Hindawi
Scientific Programming
Volume 2019, Article ID 2401818, 20 pages
https://doi.org/10.1155/2019/2401818

Research Article

Dynamically Dimensioned Search Embedded with Piecewise Opposition-Based Learning for Global Optimization

Jianzhong Xu,[1] Fu Yan ,[1] Kumchol Yun,[1,2] Sakaya Ronald,[3] Fengshu Li,[1] and Jun Guan[4]

[1]*School of Economics and Management, Harbin Engineering University, Harbin 150001, China*
[2]*Faculty of Mechanics, Kim Il Sung University, Pyongyang 950003, Democratic People's Republic of Korea*
[3]*College of Civil and Building Engineering, Kyambogo University, Kampala, Uganda*
[4]*College of Economics and Management, Northeast Forestry University, Harbin 150040, China*

Correspondence should be addressed to Fu Yan; yanfuphd@163.com

Received 2 January 2019; Revised 4 April 2019; Accepted 12 May 2019; Published 26 May 2019

Academic Editor: Basilio B. Fraguela

Dynamically dimensioned search (DDS) is a well-known optimization algorithm in the field of single solution-based heuristic global search algorithms. Its successful application in the calibration of watershed environmental parameters has attracted researcher's extensive attention. The dynamically dimensioned search algorithm is a kind of algorithm that converges to the global optimum under the best condition or the good local optimum in the worst case. In other words, the performance of DDS is easily affected by the optimization conditions. Therefore, this algorithm has also suffered from low robustness and limited scalability. In this work, an improved version of DDS called DDS-POBL is proposed. In the DDS-POBL, two effective methods are applied to improve the performance of the DDS algorithm. Piecewise opposition-based learning is introduced to guide DDS search in the right direction, and the golden section method is used to search for more promising areas. Numerical experiments are performed on a set of 23 classic test functions, and the results represent significant improvements in the optimization performance of DDS-POBL compared to DDS. Several experimental results using different parameter values demonstrate the high solution quality, strong robustness, and scalability of the proposed DDS-POBL algorithm. A comparative performance analysis between the DDS-POBL and other powerful algorithms has been carried out by statistical methods by using the significance of the results. The results show that DDS-POBL works better than PSO, CoDA, MHDA, NