 [80] benchmark functions are taken to test the performance of the proposed algorithm. F1-F3 are three unimodal functions, shown in Fig. 4.1. F4-F10 are seven multimodal functions, shown in Fig. 4.2. F11-F20 are ten hybrid functions, due to they are not defined for 2D, they have no 3D and contour maps. F21-F30 are ten composition functions, shown in Fig. 4.3. It should be mentioned that F2 was excluded because it shows instability in higher dimensions. Thus, 29 IEEE CEC 2017 benchmark functions are used. All functions are minimization optimization problems [81].

The global parameters setting are as follows: Search space range is in $[-100, 100]$. The population size is 100. The maximum number of function evaluations (NFEs) is $10000 * D$, where D is the dimension of functions and set to be 10, 30, 50 or 100. In order to obtain stable experimental data, each algorithm individually runs 51 times on each benchmark function.

The experimental results include mean and standard deviation of 51 independent runs. The mean represents the average performance of algorithm when searching for the optimal solution, the standard deviation indicates the stability in handling optimization problems, the two values are connected by the symbol "±", and the best

mean and standard deviation values among all compared algorithms are highlighted in bold.

The mean and standard deviation of 51 independent runs alone cannot fully explain the superiority of the proposed algorithm, and the statistical test is needed. The statistical results are calculated by a non-parametric statistical method, i.e., Wilcoxon rank-sum test, with the significant level $\alpha = 0.05$ [82–84]. When p-value is less than 0.05, it can be considered to reject the null hypothesis, indicating that there is a significant difference between the two algorithms, where null hypothesis means that there is no significant difference. Symbol "+" represents that the proposed algorithm is superior to its competitor, symbol "−" suggests that the proposed algorithm is worse than its competitor. When p-value is greater than 0.05, it can be considered to accept the null hypothesis, denoting that the two algorithms have the same optimization performance, and it is recorded as symbol "∼". $w/t/l$ presents the number of functions that the proposed algorithm wins, ties and loses to its competitor. All algorithms are executed in MATLAB R2018a and run on a 2.60GHz Intel(R) Core(TM) i7-9750H CPU with 16GB RAM.

4.2 Comparison between SSGSA and competitive algorithms

To evaluate the performance of SSGSA, we compared it with the SS, GSA, PSO and WOA. Their parameter settings are shown in Table 4.1. The tested benchmark functions are the IEEE CEC 2017. The specific experimental conditions are as follows: the population size is 100, dimension of the optimization is 30, maximum iteration is 3000, and the number of independent runs is 30.

Table 4.2 lists the optimization results of the five algorithms on IEEE CEC 2017 benchmark functions, where the "mean" represents the average of the results of each question after 3000 iterations, which represents the ability to find optimization, and

"Std" represents the standard deviation of the results, which represents the robustness of the algorithm when dealing with problems. In the table, the results are best highlighted in bold of the 30 benchmark functions. From it, we can find that SSGSA has the best results of 22, SS has the best results of 5, GSA has three, and PSO and WOA are all have no best result. Obviously, the SSGSA algorithm achieves the best results on most benchmark functions, indicating that SSGSA proposed in this paper has a strong competitive advantage. In the last row of Table 4.2, W/T/L represents the results of SSGSA compared to other algorithms Win/Tied/Loser when the significance level is 0.05. Obviously, SSGSA performs significantly better than other four algorithms.

In order to further explain the superiority of the SSGSA, the experimental results are made into convergence (Figs. 4.5) and box-and-whisker graphs (Fig. 4.4), respectively [85–87]. From the convergence graphs, it can be seen that the convergence ability and speed of SSGSA are better than its peers. It is worth noting that the SSGSA and SS in the convergence graph are almost the same in the first 1500 iterations, and when the iteration number reach 1500, the average optimization value of SSGSA has a process of decreasing, which is due to the incorporation of the gravitational search inherited from GSA, which is replacing the towards-best population update mechanism in SS. It is more intuitively to show that SSGSA has a stronger exploitation ability. From the Box-and-whisker graphs, it can be found that SSGSA can find better results, which suggests that it has a strong ability to find optimization results.

4.3 Analysis for parameters of CFDBWDO

CFDBWDO combines fitness-distance balance and chaotic local search. The weight coefficient w of fitness-distance balance determines the respective weights of fitness

Table 4.1: Parameter settings of algorithms

Algorithm	parameters
SSGSA	$PbestRate = 0.1$, $rd = 0.95$, $c = [0.5\ 0.7]$, $G_0 = 100$, $\alpha = 20$
SS	$PbestRate = 0.1$, $rd = 0.95$, $c = [0.5\ 0.7]$
GSA	$G_0 = 100$, $\alpha = 20$
PSO	$\omega_f = 0.9$, $\omega_f = 0.4$, $c_1 f = 0.5$, $c_1 e = 2.5$, $c_2 f = 2.5$, $c_2 e = 0.5$, $\sigma = 0.2$
WOA	α linearly decreases from 2 to 0

Table 4.2: Experiment results on CEC 2017.

	SSGSA		SS			GSA			PSO			WOA		
	Mean	Std	Mean	Std		Mean	Std		Mean	Std		Mean	Std	
F1	**1.000E+02**	1.370E-10	1.000E+02	4.441E-06	+	1.745E+03	6.830E+02	+	5.065E+10	5.452E+09	+	3.367E+06	2.513E+06	+
F2	**2.000E+02**	2.053E-10	2.000E+02	7.923E-11	−	4.643E+17	1.388E+18	+	1.638E+40	6.905E+40	+	1.136E+20	2.855E+20	+
F3	**3.000E+02**	3.267E-04	3.000E+02	1.100E-04	+	8.449E+04	4.620E+03	+	1.099E+05	1.496E+04	+	1.547E+05	5.728E+04	+
F4	4.594E+02	3.425E+01	**4.508E+02**	3.441E+01	≈	5.377E+02	1.317E+01	+	9.694E+03	1.502E+03	+	5.508E+02	3.194E+01	+
F5	**6.583E+02**	1.021E+01	6.668E+02	1.149E+01	+	7.244E+02	2.000E+01	+	9.254E+02	2.170E+01	+	7.860E+02	6.861E+01	+
F6	**6.000E+02**	7.599E-04	6.000E+02	3.522E-05	≈	6.515E+02	3.406E+00	+	6.856E+02	4.003E+00	+	6.688E+02	1.175E+01	+
F7	8.861E+02	9.916E+00	8.923E+02	1.033E+01	+	**7.846E+02**	1.218E+01	−	2.162E+03	1.502E+02	+	1.243E+03	9.179E+01	+
F8	9.606E+02	1.146E+01	9.681E+02	8.056E+00	+	**9.504E+02**	1.221E+01	−	1.186E+03	2.314E+01	+	1.003E+03	5.129E+01	+
F9	**9.000E+02**	8.640E-02	9.000E+02	8.399E-02	≈	2.993E+03	3.173E+02	+	1.457E+04	1.413E+03	+	7.051E+03	2.493E+03	+
F10	7.883E+03	2.901E+02	8.092E+03	2.218E+02	+	**4.961E+03**	4.876E+02	−	8.255E+03	3.185E+02	+	6.071E+03	7.754E+02	−
F11	**1.141E+03**	2.621E+01	1.159E+03	1.906E+01	+	1.470E+03	9.458E+01	+	7.081E+03	1.071E+03	+	1.441E+03	8.353E+01	+
F12	4.334E+03	2.133E+03	**3.848E+03**	2.619E+03	≈	1.809E+07	3.763E+07	+	5.978E+09	6.379E+08	+	2.919E+07	2.350E+07	+
F13	**1.385E+03**	3.575E+01	1.402E+03	3.082E+01	+	2.863E+04	5.188E+03	+	2.416E+09	5.538E+08	+	1.413E+05	1.066E+05	+
F14	**1.461E+03**	4.670E+00	1.469E+03	5.644E+00	+	5.066E+05	1.119E+05	+	6.046E+05	2.662E+05	+	7.675E+05	9.587E+05	+
F15	**1.525E+03**	6.443E+00	1.542E+03	6.341E+00	+	1.160E+04	1.609E+03	+	1.799E+08	8.672E+07	+	1.016E+05	6.538E+04	+
F16	**2.855E+03**	1.534E+02	2.951E+03	1.500E+02	+	3.091E+03	3.255E+02	+	4.412E+03	2.786E+02	+	3.407E+03	3.850E+02	+
F17	**1.980E+03**	7.002E+01	2.040E+03	7.235E+01	+	2.926E+03	2.017E+02	+	3.018E+03	1.784E+02	+	2.571E+03	3.023E+02	+
F18	**1.829E+03**	2.087E+00	1.838E+03	2.757E+00	+	3.045E+05	1.486E+05	+	8.583E+06	2.806E+06	+	2.485E+06	2.417E+06	+
F19	**1.924E+03**	2.073E+00	1.929E+03	2.178E+00	+	1.182E+04	3.699E+03	+	3.056E+08	1.148E+08	+	2.133E+06	1.557E+06	+
F20	**2.442E+03**	9.126E+01	2.509E+03	7.262E+01	+	3.016E+03	1.736E+02	+	2.797E+03	8.485E+01	+	2.699E+03	2.203E+02	+
F21	**2.452E+03**	1.061E+01	2.463E+03	9.875E+00	+	2.554E+03	2.718E+01	+	2.690E+03	1.897E+01	+	2.554E+03	8.152E+01	+
F22	**2.300E+03**	0.000E+00	2.300E+03	0.000E+00	≈	6.202E+03	2.043E+03	+	7.737E+03	4.874E+02	+	6.315E+03	2.197E+03	+
F23	**2.736E+03**	6.050E+01	2.804E+03	1.061E+01	+	3.575E+03	1.531E+02	+	3.264E+03	4.177E+01	+	3.072E+03	9.842E+01	+
F24	**2.847E+03**	3.436E+01	2.960E+03	3.076E+01	+	3.288E+03	5.028E+01	+	3.463E+03	4.598E+01	+	3.193E+03	9.720E+01	+
F25	**2.887E+03**	2.548E-02	2.887E+03	2.312E-02	≈	2.933E+03	1.171E+01	+	6.438E+03	5.840E+02	+	2.951E+03	3.075E+01	+
F26	**3.503E+03**	3.399E+02	4.219E+03	7.940E+02	+	6.606E+03	1.046E+03	+	9.724E+03	4.602E+02	+	7.425E+03	1.210E+03	+
F27	3.190E+03	8.018E+00	**3.187E+03**	1.132E+01	≈	4.484E+03	3.182E+02	+	3.789E+03	8.610E+01	+	3.350E+03	9.578E+01	+
F28	3.118E+03	4.161E+01	**3.114E+03**	3.571E+01	≈	3.323E+03	5.211E+01	+	6.338E+03	3.743E+02	+	3.301E+03	3.846E+01	+
F29	**3.634E+03**	9.292E+01	3.726E+03	6.664E+01	+	4.740E+03	2.204E+02	+	5.563E+03	2.920E+02	+	4.694E+03	3.719E+02	+
F30	5.222E+03	1.898E+02	**5.158E+03**	8.215E+01	≈	1.518E+05	7.722E+04	+	2.860E+08	8.559E+07	+	1.099E+07	7.981E+06	+
W/T/L	-/-/-		20/9/1			27/0/3			30/0/0			29/0/1		

and distance, which influences the exploration performance of the algorithm. The chaotic search radius r and the learning iteration count L in chaotic local search determine the search range around the current best individual and the best iteration count for choosing chaotic maps, respectively, which influences the exploitation performance of the algorithm.

Thus, w, r and L are three important parameters for CFDBWDO. To find the best parameter setting and achieve the optimal performance of CFDBWDO, the experiments with different parameters w, r and L are implemented on 29 IEEE

Table 4.3: Experimental and statistical results of CFDBWDO with different learning iteration count L on CEC'17 benchmark functions with 30 dimensions.

Parameter	F1	F3	F4	F5	F6	F7
$L = 25$	1.870E+09 ± 1.016E+09 ∼	1.402E+04 ± 8.541E+03 ∼	1.302E+03 ± 5.409E+02 +	6.768E+02 ± 4.635E+01 ∼	6.438E+02 ± 7.205E+00 ∼	**9.748E+02 ± 6.886E+01** ∼
$L = 50$	1.715E+09 ± 1.125E+09	1.297E+04 ± 8.923E+03	**1.216E+03 ± 4.056E+02**	6.647E+02 ± 4.392E+01	6.441E+02 ± 6.551E+00	9.790E+02 ± 6.086E+01
$L = 75$	**1.691E+09 ± 1.111E+09** ∼	**1.185E+04 ± 6.199E+03** ∼	1.273E+03 ± 3.531E+02 +	**6.590E+02 ± 4.275E+01** ∼	**6.434E+02 ± 6.337E+00** ∼	9.944E+02 ± 7.542E+01 ∼
$L = 100$	2.130E+09 ± 8.862E+08 +	1.285E+04 ± 8.206E+03 ∼	1.332E+03 ± 4.601E+02 +	6.653E+02 ± 2.980E+01 ∼	6.442E+02 ± 7.166E+00 ∼	9.775E+02 ± 6.352E+01 ∼
	F8	F9	F10	F11	F12	F13
$L = 25$	**9.243E+02 ± 3.774E+01** ∼	**3.867E+03 ± 1.089E+03** ∼	6.769E+03 ± 1.302E+03 ∼	1.471E+03 ± 5.774E+01 ∼	2.903E+08 ± 1.444E+08 ∼	4.346E+07 ± 4.925E+07 ∼
$L = 50$	9.335E+02 ± 4.150E+01	3.887E+03 ± 1.235E+03	6.877E+03 ± 1.308E+03	1.478E+03 ± 6.164E+01	**2.881E+08 ± 2.274E+08**	**3.673E+07 ± 4.328E+07**
$L = 75$	9.258E+02 ± 3.468E+01 ∼	4.049E+03 ± 1.007E+03 ∼	6.532E+03 ± 1.373E+03 ∼	1.479E+03 ± 6.574E+01 ∼	3.300E+08 ± 2.788E+08 ∼	3.932E+07 ± 5.532E+07 ∼
$L = 100$	9.297E+02 ± 3.456E+01 ∼	3.879E+03 ± 9.596E+02 ∼	**6.507E+03 ± 1.396E+03** ∼	**1.465E+03 ± 6.345E+01** ∼	3.026E+08 ± 1.303E+08 ∼	4.536E+07 ± 5.017E+07 ∼
	F14	F15	F16	F17	F18	F19
$L = 25$	4.832E+04 ± 5.719E+04 ∼	9.076E+04 ± 8.492E+04 ∼	3.387E+03 ± 4.674E+02 ∼	2.160E+03 ± 2.104E+02 ∼	**4.695E+05 ± 3.824E+05** ∼	1.350E+06 ± 2.330E+06 +
$L = 50$	7.501E+04 ± 1.681E+05	1.301E+05 ± 2.232E+05	**3.339E+03 ± 3.396E+02**	2.131E+03 ± 1.777E+02	5.697E+05 ± 6.026E+05	7.812E+05 ± 7.791E+05
$L = 75$	**4.341E+04 ± 5.477E+04** ∼	**8.309E+04 ± 1.033E+05** ∼	3.454E+03 ± 3.896E+02 +	**2.115E+03 ± 1.934E+02** ∼	5.784E+05 ± 5.869E+05 ∼	8.271E+05 ± 6.432E+05 ∼
$L = 100$	5.964E+04 ± 1.051E+05 ∼	2.122E+05 ± 3.292E+05 +	3.352E+03 ± 3.735E+02 ∼	2.194E+03 ± 1.701E+02 +	5.082E+05 ± 3.938E+05 ∼	**6.757E+05 ± 5.505E+05** ∼
	F20	F21	F22	F23	F24	F25
$L = 25$	**2.419E+03 ± 1.352E+02** ∼	2.504E+03 ± 6.021E+01 ∼	2.770E+03 ± 7.845E+01 ∼	3.206E+03 ± 5.223E+01 +	3.394E+03 ± 6.616E+01 ∼	3.020E+03 ± 6.055E+01 ∼
$L = 50$	2.443E+03 ± 1.411E+02	2.500E+03 ± 4.858E+01	2.769E+03 ± 8.820E+01	3.183E+03 ± 4.912E+01	3.400E+03 ± 5.943E+01	3.021E+03 ± 4.653E+01
$L = 75$	2.476E+03 ± 1.599E+02 ∼	2.504E+03 ± 5.558E+01 ∼	**2.752E+03 ± 9.656E+01** ∼	3.205E+03 ± 4.886E+01 +	3.394E+03 ± 6.773E+01 ∼	**3.015E+03 ± 5.481E+01** ∼
$L = 100$	2.488E+03 ± 1.614E+02 ∼	**2.486E+03 ± 6.575E+01** ∼	2.784E+03 ± 8.437E+01 ∼	**3.171E+03 ± 4.903E+01** ∼	**3.391E+03 ± 6.101E+01** ∼	3.020E+03 ± 4.869E+01 ∼
	F26	F27	F28	F29	F30	$w/t/l$
$L = 25$	**5.661E+03 ± 1.869E+03** ∼	**3.769E+03 ± 7.025E+01** ∼	**3.512E+03 ± 4.842E+01** ∼	4.452E+03 ± 3.543E+02 ∼	**8.916E+06 ± 9.185E+06** ∼	3-26-0
$L = 50$	5.932E+03 ± 1.910E+03	3.790E+03 ± 9.414E+01	3.529E+03 ± 7.384E+01	4.547E+03 ± 3.827E+02	1.083E+07 ± 1.608E+07	−−
$L = 75$	6.190E+03 ± 1.826E+03 ∼	3.771E+03 ± 8.158E+01 ∼	3.525E+03 ± 7.117E+01 ∼	4.517E+03 ± 3.581E+02 ∼	9.002E+06 ± 6.646E+06 ∼	3-26-0
$L = 100$	5.980E+03 ± 1.743E+03 ∼	3.772E+03 ± 8.971E+01 ∼	3.557E+03 ± 1.426E+02 ∼	**4.317E+03 ± 2.983E+02** −	9.776E+06 ± 8.343E+06 ∼	4-24-1

Table 4.4: Experimental and statistical results of CFDBWDO with different parameters w and r on CEC'17 benchmark functions with 30 dimensions.

Parameters	F1	F3	F4	F5	F6	F7
$w = 0.5, r = 0.0001$	**1.715E+09 ± 1.125E+09**	1.297E+04 ± 8.923E+03	1.216E+03 ± 4.056E+02	6.647E+02 ± 4.392E+01	6.441E+02 ± 6.551E+00	9.790E+02 ± 6.086E+01
$w = 0.4, r = 0.0001$	1.904E+09 ± 1.051E+09 ∼	1.779E+04 ± 7.432E+03 +	1.213E+03 ± 3.258E+02 ∼	**6.598E+02 ± 3.721E+01** ∼	6.443E+02 ± 7.383E+00 ∼	9.645E+02 ± 6.443E+01 ∼
$w = 0.4, r = 0.0004$	1.866E+09 ± 1.054E+09 ∼	1.787E+04 ± 9.104E+03 +	1.200E+03 ± 2.071E+02 +	6.678E+02 ± 4.740E+01 ∼	6.439E+02 ± 6.534E+00 ∼	9.774E+02 ± 8.003E+01 ∼
$w = 0.4, r = 0.0007$	1.858E+09 ± 1.095E+09 ∼	1.780E+04 ± 7.957E+03 +	**1.187E+03 ± 2.330E+02** ∼	6.685E+02 ± 4.906E+01 ∼	6.427E+02 ± 5.602E+00 ∼	**9.565E+02 ± 5.225E+01** −
$w = 0.5, r = 0.0004$	1.825E+09 ± 1.091E+09 ∼	**1.128E+04 ± 5.613E+03** ∼	1.211E+03 ± 2.731E+02 ∼	6.748E+02 ± 5.393E+01 ∼	**6.403E+02 ± 7.187E+00** −	9.721E+02 ± 6.457E+01 ∼
$w = 0.5, r = 0.0007$	1.788E+09 ± 1.171E+09 ∼	1.259E+04 ± 8.341E+03 ∼	1.214E+03 ± 2.360E+02 +	6.756E+02 ± 3.930E+01 ∼	6.425E+02 ± 8.249E+00 ∼	9.880E+02 ± 8.240E+01 ∼
$w = 0.6, r = 0.0001$	2.058E+09 ± 9.568E+08 ∼	1.585E+04 ± 7.335E+03 +	1.370E+03 ± 5.519E+02 +	6.729E+02 ± 3.972E+01 ∼	6.434E+02 ± 7.265E+00 ∼	9.970E+02 ± 6.743E+01 ∼
$w = 0.6, r = 0.0004$	2.051E+09 ± 8.454E+08 ∼	1.606E+04 ± 7.687E+03 +	1.218E+03 ± 2.615E+02 +	6.699E+02 ± 4.203E+01 ∼	6.437E+02 ± 6.674E+00 ∼	9.924E+02 ± 8.130E+01 ∼
$w = 0.6, r = 0.0007$	1.984E+09 ± 8.890E+08 ∼	1.436E+04 ± 6.719E+03 ∼	1.213E+03 ± 2.427E+02 +	6.790E+02 ± 4.041E+01 +	6.424E+02 ± 7.128E+00 ∼	9.798E+02 ± 7.060E+01 ∼
	F8	F9	F10	F11	F12	F13
$w = 0.5, r = 0.0001$	9.335E+02 ± 4.150E+01	3.887E+03 ± 1.235E+03	6.877E+03 ± 1.308E+03	1.478E+03 ± 6.164E+01	2.881E+08 ± 2.274E+08	3.673E+07 ± 4.328E+07
$w = 0.4, r = 0.0001$	9.327E+02 ± 4.726E+01 ∼	3.986E+03 ± 1.480E+03 ∼	5.955E+03 ± 1.462E+03 −	1.472E+03 ± 7.606E+01 ∼	**2.395E+08 ± 8.280E+07** −	5.931E+07 ± 5.244E+07 +
$w = 0.4, r = 0.0004$	**9.149E+02 ± 2.865E+01** −	4.020E+03 ± 1.255E+03 ∼	5.864E+03 ± 1.421E+03 −	1.480E+03 ± 6.822E+01 ∼	2.562E+08 ± 8.645E+07 ∼	4.746E+07 ± 5.326E+07 +
$w = 0.4, r = 0.0007$	9.222E+02 ± 3.187E+01 ∼	3.594E+03 ± 8.654E+02 ∼	**5.629E+03 ± 1.423E+03** −	1.469E+03 ± 6.293E+01 ∼	2.900E+08 ± 8.997E+07 ∼	4.397E+07 ± 3.809E+07 +
$w = 0.5, r = 0.0004$	9.364E+02 ± 3.923E+01 ∼	3.744E+03 ± 8.245E+02 ∼	6.507E+03 ± 1.366E+03 ∼	1.468E+03 ± 5.369E+01 ∼	3.645E+08 ± 4.435E+08 ∼	3.545E+07 ± 4.664E+07 ∼
$w = 0.5, r = 0.0007$	9.223E+02 ± 3.256E+01 ∼	3.779E+03 ± 1.019E+03 ∼	6.545E+03 ± 1.480E+03 ∼	1.470E+03 ± 5.327E+01 ∼	2.865E+08 ± 8.179E+07 +	5.994E+07 ± 8.930E+07 ∼
$w = 0.6, r = 0.0001$	9.364E+02 ± 3.938E+01 ∼	3.516E+03 ± 1.137E+03 −	6.859E+03 ± 1.390E+03 ∼	**1.455E+03 ± 5.615E+01** −	2.612E+08 ± 2.096E+08 ∼	4.509E+07 ± 4.504E+07 +
$w = 0.6, r = 0.0004$	9.406E+02 ± 4.131E+01 ∼	**3.439E+03 ± 9.017E+02** −	7.258E+03 ± 1.195E+03 ∼	1.478E+03 ± 5.373E+01 ∼	3.363E+08 ± 4.134E+08 ∼	6.537E+07 ± 9.956E+07 +
$w = 0.6, r = 0.0007$	9.282E+02 ± 2.822E+01 ∼	3.619E+03 ± 9.571E+02 ∼	7.002E+03 ± 1.338E+03 ∼	1.466E+03 ± 7.010E+01 ∼	2.755E+08 ± 1.222E+08 ∼	**3.148E+07 ± 4.194E+07** ∼
	F14	F15	F16	F17	F18	F19
$w = 0.5, r = 0.0001$	7.501E+04 ± 1.681E+05	1.301E+05 ± 2.232E+05	3.339E+03 ± 3.396E+02	2.131E+03 ± 1.777E+02	5.697E+05 ± 6.026E+05	**7.812E+05 ± 7.791E+05**
$w = 0.4, r = 0.0001$	4.417E+04 ± 4.896E+04 ∼	2.162E+05 ± 2.126E+05 +	3.332E+03 ± 3.578E+02 ∼	**2.114E+03 ± 1.560E+02** ∼	7.041E+05 ± 9.652E+05 ∼	8.564E+05 ± 6.457E+05 ∼
$w = 0.4, r = 0.0004$	4.805E+04 ± 4.839E+04 ∼	6.474E+05 ± 2.198E+06 +	3.322E+03 ± 3.642E+02 ∼	2.184E+03 ± 1.537E+02 +	6.959E+05 ± 7.586E+05 ∼	1.112E+06 ± 1.265E+06 +
$w = 0.4, r = 0.0007$	**3.850E+04 ± 3.689E+04** ∼	2.548E+05 ± 2.884E+05 +	3.339E+03 ± 3.660E+02 ∼	2.168E+03 ± 1.930E+02 ∼	5.175E+05 ± 4.445E+05 ∼	9.323E+05 ± 8.684E+05 ∼
$w = 0.5, r = 0.0004$	5.110E+04 ± 5.033E+04 ∼	1.355E+05 ± 1.955E+05 ∼	3.385E+03 ± 4.693E+02 ∼	2.150E+03 ± 2.005E+02 ∼	**4.605E+05 ± 3.260E+05** ∼	7.882E+05 ± 6.982E+05 ∼
$w = 0.5, r = 0.0007$	3.893E+04 ± 4.083E+04 ∼	1.378E+05 ± 2.170E+05 ∼	**3.280E+03 ± 3.564E+02** ∼	2.145E+03 ± 2.101E+02 ∼	7.100E+05 ± 9.799E+05 ∼	1.113E+06 ± 2.243E+06 ∼
$w = 0.6, r = 0.0001$	3.890E+04 ± 5.239E+04 ∼	1.510E+05 ± 2.088E+05 +	3.415E+03 ± 4.697E+02 ∼	2.160E+03 ± 1.971E+02 ∼	6.123E+05 ± 7.282E+05 ∼	1.285E+06 ± 2.081E+06 +
$w = 0.6, r = 0.0004$	5.096E+04 ± 8.956E+04 ∼	2.026E+05 ± 2.960E+05 +	3.543E+03 ± 4.402E+02 +	2.131E+03 ± 1.945E+02 ∼	7.623E+05 ± 8.087E+05 ∼	8.922E+05 ± 6.807E+05 ∼
$w = 0.6, r = 0.0007$	7.012E+04 ± 8.090E+04 ∼	**1.261E+05 ± 1.625E+05** +	3.431E+03 ± 4.657E+02 ∼	2.142E+03 ± 2.125E+02 ∼	5.518E+05 ± 4.822E+05 ∼	1.083E+06 ± 1.016E+06 +
	F20	F21	F22	F23	F24	F25
$w = 0.5, r = 0.0001$	2.443E+03 ± 1.411E+02	2.500E+03 ± 4.858E+01	2.769E+03 ± 8.820E+01	3.183E+03 ± 4.912E+01	3.400E+03 ± 5.943E+01	3.021E+03 ± 4.653E+01
$w = 0.4, r = 0.0001$	**2.412E+03 ± 1.251E+02** ∼	2.508E+03 ± 5.436E+01 ∼	2.796E+03 ± 9.723E+01 +	3.182E+03 ± 6.045E+01 ∼	**3.379E+03 ± 6.933E+01** ∼	3.033E+03 ± 3.688E+01 ∼
$w = 0.4, r = 0.0004$	2.444E+03 ± 1.422E+02 ∼	2.512E+03 ± 4.983E+01 ∼	2.764E+03 ± 8.862E+01 ∼	3.187E+03 ± 5.820E+01 ∼	3.391E+03 ± 7.615E+01 ∼	3.037E+03 ± 3.924E+01 ∼
$w = 0.4, r = 0.0007$	2.444E+03 ± 1.403E+02 ∼	2.517E+03 ± 5.273E+01 +	2.855E+03 ± 7.371E+02 ∼	**3.177E+03 ± 4.694E+01** ∼	3.391E+03 ± 6.922E+01 ∼	3.033E+03 ± 3.875E+01 ∼
$w = 0.5, r = 0.0004$	2.457E+03 ± 1.366E+02 ∼	2.521E+03 ± 4.843E+01 +	2.846E+03 ± 5.290E+02 ∼	3.191E+03 ± 4.840E+01 ∼	3.400E+03 ± 6.891E+01 ∼	3.027E+03 ± 4.867E+01 ∼
$w = 0.5, r = 0.0007$	2.429E+03 ± 1.325E+02 ∼	**2.497E+03 ± 5.317E+01** ∼	2.885E+03 ± 8.448E+02 ∼	3.191E+03 ± 7.042E+01 ∼	3.393E+03 ± 7.179E+01 ∼	**3.011E+03 ± 5.347E+01** ∼
$w = 0.6, r = 0.0001$	2.484E+03 ± 1.754E+02 ∼	2.508E+03 ± 5.550E+01 ∼	2.777E+03 ± 1.051E+02 ∼	3.192E+03 ± 4.817E+01 ∼	3.407E+03 ± 6.995E+01 ∼	3.023E+03 ± 3.982E+01 ∼
$w = 0.6, r = 0.0004$	2.461E+03 ± 1.401E+02 ∼	2.515E+03 ± 5.569E+01 ∼	**2.749E+03 ± 9.141E+01** ∼	3.187E+03 ± 4.895E+01 ∼	3.396E+03 ± 5.521E+01 ∼	3.032E+03 ± 3.714E+01 ∼
$w = 0.6, r = 0.0007$	2.459E+03 ± 1.596E+02 ∼	2.515E+03 ± 5.547E+01 ∼	2.766E+03 ± 8.862E+01 ∼	3.196E+03 ± 5.421E+01 ∼	3.397E+03 ± 7.946E+01 ∼	3.033E+03 ± 4.205E+01 ∼
	F26	F27	F28	F29	F30	$w/t/l$
$w = 0.5, r = 0.0001$	**5.932E+03 ± 1.910E+03**	3.790E+03 ± 9.414E+01	3.529E+03 ± 7.384E+01	4.547E+03 ± 3.827E+02	1.083E+07 ± 1.608E+07	−−
$w = 0.4, r = 0.0001$	6.084E+03 ± 1.714E+03 ∼	3.757E+03 ± 9.472E+01 −	3.524E+03 ± 6.119E+01 ∼	4.471E+03 ± 3.305E+02 ∼	1.005E+07 ± 6.792E+06 +	5-21-3
$w = 0.4, r = 0.0004$	6.044E+03 ± 1.779E+03 ∼	**3.755E+03 ± 8.696E+01** −	3.516E+03 ± 6.711E+01 ∼	4.481E+03 ± 3.622E+02 ∼	1.015E+07 ± 8.550E+06 +	7-19-3
$w = 0.4, r = 0.0007$	6.160E+03 ± 1.801E+03 ∼	3.779E+03 ± 7.685E+01 ∼	**3.514E+03 ± 5.798E+01** ∼	4.487E+03 ± 2.799E+02 ∼	1.238E+07 ± 8.981E+06 +	5-22-2
$w = 0.5, r = 0.0004$	6.667E+03 ± 1.695E+03 +	3.764E+03 ± 8.195E+01 ∼	3.540E+03 ± 1.050E+02 ∼	4.465E+03 ± 3.459E+02 ∼	8.397E+06 ± 7.659E+06 ∼	2-26-1
$w = 0.5, r = 0.0007$	6.536E+03 ± 1.833E+03 ∼	3.789E+03 ± 8.057E+01 ∼	3.524E+03 ± 6.563E+01 ∼	**4.411E+03 ± 3.408E+02** ∼	1.048E+07 ± 7.484E+06 +	3-26-0
$w = 0.6, r = 0.0001$	6.196E+03 ± 1.740E+03 ∼	3.783E+03 ± 7.878E+01 ∼	3.530E+03 ± 7.561E+01 ∼	4.508E+03 ± 3.325E+02 ∼	9.078E+06 ± 6.547E+06 ∼	5-22-2
$w = 0.6, r = 0.0004$	6.372E+03 ± 1.959E+03 +	3.770E+03 ± 7.833E+01 ∼	3.531E+03 ± 7.251E+01 ∼	4.499E+03 ± 3.825E+02 ∼	**7.560E+06 ± 4.706E+06** ∼	6-22-1
$w = 0.6, r = 0.0007$	6.125E+03 ± 1.935E+03 ∼	3.776E+03 ± 6.675E+01 ∼	3.532E+03 ± 7.944E+01 ∼	4.478E+03 ± 3.478E+02 ∼	1.084E+07 ± 7.960E+06 ∼	4-25-0

CEC 2017 benchmark functions with 30 dimensions. Four values are set for L, i.e., $L \in \{25, 50, 75, 100\}$. The experimental and statistical results are shown in Table 4.3. Three values are set for w and r respectively, i.e., $w \in \{0.4, 0.5, 0.6\}$ and

Table 4.5: Parameter settings of four WDO variants.

Algorithm	Parameters
WDO	$\alpha = 0.4$, $g = 0.2$, $RT = 3$, $c = 0.4$, $u_{\max} = 0.1$
AWDO	CMAES updates the α, g, RT, c, $u_{\max} = 0.3$
FDBWDO	$\alpha = 0.4$, $g = 0.2$, $RT = 3$, $c = 0.4$, $w = 0.5$
CFDBWDO	$\alpha = 0.4$, $g = 0.2$, $RT = 3$, $c = 0.4$, $w = 0.5$, $r = 0.0001$, $L = 50$, $\varepsilon = 0.01$, $J = 12$

$r \in \{0.0001, 0.0004, 0.0007\}$. Nine combinations are tested on functions. The results are displayed in Table 4.4. From these tables, we can find that CFDBWDO with $w = 0.5$, $r = 0.0001$ and $L = 50$ performs the best.

4.4 Comparison for WDO variants

In order to examine the superiority of CFDBWDO, we compare it with other representative WDO variants, including the original WDO [49], AWDO [58] and FDBWDO. These algorithms are tested on 29 IEEE CEC 2017 benchmark functions with different dimensions. It is notable that AWDO adopted covariance matrix adaptive evolutionary strategy (CMAES) to update the parameters, while FDBWDO only introduced fitness distance balance strategy. The experimental parameter settings of four algorithms are presented in Table 4.5. Note that since there is no velocity limitation in CFDBWDO and FDBWDO, the parameter $\boldsymbol{u_{\max}}$ is removed. The experimental and statistical results are given in Tables 4.8, 4.9, 4.10 and 4.11.

In Table 4.8, all 29 IEEE CEC 2017 benchmark functions with 10 dimensions are used. CFDBWDO attains the best mean on 13 benchmark functions. $w/t/l$ illustrates that CFDBWDO performs significantly better than FDBWDO, AWDO and WDO on 4, 21 and 13 benchmark functions, respectively. It manifests that CFDBWDO has excellent performance on benchmark functions with low dimensions. In Table 4.9, all benchmark functions with 30 dimensions are adopted. CFDBWDO gains the best mean on 10 benchmark functions. Through $w/t/l$, the number of wins of CFDB-

Table 4.6: Friedman test ranking of four WDO variants on IEEE CEC 2017.

Algorithms	Dimension=10	Dimension=30	Dimension=50	Dimension=100	Mean Rank
CFDBWDO	**1.7586**	1.8621	**1.6207**	**1.7931**	**1.7586**
FDBWDO	2	**1.7931**	1.7931	2	1.8966
AWDO	3.6552	3.2069	3.2759	2.8276	3.2414
WDO	2.5862	3.1379	3.3103	3.3793	3.1034

Table 4.7: Wilcoxon test comparison results for CFDBWDO and WDO variants on IEEE CEC 2017.

Dimension=10		Dimension=30		Dimension=50		Dimension=100	
Algorithm	$w/t/l$	Algorithm	$w/t/l$	Algorithm	$w/t/l$	Algorithm	$w/t/l$
CFDBWDO	-/-/-	CFDBWDO	-/-/-	CFDBWDO	-/-/-	CFDBWDO	-/-/-
FDBWDO	4/24/1	FDBWDO	3/26/0	FDBWDO	3/26/0	FDBWDO	4/23/2
AWDO	21/3/5	AWDO	16/3/10	AWDO	14/3/12	AWDO	13/2/14
WDO	13/15/1	WDO	15/13/1	WDO	24/5/0	WDO	22/7/0

WDO in comparison with FDBWDO, AWDO and WDO is 3, 16 and 15, respectively. It suggests that CFDBWDO has superiority on benchmark functions with medium dimensions. In Table 4.10, all functions with 50 dimensions are used. CFDBWDO acquires the best mean on 12 benchmark functions. In the light of $w/t/l$, CFDBWDO significantly surpasses three variants of WDO on 3, 14 and 24 benchmark functions, respectively. The result indicates that CFDBWDO performs excellent performance on benchmark functions with high dimensions. In Table 4.11, all functions with 100 dimensions are tested. CFDBWDO gets the best mean on 10 benchmark functions. Statistical results demonstrate that CFDBWDO significantly outperforms the others on 4, 13 and 22 benchmark functions, respectively. It proves that CFDBWDO is able to optimize functions with large dimensions. In addition, when optimizing these functions with large dimensions, CFDBWDO has similar results compared to the AWDO.

From Tables 4.8, 4.9, 4.10 and 4.11, it can be summarized below: (1) The comparison between CFDBWDO and WDO denotes that the incorporation of chaotic local search and fitness distance balance strategy significantly improved WDO. (2) The comparison between CFDBWDO and FDBWDO suggests that chaotic local search

Table 4.8: Experimental and statistical results of four WDO variants on CEC'17 benchmark functions with 10 dimensions.

Algorithm	F1	F3	F4	F5	F6	F7
CFDBWDO	2.908E+08 ± 3.218E+08	**3.961E+02 ± 1.344E+02**	**4.266E+02 ± 1.322E+01**	5.394E+02 ± 1.370E+01	**6.099E+02 ± 3.264E+00**	7.406E+02 ± 1.305E+01
FDBWDO	**2.459E+08 ± 2.421E+08** ∼	4.232E+02 ± 1.654E+02 ∼	4.330E+02 ± 1.871E+01 +	**5.384E+02 ± 1.267E+01** ∼	6.101E+02 ± 3.454E+00 ∼	**7.390E+02 ± 1.346E+01** ∼
AWDO	3.236E+08 ± 7.511E+08 −	2.794E+03 ± 3.883E+03 ∼	4.375E+02 ± 6.817E+01 −	5.504E+02 ± 1.931E+01 +	6.197E+02 ± 1.103E+01 +	7.638E+02 ± 2.663E+01 +
WDO	5.991E+08 ± 5.133E+08 +	2.888E+04 ± 7.217E+03 +	4.433E+02 ± 2.193E+01 +	5.414E+02 ± 1.407E+01 ∼	6.122E+02 ± 4.450E+00 +	7.419E+02 ± 1.247E+01 ∼
	F8	F9	F10	F11	F12	F13
CFDBWDO	8.189E+02 ± 7.832E+00	**9.331E+02 ± 3.330E+01**	1.982E+03 ± 3.274E+02	1.127E+03 ± 2.098E+01	3.617E+06 ± 5.576E+06	**1.443E+04 ± 1.350E+04**
FDBWDO	**8.183E+02 ± 5.619E+00** ∼	9.403E+02 ± 4.464E+01 ∼	**1.969E+03 ± 3.019E+02** ∼	**1.123E+03 ± 1.234E+01** ∼	**2.241E+06 ± 3.211E+06** ∼	2.184E+04 ± 2.122E+04 +
AWDO	8.385E+02 ± 1.753E+01 +	1.149E+03 ± 2.304E+02 +	2.342E+03 ± 3.245E+02 +	1.256E+03 ± 2.023E+02 +	1.129E+07 ± 2.258E+07 ∼	1.756E+05 ± 2.619E+05 +
WDO	8.213E+02 ± 7.609E+00 +	9.346E+02 ± 4.636E+01 ∼	2.123E+03 ± 3.546E+02 +	1.124E+03 ± 1.317E+01 ∼	6.295E+06 ± 1.468E+07 ∼	1.724E+04 ± 1.206E+04 +
	F14	F15	F16	F17	F18	F19
CFDBWDO	**1.551E+03 ± 1.297E+02**	**1.889E+03 ± 2.802E+02**	1.697E+03 ± 7.107E+01	1.752E+03 ± 1.855E+01	9.823E+04 ± 1.337E+05	2.789E+03 ± 1.313E+03
FDBWDO	1.613E+03 ± 2.807E+02 ∼	1.959E+03 ± 4.058E+02 ∼	1.705E+03 ± 7.320E+01 ∼	1.759E+03 ± 2.007E+01 +	**6.902E+04 ± 7.118E+04** ∼	2.740E+03 ± 1.569E+03 ∼
AWDO	1.835E+03 ± 4.463E+02 +	4.425E+03 ± 2.307E+03 +	1.840E+03 ± 1.198E+02 +	1.795E+03 ± 3.726E+01 +	8.558E+05 ± 1.227E+06 +	8.182E+03 ± 7.021E+03 +
WDO	1.596E+03 ± 2.374E+02 +	2.248E+03 ± 5.939E+02 +	**1.688E+03 ± 6.070E+01** ∼	**1.752E+03 ± 1.328E+01** ∼	1.098E+05 ± 1.475E+05 ∼	**2.382E+03 ± 6.249E+02** ∼
	F20	F21	F22	F23	F24	F25
CFDBWDO	**2.070E+03 ± 2.446E+01**	2.206E+03 ± 4.456E+00	**2.306E+03 ± 2.080E+01**	**2.635E+03 ± 1.148E+02**	2.514E+03 ± 2.471E+01	**2.922E+03 ± 2.005E+01**
FDBWDO	2.079E+03 ± 2.770E+01 +	**2.205E+03 ± 3.118E+00** ∼	2.310E+03 ± 2.096E+01 ∼	2.656E+03 ± 8.513E+01 ∼	2.519E+03 ± 5.770E+01 ∼	2.928E+03 ± 2.155E+01 ∼
AWDO	2.121E+03 ± 4.644E+01 +	2.226E+03 ± 3.050E+01 +	2.318E+03 ± 4.185E+01 ∼	2.655E+03 ± 3.003E+01 −	2.647E+03 ± 1.109E+02 +	2.960E+03 ± 8.386E+01 +
WDO	2.079E+03 ± 2.774E+01 +	2.209E+03 ± 6.303E+00 +	2.309E+03 ± 1.760E+01 ∼	2.663E+03 ± 7.336E+01 ∼	**2.512E+03 ± 2.598E+01** ∼	2.928E+03 ± 2.847E+01 ∼
	F26	F27	F28	F29	F30	$w/t/l$
CFDBWDO	3.014E+03 ± 7.525E+01	3.141E+03 ± 9.149E+00	**3.205E+03 ± 7.902E+01**	3.237E+03 ± 3.703E+01	**7.919E+05 ± 8.675E+05**	−−
FDBWDO	**2.995E+03 ± 7.616E+01** ∼	3.137E+03 ± 9.069E+00 −	3.210E+03 ± 7.967E+01 ∼	**3.228E+03 ± 2.802E+01** ∼	9.633E+05 ± 1.066E+06 ∼	4/24/1
AWDO	3.011E+03 ± 1.722E+02 −	**3.126E+03 ± 2.194E+01** −	3.258E+03 ± 1.242E+02 +	3.294E+03 ± 6.209E+01 +	2.380E+06 ± 2.314E+06 +	21/3/5
WDO	3.061E+03 ± 1.168E+02 +	3.134E+03 ± 7.297E+00 −	3.245E+03 ± 8.388E+01 +	3.230E+03 ± 3.680E+01 ∼	1.322E+06 ± 1.695E+06 ∼	13/15/1

Table 4.9: Experimental and statistical results of four WDO variants on CEC'17 benchmark functions with 30 dimensions.

Algorithm	F1	F3	F4	F5	F6	F7
CFDBWDO	1.715E+09 ± 1.125E+09	1.297E+04 ± 8.923E+03	1.216E+03 ± 4.056E+02	**6.647E+02 ± 4.392E+01**	6.441E+02 ± 6.551E+00	**9.790E+02 ± 6.086E+01**
FDBWDO	1.975E+09 ± 9.744E+08 ∼	1.370E+04 ± 9.244E+03 ∼	1.158E+03 ± 2.107E+02 ∼	6.843E+02 ± 5.479E+01 +	**6.429E+02 ± 7.227E+00** ∼	9.800E+02 ± 7.148E+01 ∼
AWDO	**1.462E+09 ± 7.671E+09** −	5.856E+04 ± 3.786E+04 +	**1.018E+03 ± 1.342E+03** −	7.862E+02 ± 8.823E+01 +	6.634E+02 ± 1.764E+01 +	1.181E+03 ± 1.525E+02 +
WDO	2.394E+09 ± 2.679E+09 ∼	**1.196E+04 ± 6.232E+03** ∼	2.206E+03 ± 7.318E+02 +	6.975E+02 ± 6.813E+01 +	6.480E+02 ± 7.280E+00 +	1.024E+03 ± 7.539E+01 +
	F8	F9	F10	F11	F12	F13
CFDBWDO	9.335E+02 ± 4.150E+01	3.887E+03 ± 1.235E+03	6.877E+03 ± 1.308E+03	1.478E+03 ± 6.164E+01	2.881E+08 ± 2.274E+08	**3.673E+07 ± 4.328E+07**
FDBWDO	**9.300E+02 ± 3.537E+01** ∼	**3.485E+03 ± 8.605E+02** ∼	**6.513E+03 ± 1.511E+03** ∼	**1.472E+03 ± 5.202E+01** ∼	**2.738E+08 ± 1.443E+08** ∼	5.875E+07 ± 8.729E+07 ∼
AWDO	1.013E+03 ± 8.869E+01 +	8.655E+03 ± 3.281E+03 +	7.321E+03 ± 1.655E+03 +	3.228E+03 ± 2.426E+03 ∼	3.675E+08 ± 9.400E+08 −	4.914E+08 ± 1.007E+09 −
WDO	9.395E+02 ± 3.294E+01 ∼	4.004E+03 ± 1.167E+03 ∼	7.366E+03 ± 1.256E+03 ∼	1.523E+03 ± 1.353E+02 +	9.170E+08 ± 6.555E+08 +	2.175E+08 ± 2.350E+08 +
	F14	F15	F16	F17	F18	F19
CFDBWDO	7.501E+04 ± 1.681E+05	**1.301E+05 ± 2.232E+05**	**3.339E+03 ± 3.396E+02**	**2.131E+03 ± 1.777E+02**	**5.697E+05 ± 6.026E+05**	**7.812E+05 ± 7.791E+05**
FDBWDO	**5.310E+04 ± 5.647E+04** ∼	3.225E+05 ± 1.595E+06 ∼	3.354E+03 ± 4.098E+02 ∼	2.134E+03 ± 1.802E+02 ∼	6.575E+05 ± 8.319E+05 ∼	1.597E+06 ± 3.738E+06 +
AWDO	6.005E+05 ± 6.454E+05 +	2.385E+07 ± 5.110E+07 +	3.708E+03 ± 8.925E+02 +	2.593E+03 ± 3.991E+02 +	6.745E+06 ± 9.174E+06 +	3.485E+07 ± 7.881E+07 +
WDO	6.505E+04 ± 5.034E+04 ∼	3.157E+06 ± 1.244E+07 ∼	3.763E+03 ± 3.618E+02 +	2.227E+03 ± 2.670E+02 ∼	1.433E+06 ± 1.821E+06 +	1.983E+06 ± 5.658E+06 ∼
	F20	F21	F22	F23	F24	F25
CFDBWDO	2.443E+03 ± 1.411E+02	**2.500E+03 ± 4.858E+01**	2.769E+03 ± 8.820E+01	3.183E+03 ± 4.912E+01	3.400E+03 ± 5.943E+01	3.021E+03 ± 4.653E+01
FDBWDO	**2.407E+03 ± 1.392E+02** ∼	2.504E+03 ± 5.301E+01 ∼	2.890E+03 ± 8.072E+02 ∼	3.200E+03 ± 4.047E+01 +	3.388E+03 ± 6.820E+01 ∼	**3.020E+03 ± 4.911E+01** ∼
AWDO	2.768E+03 ± 1.893E+02 +	2.557E+03 ± 1.045E+02 +	**2.740E+03 ± 1.244E+03** −	**3.123E+03 ± 1.483E+02** −	**3.263E+03 ± 1.680E+02** −	3.185E+03 ± 5.473E+02 −
WDO	2.483E+03 ± 1.605E+02 ∼	2.534E+03 ± 6.281E+01 +	3.235E+03 ± 2.946E+02 +	3.194E+03 ± 4.012E+01 ∼	3.398E+03 ± 6.018E+01 ∼	3.060E+03 ± 8.635E+01 +
	F26	F27	F28	F29	F30	$w/t/l$
CFDBWDO	5.932E+03 ± 1.910E+03	3.790E+03 ± 9.414E+01	3.529E+03 ± 7.384E+01	4.547E+03 ± 3.827E+02	**1.083E+07 ± 1.608E+07**	−−
FDBWDO	**5.651E+03 ± 1.795E+03** ∼	3.766E+03 ± 7.475E+01 ∼	**3.529E+03 ± 7.753E+01** ∼	**4.479E+03 ± 3.400E+02** ∼	1.154E+07 ± 1.399E+07 ∼	3/26/0
AWDO	6.035E+03 ± 2.356E+03 ∼	**3.609E+03 ± 1.987E+02** −	3.555E+03 ± 7.936E+02 −	5.037E+03 ± 6.002E+02 +	5.600E+07 ± 8.321E+07 ∼	16/3/10
WDO	6.263E+03 ± 1.991E+03 ∼	3.760E+03 ± 7.255E+01 −	3.857E+03 ± 1.893E+02 +	4.777E+03 ± 3.817E+02 +	2.400E+07 ± 2.789E+07 +	15/13/1

Table 4.10: Experimental and statistical results of four WDO variants on CEC'17 benchmark functions with 50 dimensions.

Algorithm	F1	F3	F4	F5	F6	F7
CFDBWDO	**6.318E+09 ± 3.764E+09**	**4.406E+04 ± 1.538E+04**	2.830E+03 ± 5.282E+02	7.881E+02 ± 6.653E+01	6.547E+02 ± 6.209E+00	1.392E+03 ± 1.541E+02
FDBWDO	6.324E+09 ± 3.898E+09 ∼	4.419E+04 ± 1.411E+04 ∼	2.866E+03 ± 5.922E+02 ∼	**7.854E+02 ± 4.881E+01** ∼	**6.544E+02 ± 8.058E+00** ∼	**1.355E+03 ± 1.436E+02** ∼
AWDO	7.360E+09 ± 2.171E+10 −	2.044E+05 ± 4.305E+04 +	**1.928E+03 ± 4.977E+03** −	9.251E+02 ± 1.586E+02 +	6.807E+02 ± 1.928E+01 +	1.617E+03 ± 1.878E+02 +
WDO	1.190E+10 ± 8.037E+09 +	5.619E+04 ± 1.693E+04 +	5.105E+03 ± 1.491E+03 +	8.246E+02 ± 6.244E+01 +	6.607E+02 ± 9.067E+00 +	1.492E+03 ± 1.520E+02 +
	F8	F9	F10	F11	F12	F13
CFDBWDO	1.110E+03 ± 5.800E+01	**1.352E+04 ± 3.996E+03**	1.101E+04 ± 2.781E+03	**2.573E+03 ± 4.428E+02**	**3.170E+09 ± 1.417E+09**	**3.014E+08 ± 4.837E+08**
FDBWDO	**1.105E+03 ± 6.979E+01** ∼	1.381E+04 ± 2.392E+03 ∼	**1.034E+04 ± 2.626E+03** ∼	2.629E+03 ± 5.001E+02 ∼	3.178E+09 ± 1.200E+09 ∼	3.646E+08 ± 3.667E+08 +
AWDO	1.301E+03 ± 1.882E+02 +	3.217E+04 ± 1.240E+04 +	1.273E+04 ± 3.014E+03 +	7.983E+03 ± 8.539E+03 ∼	4.109E+09 ± 1.312E+10 −	1.521E+09 ± 3.723E+09 −
WDO	1.136E+03 ± 6.312E+01 +	1.779E+04 ± 5.182E+03 +	1.262E+04 ± 2.484E+03 +	3.254E+03 ± 1.252E+03 +	9.561E+09 ± 3.841E+09 +	2.123E+09 ± 1.617E+09 +
	F14	F15	F16	F17	F18	F19
CFDBWDO	1.700E+06 ± 1.988E+06	**2.285E+07 ± 2.365E+07**	**4.464E+03 ± 6.756E+02**	3.084E+03 ± 3.346E+02	3.255E+06 ± 1.829E+06	**5.113E+06 ± 8.055E+06**
FDBWDO	**1.459E+06 ± 1.122E+06** ∼	3.413E+07 ± 2.842E+07 +	4.602E+03 ± 7.289E+02 ∼	**3.021E+03 ± 3.018E+02** ∼	**3.026E+06 ± 1.294E+06** ∼	5.701E+06 ± 1.206E+07 ∼
AWDO	6.641E+06 ± 7.101E+06 +	3.622E+08 ± 6.853E+08 −	5.291E+03 ± 1.479E+03 +	4.107E+03 ± 6.650E+02 +	2.837E+07 ± 3.638E+07 +	1.635E+08 ± 3.962E+08 −
WDO	2.251E+06 ± 1.518E+06 +	4.629E+07 ± 8.272E+07 +	5.189E+03 ± 9.958E+02 +	3.225E+03 ± 4.434E+02 ∼	6.837E+06 ± 6.715E+06 +	8.407E+06 ± 1.262E+07 +
	F20	F21	F22	F23	F24	F25
CFDBWDO	3.185E+03 ± 4.011E+02	**2.726E+03 ± 1.136E+02**	1.205E+04 ± 4.044E+03	3.920E+03 ± 9.225E+01	4.219E+03 ± 1.109E+02	4.257E+03 ± 1.527E+02
FDBWDO	**3.184E+03 ± 3.449E+02** ∼	2.732E+03 ± 1.052E+02 ∼	**1.192E+04 ± 3.976E+03** ∼	3.933E+03 ± 9.639E+01 ∼	4.216E+03 ± 1.183E+02 ∼	4.266E+03 ± 1.723E+02 ∼
AWDO	3.797E+03 ± 4.564E+02 +	2.784E+03 ± 1.790E+02 ∼	1.475E+04 ± 3.307E+03 +	**3.763E+03 ± 2.659E+02** −	**3.948E+03 ± 2.956E+02** −	**3.965E+03 ± 2.862E+03** −
WDO	3.343E+03 ± 4.386E+02 +	2.773E+03 ± 1.455E+02 +	1.252E+04 ± 3.420E+03 ∼	3.938E+03 ± 9.726E+01 ∼	4.249E+03 ± 9.898E+01 ∼	4.847E+03 ± 5.174E+02 +
	F26	F27	F28	F29	F30	$w/t/l$
CFDBWDO	**8.310E+03 ± 3.004E+03**	5.388E+03 ± 2.453E+02	4.869E+03 ± 1.662E+02	6.567E+03 ± 5.359E+02	**1.951E+08 ± 7.061E+07**	−−
FDBWDO	8.401E+03 ± 2.822E+03 ∼	5.408E+03 ± 2.129E+02 ∼	4.913E+03 ± 1.542E+02 +	**6.495E+03 ± 5.219E+02** ∼	1.952E+08 ± 7.404E+07 ∼	3/26/0
AWDO	1.025E+04 ± 4.114E+03 +	**4.938E+03 ± 6.243E+02** −	**4.090E+03 ± 1.762E+03** −	7.666E+03 ± 3.884E+03 ∼	4.685E+08 ± 9.958E+08 −	14/3/12
WDO	9.903E+03 ± 2.788E+03 +	5.411E+03 ± 2.167E+02 ∼	5.727E+03 ± 7.067E+02 +	7.387E+03 ± 7.375E+02 +	2.555E+08 ± 1.480E+08 +	24/5/0

Table 4.11: Experimental and statistical results of four WDO variants on CEC'17 benchmark functions with 100 dimensions.

Algorithm	F1	F3	F4	F5	F6	F7
CFDBWDO	**2.950E+10 ± 1.085E+10**	1.677E+05 ± 4.478E+04	1.256E+04 ± 1.564E+03	1.267E+03 ± 1.487E+02	6.618E+02 ± 5.292E+00	**2.644E+03 ± 3.062E+02**
FDBWDO	3.268E+10 ± 1.306E+10 ∼	1.847E+05 ± 4.645E+04 +	1.257E+04 ± 1.578E+03 ∼	**1.236E+03 ± 1.102E+02** ∼	**6.614E+02 ± 6.189E+00** ∼	2.796E+03 ± 2.762E+02 +
AWDO	2.979E+10 ± 8.275E+10 −	4.105E+05 ± 1.181E+05 +	**3.749E+03 ± 1.154E+04** −	1.623E+03 ± 3.860E+02 +	6.940E+02 ± 2.316E+01 +	3.350E+03 ± 4.840E+02 +
WDO	5.822E+10 ± 1.493E+10 +	**1.511E+05 ± 2.584E+04** ∼	1.693E+04 ± 3.517E+03 +	1.329E+03 ± 1.247E+02 +	6.696E+02 ± 8.747E+00 +	2.959E+03 ± 2.527E+02 +
	F8	F9	F10	F11	F12	F13
CFDBWDO	1.668E+03 ± 1.679E+02	**3.215E+04 ± 5.041E+03**	**2.276E+04 ± 6.845E+03**	**3.182E+04 ± 1.068E+04**	1.898E+10 ± 4.618E+09	**1.134E+09 ± 8.014E+08**
FDBWDO	**1.639E+03 ± 1.681E+02** ∼	3.319E+04 ± 7.966E+03 ∼	2.403E+04 ± 6.100E+03 ∼	3.219E+04 ± 1.050E+04 ∼	1.847E+10 ± 4.697E+09 ∼	1.385E+09 ± 8.671E+08 ∼
AWDO	2.054E+03 ± 3.895E+02 +	6.357E+04 ± 2.732E+04 +	2.738E+04 ± 6.757E+03 +	1.966E+05 ± 8.247E+04 +	**7.813E+09 ± 2.898E+10** −	1.657E+09 ± 5.891E+09 −
WDO	1.690E+03 ± 1.443E+02 ∼	4.305E+04 ± 9.680E+03 +	2.817E+04 ± 4.743E+03 +	3.862E+04 ± 1.193E+04 +	3.704E+10 ± 1.130E+10 +	3.334E+09 ± 1.221E+09 +
	F14	F15	F16	F17	F18	F19
CFDBWDO	1.167E+06 ± 1.170E+06	5.306E+07 ± 8.742E+07	**1.019E+04 ± 1.447E+03**	**6.920E+03 ± 2.410E+03**	1.775E+06 ± 7.960E+05	**2.181E+08 ± 1.518E+08**
FDBWDO	**9.908E+05 ± 5.071E+05** ∼	**4.990E+07 ± 9.100E+07** ∼	1.051E+04 ± 1.487E+03 +	7.527E+03 ± 2.427E+03 +	**1.539E+06 ± 5.373E+05** −	2.639E+08 ± 1.994E+08 ∼
AWDO	1.730E+07 ± 3.050E+07 ∼	4.192E+08 ± 2.183E+09 −	1.189E+04 ± 4.420E+03 +	3.634E+04 ± 1.336E+05 +	2.227E+07 ± 4.413E+07 −	5.661E+08 ± 2.284E+09 −
WDO	3.098E+06 ± 4.254E+06 +	5.351E+08 ± 3.297E+08 +	1.184E+04 ± 1.880E+03 +	9.031E+03 ± 2.534E+03 +	4.796E+06 ± 1.051E+07 ∼	1.131E+09 ± 1.048E+09 +
	F20	F21	F22	F23	F24	F25
CFDBWDO	5.809E+03 ± 8.128E+02	3.983E+03 ± 2.036E+02	**2.776E+04 ± 5.759E+03**	5.921E+03 ± 1.550E+02	8.987E+03 ± 4.504E+02	7.698E+03 ± 4.597E+02
FDBWDO	**5.571E+03 ± 9.897E+02** ∼	3.994E+03 ± 1.987E+02 ∼	2.785E+04 ± 6.035E+03 ∼	5.931E+03 ± 1.763E+02 ∼	8.927E+03 ± 5.467E+02 ∼	7.539E+03 ± 3.840E+02 −
AWDO	7.173E+03 ± 9.592E+02 +	**3.782E+03 ± 4.596E+02** −	2.975E+04 ± 5.569E+03 +	**5.370E+03 ± 7.959E+02** −	**7.538E+03 ± 1.302E+03** −	**5.953E+03 ± 6.374E+03** −
WDO	6.441E+03 ± 1.004E+03 +	4.185E+03 ± 1.057E+02 +	3.158E+04 ± 4.554E+03 +	5.925E+03 ± 1.830E+02 ∼	8.959E+03 ± 4.156E+02 ∼	7.723E+03 ± 1.041E+03 ∼
	F26	F27	F28	F29	F30	$w/t/l$
CFDBWDO	2.350E+04 ± 4.385E+03	8.732E+03 ± 8.241E+02	1.277E+04 ± 9.677E+02	1.317E+04 ± 1.865E+03	2.328E+09 ± 8.172E+08	−−
FDBWDO	**2.280E+04 ± 3.857E+03** ∼	8.839E+03 ± 8.456E+02 ∼	1.286E+04 ± 9.968E+02 ∼	**1.278E+04 ± 1.716E+03** ∼	2.388E+09 ± 8.708E+08 ∼	4/23/2
AWDO	2.818E+04 ± 1.188E+04 +	**7.739E+03 ± 1.892E+03** −	**7.555E+03 ± 9.010E+03** −	2.167E+04 ± 2.814E+04 ∼	**1.806E+09 ± 5.231E+09** −	13/2/14
WDO	2.810E+04 ± 5.381E+03 +	9.265E+03 ± 5.151E+02 +	1.371E+04 ± 1.393E+03 +	1.335E+04 ± 1.673E+03 ∼	5.646E+09 ± 1.781E+09 +	22/7/0

Table 4.12: Experimental results of four WDO variants on PEFMSW.

Algorithm	Mean	Std	Best	Worst
CFDBWDO	**19.62**	3.66	**11.64**	25.11
FDBWDO	20.00	3.22	11.73	24.20
AWDO	29.43	0.83	25.33	29.86
WDO	29.23	1.11	22.43	29.79

can further enhance the performance of the algorithm. (3) The comparison between CFDBWDO and AWDO shows that chaotic wind driven optimization with fitness distance balance strategy is better than the previously proposed AWDO employing covariance matrix adaptive evolutionary strategy (CMAES) to update the α, g, RT and c.

In order to intuitively determine the best algorithm, the algorithms are ranked by the Friedman test [88], and the test results are shown in Table 4.6. The smaller the value obtained, the higher the ranking of the algorithm and the better the performance. Accordingly, CFDBWDO is superior to other algorithms.

In order to display the difference between CFDBWDO and other WDO variants, Figs. 4.6-4.13 depict box-and-whisker diagrams and convergence graphs for functions. Figs. 4.6, 4.8, 4.10 and 4.12 are box-and-whisker diagrams. Horizontal axis of box-and-whisker diagram represents four algorithms, and the vertical axis signifies the

Table 4.13: Experimental results of four WDO variants on SSRPPCD.

Algorithm	Mean	Std	Best	Worst
CFDBWDO	**1.752957**	0.115015	**1.46364**	1.965598
FDBWDO	1.774544	0.094761	1.561208	2.023126
AWDO	2.507209	0.251714	1.714494	3.174049
WDO	2.766931	0.245221	1.919205	3.312676

Table 4.14: Experimental results of four WDO variants on ELD.

Algorithm	Mean	Std	Best	Worst
CFDBWDO	**15762.73**	311.65	**15488.88**	17094.55
FDBWDO	15987.99	496.90	15489.65	17364.93
AWDO	16014.53	586.67	15532.57	17586.66
WDO	15914.26	389.34	15532.57	16940.53

error values. The line above the blue box represents the maximum, the line below the box represents the minimum. The lower edge of the blue box denotes the first quartile, the red line denotes the second quartile (median), the upper of the box denotes the third quartile. The red symbol "+" displays the outlier value. The median indicates the average level of sample data. The height of the box indicates the fluctuation of the data. From these box-and-whisker diagrams, it can be seen that the fitness of AWDO fluctuates the most among the four algorithms, indicating that it is the most unstable. The fitness of CFDBWDO is relatively the most concentrated and its medians are also the smallest. Thus, CFDBWDO has superior and more stable performance. Figs. 4.7, 4.9, 4.11 and 4.13 are convergence graphs. The convergence curve can intuitively reflect the convergence speed and accuracy of different algorithms, and the ability to escape from local optima. Horizontal axis of convergence graph expresses NFEs, and the vertical axis shows the logarithm of average optimizations. From these convergence graphs, it can be observed that WDO and AWDO converge prematurely, whereas CFDBWDO and FDBWDO gradually converge, and the curves of CFDBWDO drop even lower. It verifies that the incorporation of fitness distance balance strategy and chaotic local search into the algorithm can improve the exploration ability and explore better solutions. Although the convergence trends

Table 4.15: Experimental results of four WDO variants on OCNLSTR.

Algorithm	Mean	Std	Best	Worst
CFDBWDO	**15.37067**	1.552393	**13.9507**	21.07747
FDBWDO	15.72222	1.658071	14.34348	20.86547
AWDO	20.0472	1.982705	14.44839	21.97958
WDO	20.82233	2.140184	13.95148	23.86827

Table 4.16: Experimental results of four WDO variants on TNEP.

Algorithm	Mean	Std	Best	Worst
CFDBWDO	**223.4118**	7.269598	**220**	258
FDBWDO	225.4171	11.1235	**220**	262
AWDO	1441.315	876.1988	240	3642
WDO	2401.47	1098.524	258	4671.453

of CFDBWDO and FDBWDO are very close, the average optimization values of CFDBWDO are smaller than FDBWDO's, illustrating that the algorithm has been improved after introducing chaotic local search into FDBWDO.

4.5 Six real-world optimization problems

Parameter estimation for frequency-modulated sound waves (PEFMSW) [89] is a highly complex multimodal problem where the optimized parameter vector is $X = (a_1, \omega_1, a_2, \omega_2, a_3, \omega_3)$. The parameter vector is used to generate a sound similar to target sound, the equations are as follows:

$$y(t) = a_1 \sin\left(\omega_1 t\theta + a_2 \sin\left(\omega_2 t\theta + a_3 \sin\left(\omega_3 t\theta\right)\right)\right) \tag{4.1}$$

$$y_0(t) = \sin((5.0)t\theta - (1.5)\sin((4.8)t\theta + (2.0)\sin((4.9)t\theta))) \tag{4.2}$$

where $y(t)$ is the estimated sound wave. $y_0(t)$ is the target sound wave and $\theta = 2\pi/100$. The fitness function is given as:

$$f(\vec{X}) = \sum_{t=0}^{100} \left(y(t) - y_0(t)\right)^2 \tag{4.3}$$

Table 4.17: Experimental results of four WDO variants on CAADP.

Algorithm	Mean	Std	Best	Worst
CFDBWDO	**-11.7621**	0.956935	**-14.151**	-9.84545
FDBWDO	-11.545	1.021445	-13.9281	-7.56846
AWDO	-9.99738	0.209763	-10.2789	-9.44029
WDO	-10.2973	0.100986	-10.4488	-9.99132

Spread spectrum radar poly phase code design (SSRPPCD) [89] is the class of continuous min–max global optimization problem where the continuous variables are $X = \{(x_1, ..., x_n) \in R^n | 0 \leq x_j \leq 2\pi, j = 1, ..., n\}$. It is to minimize the biggest module in the auto-correlation function samples, and the objective function is piecewise smooth. The equations are formulated as follows:

$$\min_{x \in \boldsymbol{X}} f(x) = \max\{\phi_1(x), ..., \phi_{2m}(x)\}, m = 2n - 1 \tag{4.4}$$

$$\phi_{2i-1}(x) = \sum_{j=i}^{n} \cos(\sum_{k=|2i-j-1|+1}^{j} x_k), i = 1, ..., n \tag{4.5}$$

$$\phi_{2i}(x) = 0.5 + \sum_{j=i+1}^{n} \cos(\sum_{k=|2i-j|+1}^{j} x_k), i = 1, ..., n-1 \tag{4.6}$$

$$\phi_{m+i}(x) = -\phi_i(x), i = 1, ..., m \tag{4.7}$$

Static economic load dispatch problem (ELD) is to calculate the minimum fuel cost of generating units in a specific period of operation. The fuel cost includes the cost of online generating units and various constraint costs. The constraints in this paper include power balance constraints, capacity limits, ramp rate limits, and prohibited operating zones. Their specific calculation equations can be referred in [89].

Optimal control of a non-linear stirred tank reactor (OCNLSTR) [89] is a multimodal problem. The chemical modeling process is as follows:

$$\dot{x}_1 = -(2 + u)(x_1 + 0.25) + (x_2 + 0.5) \exp\left(\frac{25x_1}{x_1 + 2}\right) \tag{4.8}$$

$$\dot{x}_2 = 0.5 - x_2 - (x_2 + 0.5) \exp\left(\frac{25x_1}{x_1 + 2}\right) \tag{4.9}$$

$$J = \int_0^{t_f=0.72} \left(x_1^2 + x_2^2 + 0.1 \cdot u^2\right) dt \tag{4.10}$$

where $u(t)$ is flow rate. x_1 and x_2 are dimensionless steady state temperature and concentration deviation, respectively. The optimization goal is to minimize the performance index J.

The objective of transmission network expansion planning problem (TNEP) is to minimize the construction cost of new lines, satisfying no overloads. The specific modeling process of TNEP is in [89].

Circular antenna array design problem (CAADP) [89] achieves sufficient null control by suppressing sidelobes, maximizing directivity of array pattern, driving the maxima close to the desired maxima, and penalizing objective function.

These six real-world optimization problems, including PEFMSW, SSRPPCD, ELD, OCNLSTR, TNEP and CAADP, are used to evaluate the practicality of CFDBWDO. The dimensions of six problems are 6, 20, 6, 1, 7 and 12, respectively. The population size is 100 and all compared algorithms individually run 51 times on each problem. Experimental results are presented in Tables 4.12, 4.13, 4.14, 4.15, 4.16 and 4.17. The Mean, Std, Best and Worst in tables represent mean value, standard deviation, best and worst solutions, respectively. The minimum values in columns of Mean and Best are highlighted in bold.

From Tables 4.12, 4.13, 4.14, 4.15, 4.16 and 4.17, it can be found that CFDBWDO obtains the best means and solutions on the six problems. In Table 4.16, although FDBWDO and CFDBWDO have the same best solution, CFDBWDO has a better mean. Thus, experimental results verify that CFDBWDO has the practicality for optimizing real-world problems.

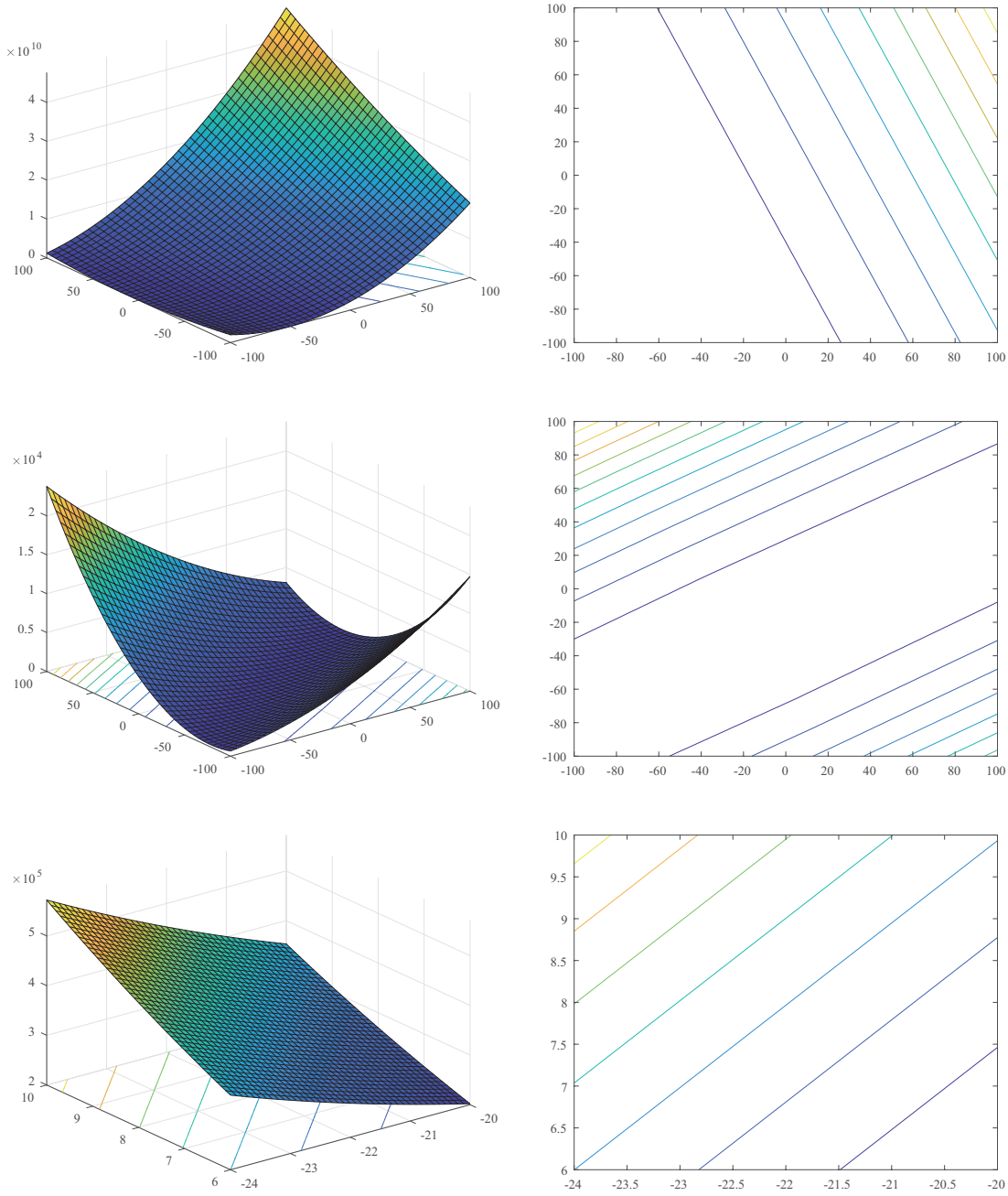

Figure 4.1: 3D and contour maps of F1,F2 and F3.

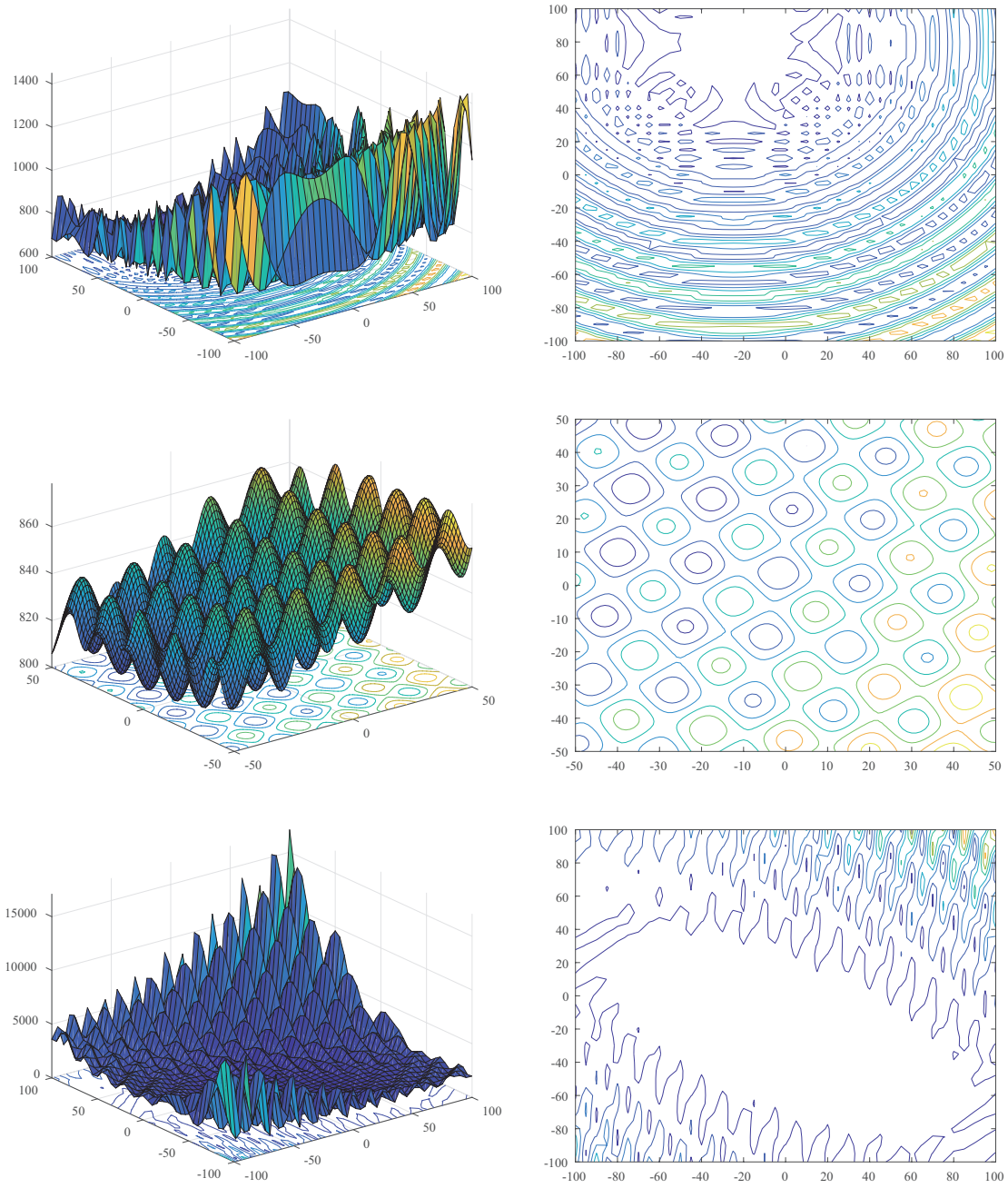

Figure 4.2: 3D and contour maps of F6,F8 and F9.

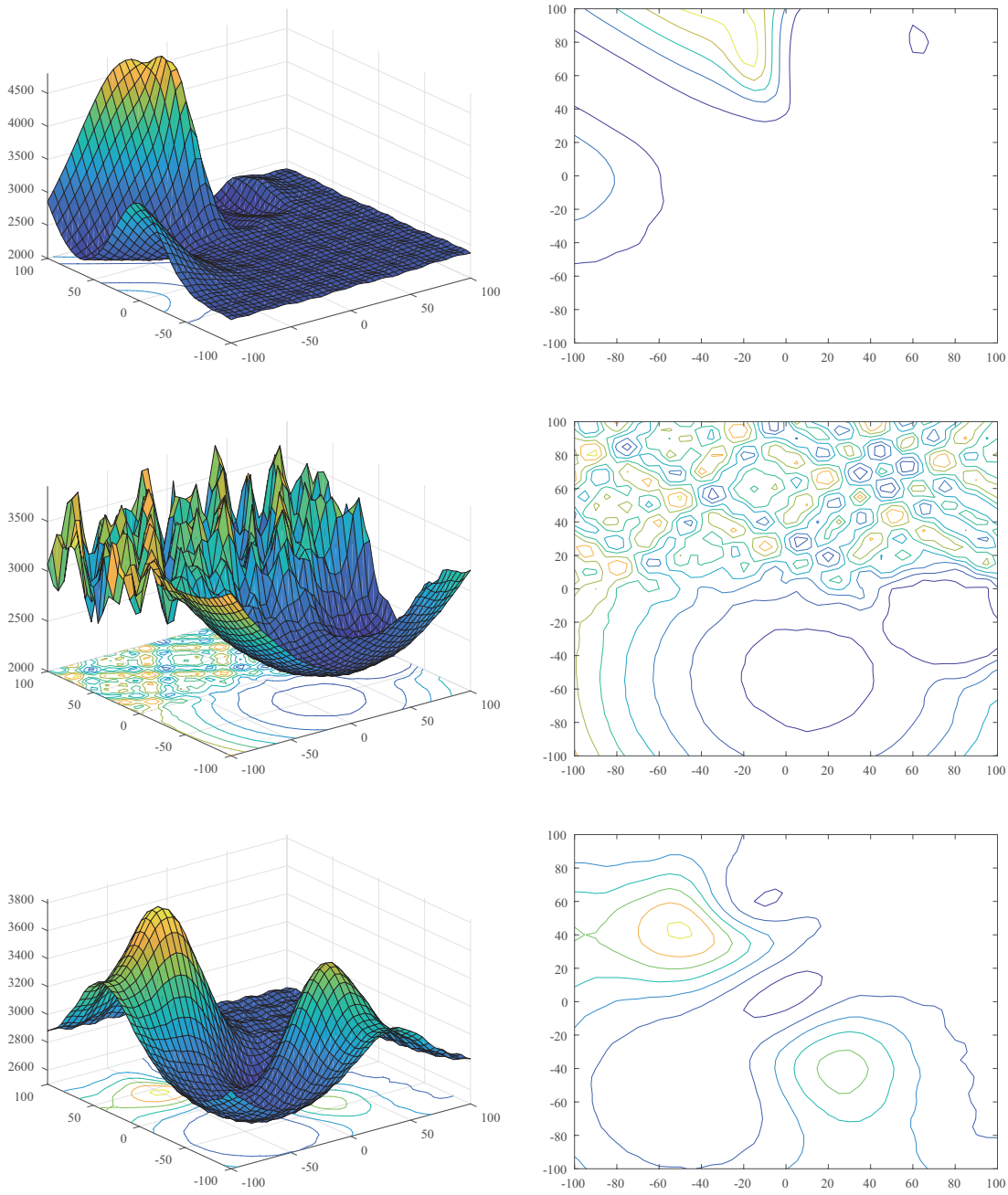

Figure 4.3: 3D and contour maps of F21,F22 and F24.

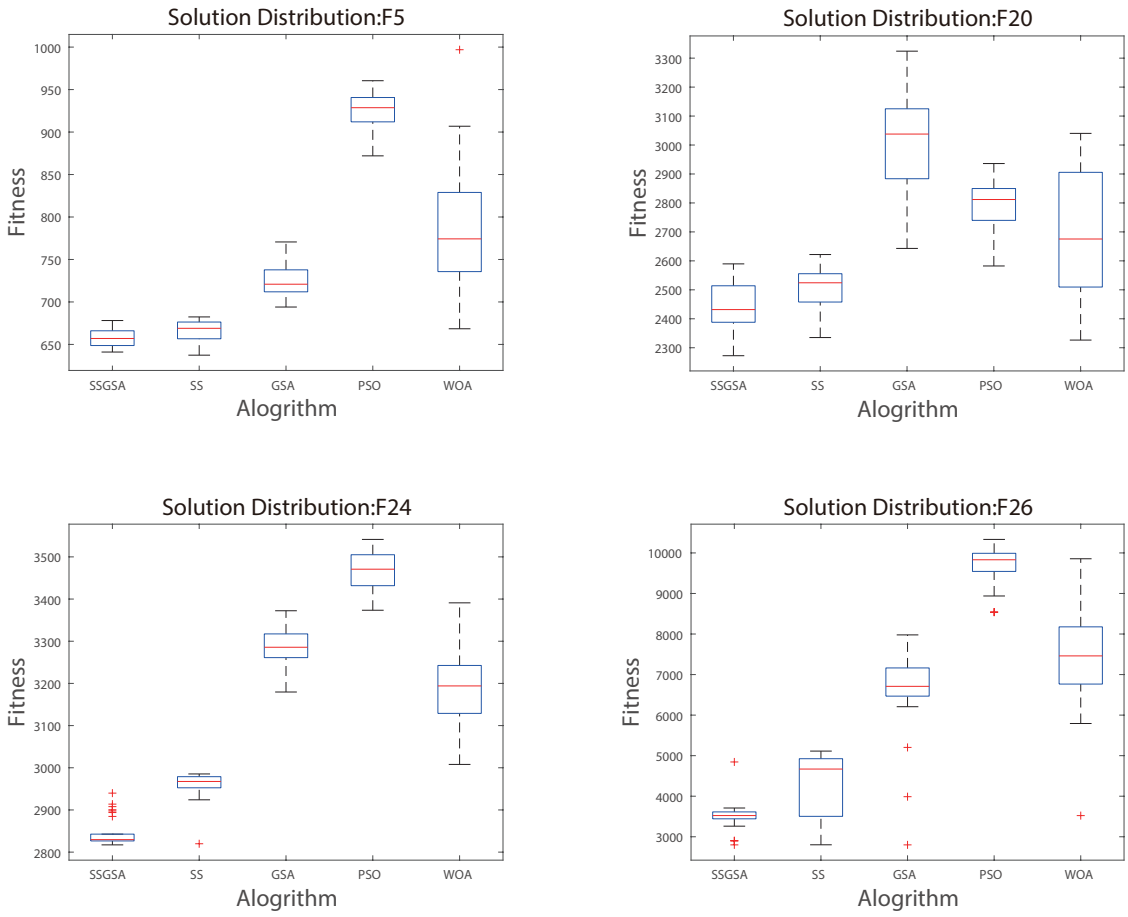

Figure 4.4: Box-and-whisker diagrams of F5, F20, F24, F26.

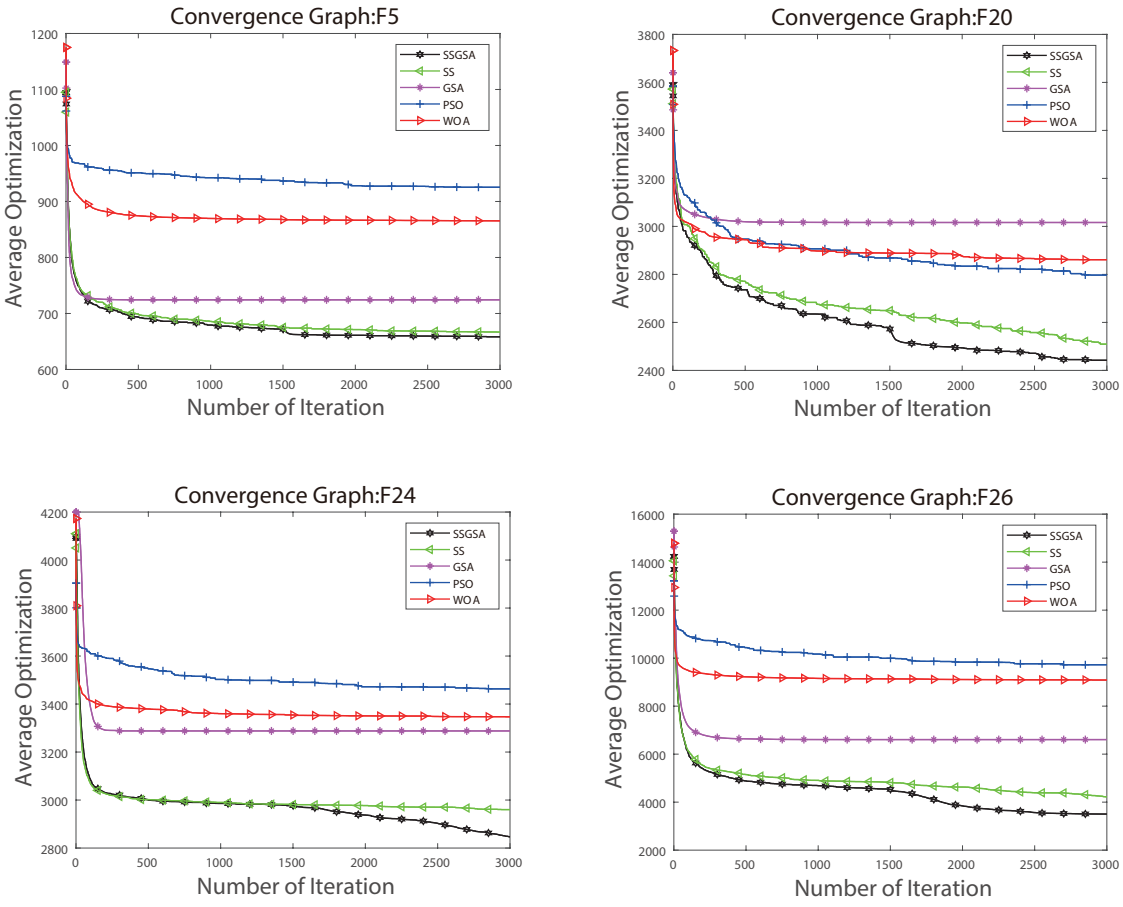

Figure 4.5: Convergence graphs of F5, F20, F24, F26.

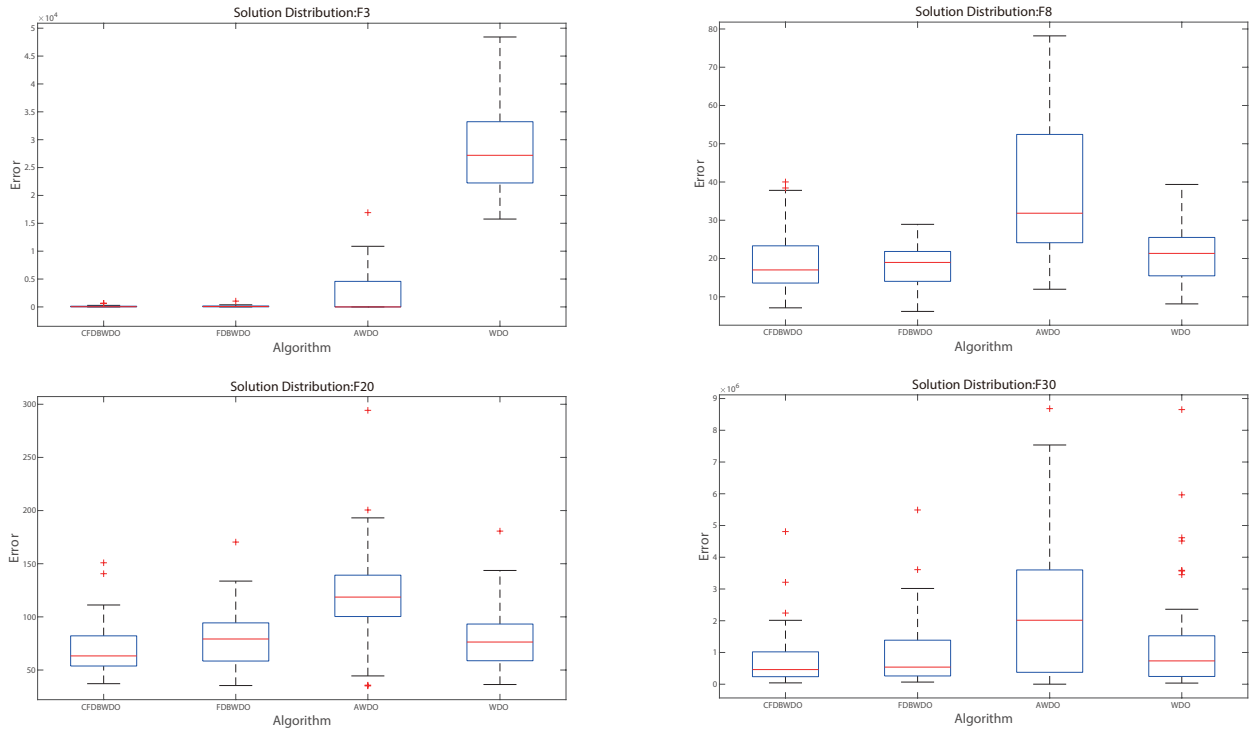

Figure 4.6: Box-and-whisker diagrams of four WDO variants on F3, F8, F20 and F30 with 10 dimensions.

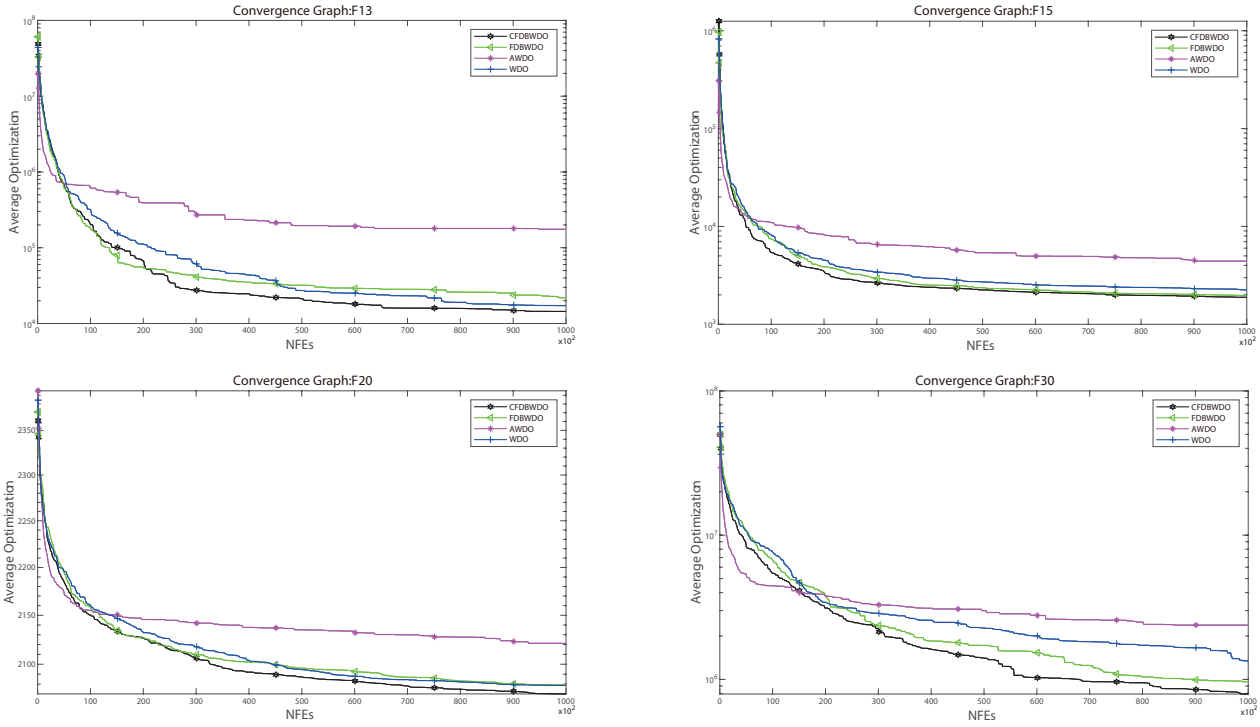

Figure 4.7: Convergence graphs of four WDO variants on F13, F15, F20 and F30 with 10 dimensions.

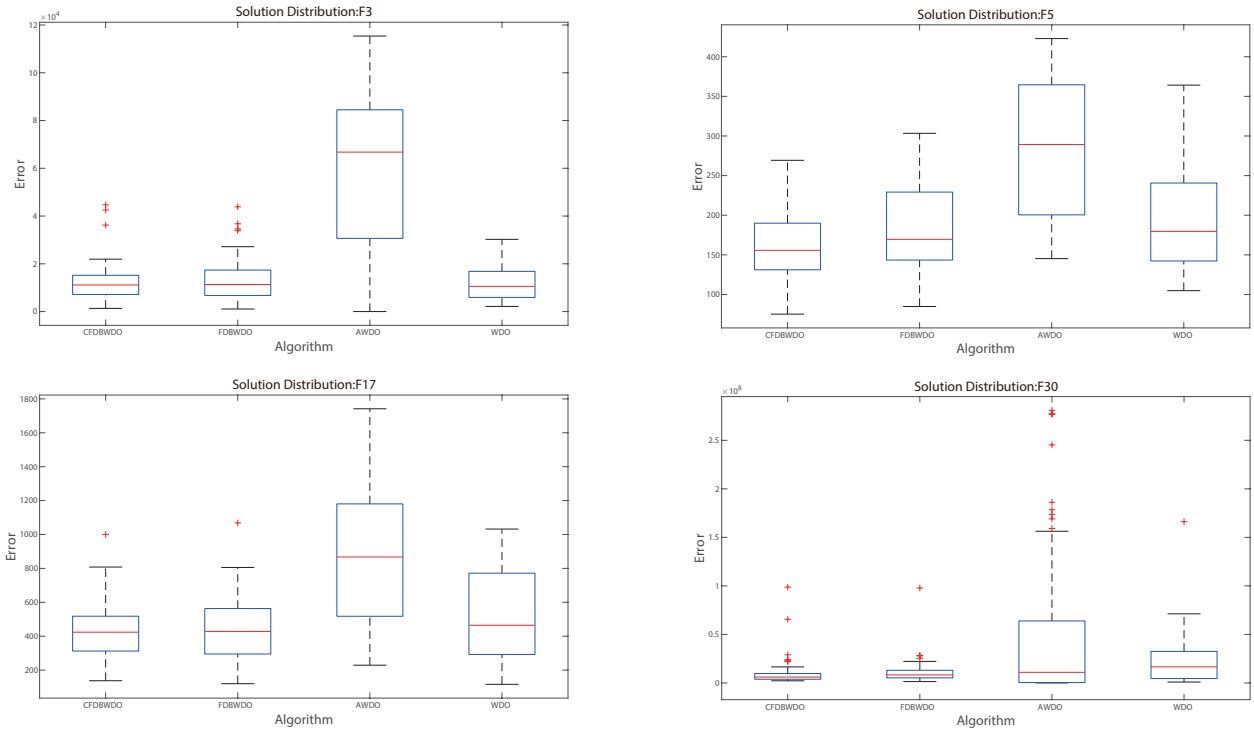

Figure 4.8: Box-and-whisker diagrams of four WDO variants on F3, F5, F17 and F30 with 30 dimensions.

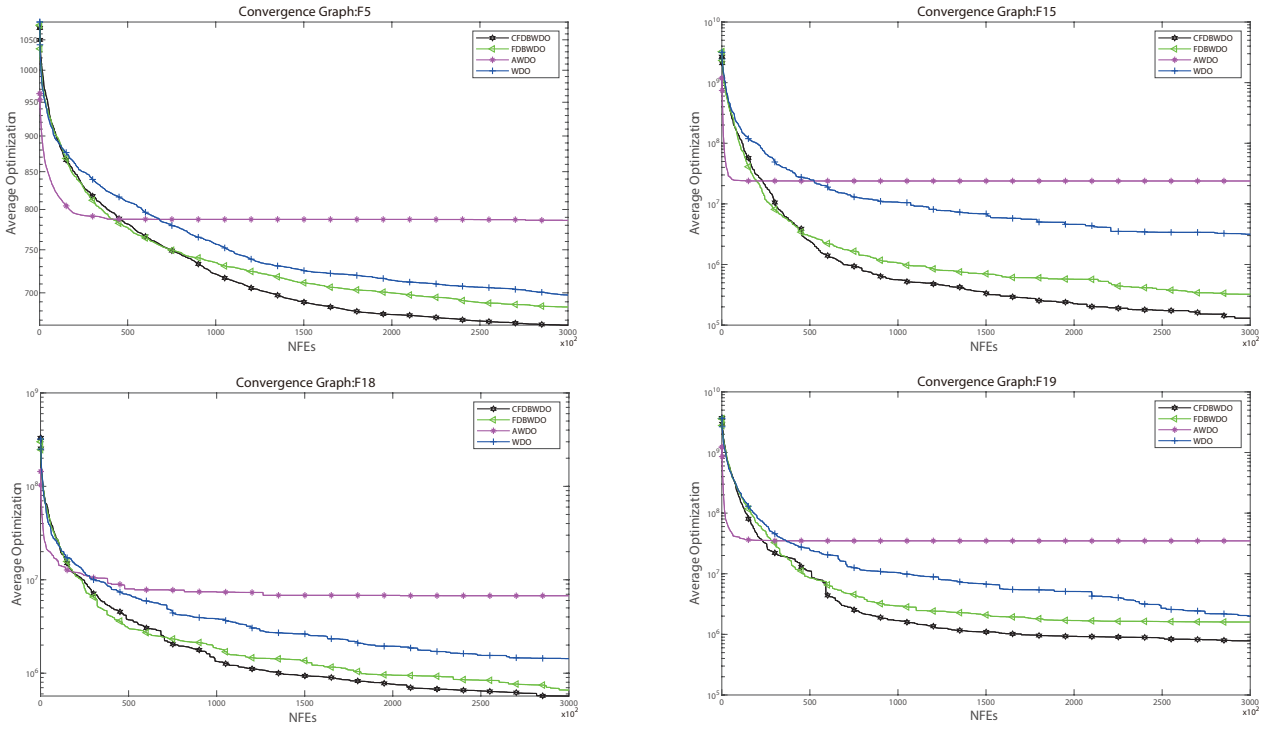

Figure 4.9: Convergence graphs of four WDO variants on F5, F15, F18 and F19 with 30 dimensions.

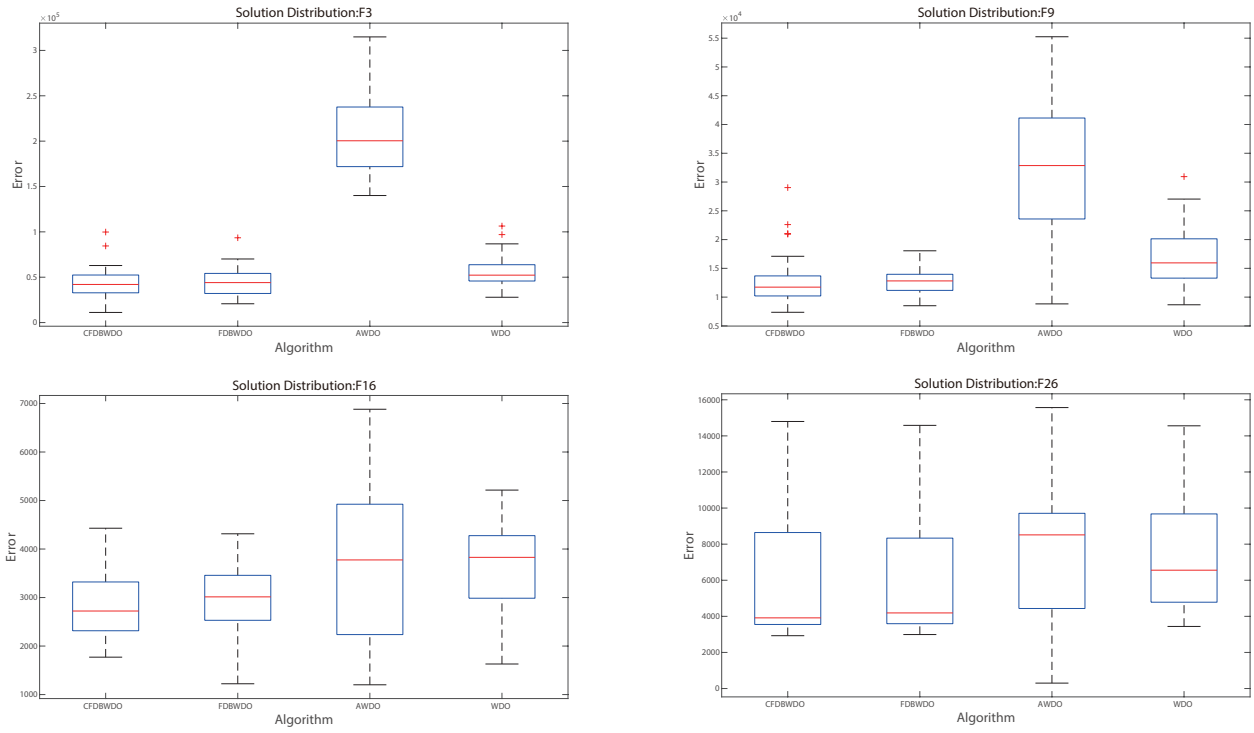

Figure 4.10: Box-and-whisker diagrams of four WDO variants on F3, F9, F16 and F26 with 50 dimensions.

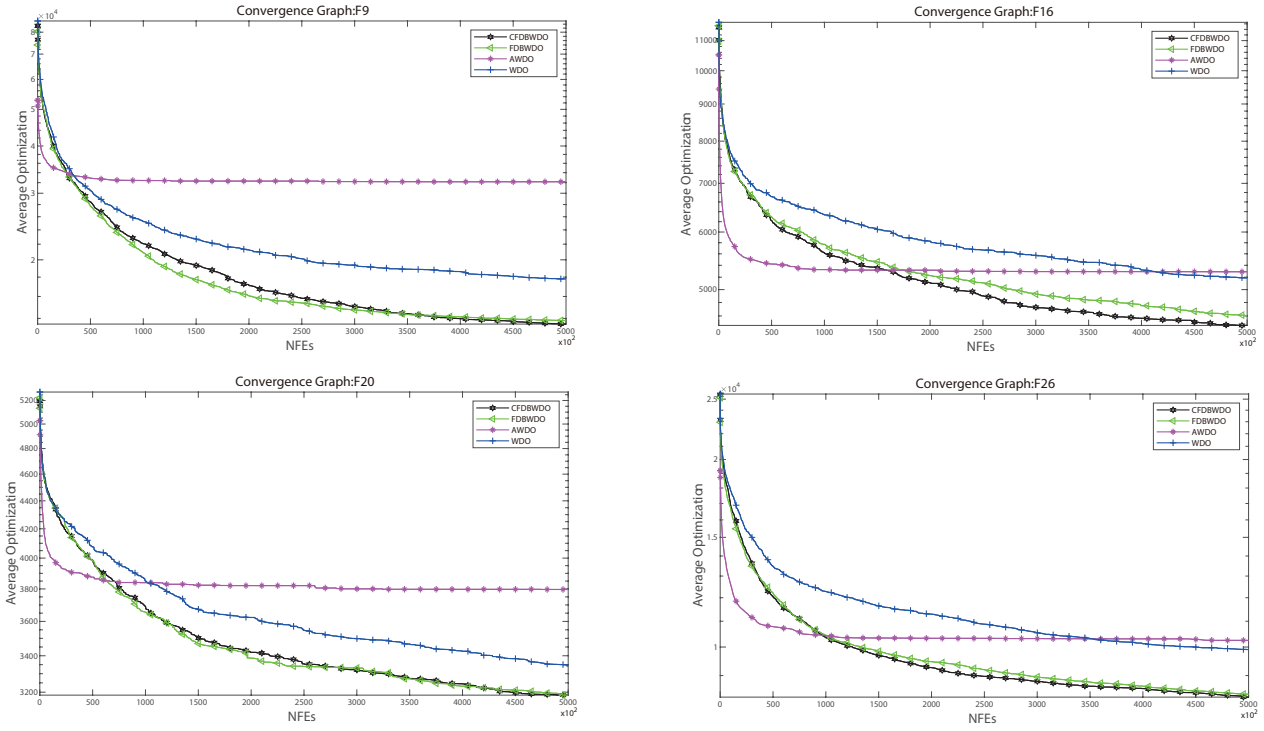

Figure 4.11: Convergence graphs of four WDO variants on F9, F16, F20 and F26 with 50 dimensions.

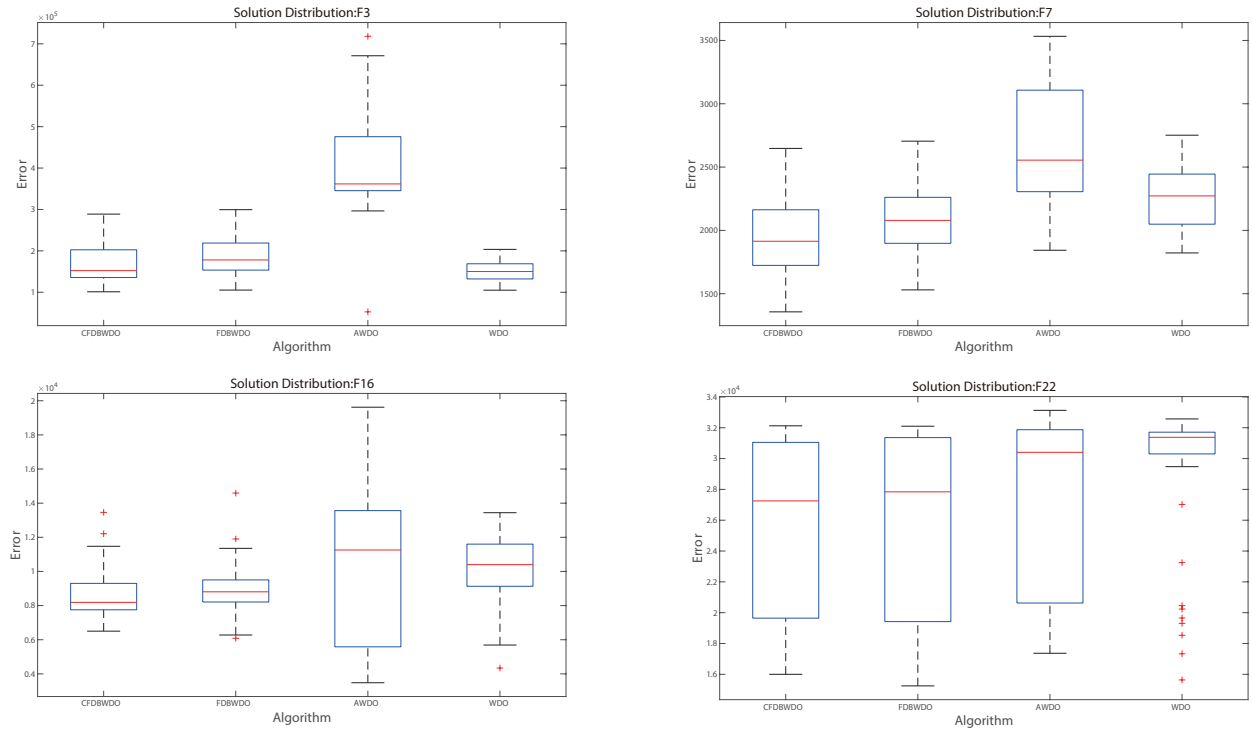

Figure 4.12: Box-and-whisker diagrams of four WDO variants on F3, F7, F16 and F22 with 100 dimensions.

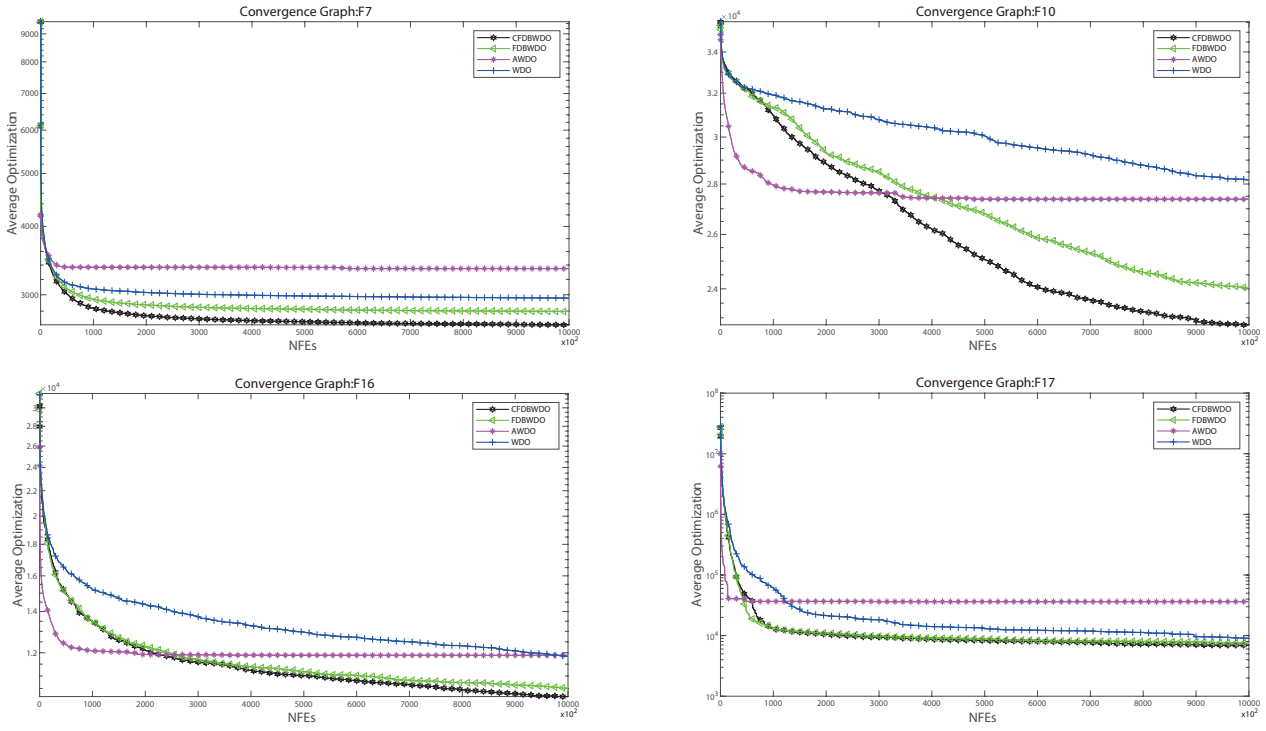

Figure 4.13: Convergence graphs of four WDO variants on F7, F10, F16 and F17 with 100 dimensions.

Chapter 5

Conclusion and future work

By improving SS and WDO, this paper introduces two methods of algorithm improvement: algorithm hybrid and selection method. In order to better improve the exploitation ability of the spherical search, the gravitational search dynamics from GSA replaces the towards-best in it. Experimental results show that the proposed SSGSA is effective in terms of search ability and search efficiency. This study proposes chaotic wind driven optimization with fitness distance balance strategy (CFDB-WDO), which introduces fitness distance balance strategy and memory-based chaotic local search into WDO to solve the problem of WDO with premature convergence and trapping in local optima, and CFDBWDO has the same time complexity as WDO. Experimental results of CFDBWDO and other three WDO variants on twenty-nine CEC 2017 benchmark functions and six real-world optimization problems indicate CFDBWDO has superior performance and practicality.

Three valuable findings in the paper are as follows:

(1) The incorporated FDB selection method can indeed help the algorithm escape from local optima by enriching population diversity, which makes the algorithm converge gradually and improves the exploration ability.

(2) The effect of CLS on improving the exploitation ability of the algorithm is limited by comparing the experimental results of CFDBWDO and FDBWDO,

which clearly verifies the effects of CLS on WDO.

(3) In the large dimensions, CFDBWDO has no significant advantage over AWDO. The possible reason is that the effect of the algorithm on optimizing functions with large dimensions needs further improvement.

In future studies, the following worth work can be considered:

(1) SSGSA and CFDBWDO could be attempted to multi-objective optimization [90], combinatorial optimization [91], neural networks learning [92], solar photovoltaic parameter estimation [93], electromagnetic [53], optimal bidding strategy [94], and robot technology [95, 96].

(2) Improved fitness-distance balance could be applied to other meta-heuristic algorithms to further verify its effectiveness.

(3) The exploitation ability of CFDBWDO could be further improved from other perspectives in addition to CLS.

Bibliography

[1] A. E. Eiben and J. Smith, "From evolutionary computation to the evolution of things," *Nature*, vol. 521, no. 7553, pp. 476–482, 2015.

[2] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003.

[3] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82–117, 2013.

[4] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 10, pp. 1627–1643, 2021.

[5] Y. Zhou, J. Wang, J. Chen, S. Gao, and L. Teng, "Ensemble of many-objective evolutionary algorithms for many-objective problems," *Soft Computing*, vol. 21, no. 9, pp. 2407–2419, 2017.

[6] A. E. Ezugwu, A. K. Shukla, R. Nath, A. A. Akinyelu, J. O. Agushaka, H. Chiroma, and P. K. Muhuri, "Metaheuristics: a comprehensive overview and classification along with bibliometric analysis," *Artificial Intelligence Review*, vol. 54, pp. 4237–4316, 2021.

[7] S. Song, S. Gao, X. Chen, D. Jia, X. Qian, and Y. Todo, "AIMOES: Archive information assisted multi-objective evolutionary strategy for ab initio protein structure prediction," *Knowledge-Based Systems*, vol. 146, pp. 58–72, 2018.

[8] T. Back, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.

[9] X. Yao, *Evolutionary computation: Theory and applications*. World Scientific, 1999.

[10] X. Yao and Y. Xu, "Recent advances in evolutionary computation," *Journal of Computer Science and Technology*, vol. 21, no. 1, pp. 1–18, 2006.

[11] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the first European Conference on Artificial Life*, vol. 142. Paris, France, 1991, pp. 134–142.

[12] R. Skinderowicz, "Implementing a GPU-based parallel MAX–MIN ant system," *Future Generation Computer Systems*, vol. 106, pp. 277–295, 2020.

[13] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, and Z. Tang, "Ant colony optimization with clustering for solving the dynamic location routing problem," *Applied Mathematics and Computation*, vol. 285, pp. 149–173, 2016.

[14] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[16] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. IEEE, 1995, pp. 39–43.

[17] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[18] Y. Wang, S. Gao, M. Zhou, and Y. Yu, "A multi-layered gravitational search algorithm for function optimization and real-world problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 94–109, 2021.

[19] K. Mori, M. Tsukiyama, and T. Fukuda, "Immune algorithm with searching diversity and its application to resource allocation problem," *IEEJ Transactions on Electronics, Information and Systems*, vol. 113, no. 10, pp. 872–878, 1993.

[20] Z. Xu, Y. Wang, S. Li, Y. Liu, Y. Todo, and S. Gao, "Immune algorithm combined with estimation of distribution for traveling salesman problem," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 11, no. S1, 2016.

[21] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[23] Z. Zhao, S. Liu, M. Zhou, and A. Abusorrah, "Dual-objective mixed integer linear program and memetic algorithm for an industrial group scheduling problem," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 6, pp. 1199–1209, 2020.

[24] S. Gao, R.-L. Wang, H. Tamura, and Z. Tang, "A multi-layered immune system for graph planarization problem," *IEICE Transactions on Information and Systems*, vol. E92-D, no. 12, pp. 2498–2507, 2009.

[25] H. Garg, M. Rani, and S. Sharma, "An efficient two phase approach for solving reliability–redundancy allocation problem using artificial bee colony technique," *Computers & Operations Research*, vol. 40, no. 12, pp. 2961–2969, 2013.

[26] Z. Abo-Hammour, O. A. Arqub, O. Alsmadi, and S. Momani, "An optimization algorithm for solving systems of singular boundary value problems," *Applied Mathematics & Information Sciences*, vol. 8, no. 6, pp. 2809–2821, 2014.

[27] Z. Abo-Hammour, O. Abu Arqub, S. Momani, and N. Shawagfeh, "Optimization solution of troesch's and bratu's problems of ordinary type using novel continuous genetic algorithm," *Discrete Dynamics in Nature and Society*, vol. 2014, 2014.

[28] O. Abu Arqub, Z. Abo-Hammour, S. Momani, and N. Shawagfeh, "Solving singular two-point boundary value problems using continuous genetic algorithm," in *Abstract and Applied Analysis*, vol. 2012. Hindawi, 2012.

[29] R. Miikkulainen and S. Forrest, "A biological perspective on evolutionary computation," *Nature Machine Intelligence*, vol. 3, no. 1, pp. 9–15, 2021.

[30] J. Ji, S. Gao, J. Cheng, Z. Tang, and Y. Todo, "An approximate logic neuron model with a dendritic structure," *Neurocomputing*, vol. 173, pp. 1775–1783, 2016.

[31] J. Yang, Y. Zhang, Z. Wang, Y. Todo, B. Lu, and S. Gao, "A cooperative coevolution wingsuit flying search algorithm with spherical evolution," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, p. 178, 2021.

[32] G. Ning and Y. Zhou, "Application of improved differential evolution algorithm in solving equations," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, p. 199, 2021.

[33] R. M. Rizk-Allah, O. Saleh, E. A. Hagag, and A. A. A. Mousa, "Enhanced unicate swarm algorithm for solving large-scale nonlinear optimization problems," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, p. 189, 2021.

[34] C. Mallika and S. Selvamuthukumaran, "A hybrid crow search and grey wolf optimization technique for enhanced medical data classification in diabetes diagnosis system," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, p. 157, 2021.

[35] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292–305, 2016.

[36] ——, "A hybrid GSA-GA algorithm for constrained optimization problems," *Information Sciences*, vol. 478, pp. 499–523, 2019.

[37] ——, "An approach for solving constrained reliability-redundancy allocation problems using cuckoo search algorithm," *Beni-Suef University Journal of Basic and Applied Sciences*, vol. 4, no. 1, pp. 14–25, 2015.

[38] T. Kundu and H. Garg, "A hybrid ITLHHO algorithm for numerical and engineering optimization problems," *International Journal of Intelligent Systems*, 2021.

[39] ——, "A hybrid TLNNABC algorithm for reliability optimization and engineering design problems," *Engineering with Computers*, 2022.

[40] Y. Yu, Z. Lei, Y. Wang, T. Zhang, C. Peng, and S. Gao, "Improving dendritic neuron model with dynamic scale-free network-based differential evolution," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 99–110, 2022.

[41] W.-J. Hong, P. Yang, and K. Tang, "Evolutionary computation for large-scale multi-objective optimization: A decade of progresses," *International Journal of Automation and Computing*, vol. 18, pp. 155–169, 2021.

[42] A. Kumar, R. K. Misra, D. Singh, S. Mishra, and S. Das, "The spherical search algorithm for bound-constrained global optimization problems," *Applied Soft Computing*, vol. 85, p. 105734, 2019.

[43] S. Gao, C. Vairappan, Y. Wang, Q. Cao, and Z. Tang, "Gravitational search algorithm combined with chaos for unconstrained numerical optimization," *Applied Mathematics and Computation*, vol. 231, pp. 48–62, 2014.

[44] S. Gao, Y. Todo, T. Gong, G. Yang, and Z. Tang, "Graph planarization problem optimization based on triple-valued gravitational search algorithm," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 9, no. 1, pp. 39–48, 2014.

[45] Z. Song, S. Gao, Y. Yu, J. Sun, and Y. Todo, "Multiple chaos embedded gravitational search algorithm," *IEICE Transactions on Information and Systems*, vol. E100.D, no. 4, pp. 888–900, 2017.

[46] Y. Wang, Y. Yu, S. Gao, H. Pan, and G. Yang, "A hierarchical gravitational search algorithm with an effective gravitational constant," *Swarm and Evolutionary Computation*, vol. 46, pp. 118–139, 2019.

[47] Y. Wang, S. Gao, Y. Yu, Z. Wang, J. Cheng, and T. Yuki, "A gravitational search algorithm with chaotic neural oscillators," *IEEE Access*, vol. 8, pp. 25 938–25 948, 2020.

[48] Z. Lei, S. Gao, S. Gupta, J. Cheng, and G. Yang, "An aggregative learning gravitational search algorithm with self-adaptive gravitational constants," *Expert Systems with Applications*, p. 113396, 2020.

[49] Z. Bayraktar, M. Komurcu, and D. H. Werner, "Wind driven optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics," in *2010 IEEE Antennas and Propagation Society International Symposium*. IEEE, 2010, pp. 1–4.

[50] P. Civicioglu and E. Besdok, "A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, vol. 39, no. 4, pp. 315–346, 2013.

[51] Z. Bayraktar, M. Komurcu, Z. H. Jiang, D. H. Werner, and P. L. Werner, "Stub-loaded inverted-F antenna synthesis via wind driven optimization," in *2011 IEEE International Symposium on Antennas and Propagation (APSURSI)*. IEEE, 2011, pp. 2920–2923.

[52] Z. Bayraktar, J. P. Turpin, and D. H. Werner, "Nature-inspired optimization of high-impedance metasurfaces with ultrasmall interwoven unit cells," *IEEE Antennas and Wireless Propagation Letters*, vol. 10, pp. 1563–1566, 2011.

[53] Z. Bayraktar, M. Komurcu, J. A. Bossard, and D. H. Werner, "The wind driven optimization technique and its application in electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 5, pp. 2745–2757, 2013.

[54] A. Boulesnane and S. Meshoul, "A new multi-region modified wind driven optimization algorithm with collision avoidance for dynamic environments," in *Advances in Swarm Intelligence*. Springer International Publishing, 2014, pp. 412–421.

[55] E. Hochsteiner De Vasconcelos Segundo, A. L. Amoroso, V. C. Mariani, and L. Dos Santos Coelho, "A wind driven approach using lévy flights for global continuous optimization," in *2014 2nd International Conference on Artificial Intelligence, Modelling and Simulation*. IEEE, 2014, pp. 75–80.

[56] B. Kuldeep, V. Singh, A. Kumar, and G. Singh, "Design of two-channel filter bank using nature inspired optimization based fractional derivative constraints," *ISA Transactions*, vol. 54, pp. 101–116, 2015.

[57] Z. Bao, Y. Zhou, L. Li, and M. Ma, "A hybrid global optimization algorithm based on wind driven optimization and differential evolution," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[58] Z. Bayraktar and M. Komurcu, "Adaptive wind driven optimization," in *Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016, pp. 124–127.

[59] ——, "Multiobjective adaptive wind driven optimization." in *IJCCI (ECTA)*, 2016, pp. 115–120.

[60] P. Di Barba, M. E. Mognaschi, S. Wiak, M. Przybylski, and B. Slusarek, "Wind-driven optimization for the design of switched reluctance motors," in *2017 18th International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering (ISEF) Book of Abstracts*. IEEE, 2017, pp. 1–2.

[61] D. Mathew, C. Rani, M. Rajesh Kumar, Y. Wang, R. Binns, and K. Busawon, "Wind-driven optimization technique for estimation of solar photovoltaic parameters," *IEEE Journal of Photovoltaics*, vol. 8, no. 1, pp. 248–256, 2018.

[62] I. A. Ibrahim, M. J. Hossain, B. C. Duck, and C. J. Fell, "An adaptive wind-driven optimization algorithm for extracting the parameters of a single-diode PV cell model," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 1054–1066, 2020.

[63] S. Sawant and P. Manoharan, "A hybrid optimization approach for hyperspectral band selection based on wind driven optimization and modified cuckoo search optimization," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 1725–1748, 2021.

[64] L. Abualigah, A. Diabat, and Z. W. Geem, "A comprehensive survey of the harmony search algorithm in clustering applications," *Applied Sciences*, vol. 10, no. 11, p. 3827, 2020.

[65] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[66] Y. Yu, S. Gao, S. Cheng, Y. Wang, S. Song, and F. Yuan, "CBSO: a memetic brain storm optimization with chaotic local search," *Memetic Computing*, vol. 10, no. 4, pp. 353–367, 2018.

[67] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 6, pp. 3954–3967, 2021.

[68] C. D. Ahrens and R. Henson, *Meteorology today: an introduction to weather, climate, and the environment.* Cengage Learning, 2021.

[69] H. Riehl, *Introduction to the Atmosphere.* McGraw-Hill, 1972.

[70] R. D. Thompson, *Atmospheric processes and systems.* Routledge, 2002.

[71] J. M. Wallace and P. V. Hobbs, *Atmospheric science: an introductory survey*. Elsevier, 2006, vol. 92.

[72] B. Chen, Z. Cao, and S. Yu, "PID parameters optimization based on wind driven optimization algorithm," *Computer Engineering and Applications*, vol. 52, no. 14, pp. 250–253, 2016.

[73] Z. Xu, H. Yang, J. Li, X. Zhang, B. Lu, and S. Gao, "Comparative study on single and multiple chaotic maps incorporated grey wolf optimization algorithms," *IEEE Access*, vol. 9, pp. 77 416–77 437, 2021.

[74] R. Parpinelli and H. Lopes, "New inspirations in swarm intelligence: a survey," *International Journal of Bio-Inspired Computation*, vol. 3, no. 1, pp. 1–16, 2011.

[75] J. Krause, J. Cordeiro, R. S. Parpinelli, and H. S. Lopes, "A survey of swarm algorithms applied to discrete optimization problems," in *Swarm Intelligence and Bio-Inspired Computation*. Elsevier, 2013, pp. 169–191.

[76] E. Alba and B. Dorronsoro, "The exploration/exploitation tradeoff in dynamic cellular genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 2, pp. 126–142, 2005.

[77] N. Lynn and P. N. Suganthan, "Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation," *Swarm and Evolutionary Computation*, vol. 24, pp. 11–24, 2015.

[78] H. T. Kahraman, S. Aras, and E. Gedikli, "Fitness-distance balance (FDB): A new selection method for meta-heuristic search algorithms," *Knowledge-Based Systems*, vol. 190, p. 105169, 2020.

[79] K. Wang, S. Tao, R.-L. Wang, Y. Todo, and S. Gao, "Fitness-distance balance with functional weights: A new selection method for evolutionary algorithms,"

IEICE Transactions on Information and Systems, vol. 104, no. 10, pp. 1789–1792, 2021.

[80] N. Awad, M. Ali, J. Liang, B. Qu, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," *Technical Report*, 2016.

[81] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization," *Technical Report*, 2017.

[82] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization," *Journal of Heuristics*, vol. 15, no. 6, p. 617, 2008.

[83] J. Luengo, S. García, and F. Herrera, "A study on the use of statistical tests for experimentation with neural networks: Analysis of parametric test conditions and non-parametric tests," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7798–7808, 2009.

[84] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.

[85] Y. Yu, S. Gao, Y. Wang, J. Cheng, and Y. Todo, "ASBSO: An improved brain storm optimization with flexible search length and memory-based selection," *IEEE Access*, vol. 6, pp. 36 977–36 994, 2018.

[86] S. Gao, Y. Wang, J. Wang, and J. Cheng, "Understanding differential evolution: A poisson law derived from population interaction network," *Journal of Computational Science*, vol. 21, pp. 140–149, 2017.

[87] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 2019, doi: 10.1109/TSMC.2019.2956121.

[88] J. Carrasco, S. García, M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm and Evolutionary Computation*, vol. 54, p. 100665, 2020.

[89] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems," *Jadavpur University, Nanyang Technological University, Kolkata*, pp. 341–359, 2010.

[90] Z. Lei, S. Gao, Z. Zhang, M. Zhou, and J. Cheng, "MO4: A many-objective evolutionary algorithm for protein structure prediction," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 3, pp. 417–430, 2022.

[91] J. Cheng, X. Wu, M. Zhou, S. Gao, Z. Huang, and C. Liu, "A novel method for detecting new overlapping community in complex evolving networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 9, pp. 1832–1844, 2019.

[92] T. Zhou, S. Gao, J. Wang, C. Chu, Y. Todo, and Z. Tang, "Financial time series prediction using a dendritic neuron model," *Knowledge-Based Systems*, vol. 105, pp. 214–224, 2016.

[93] S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, and J. Cheng, "A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models," *Energy Conversion and Management*, vol. 230, p. 113784, 2021.

[94] K. Jain and A. Saxena, "Simulation on supplier side bidding strategy at day-ahead electricity market using ant lion optimizer," *Journal of Computational and Cognitive Engineering*, 2022.

[95] J. Zan, "Research on robot path perception and optimization technology based on whale optimization algorithm," *Journal of Computational and Cognitive Engineering*, 2022.

[96] S. Momani, Z. S. Abo-Hammour, and O. MK. Alsmadi, "Solution of inverse kinematics problem using genetic algorithms," *Applied Mathematics & Information Sciences*, vol. 10, no. 1, pp. 225–233, 2016.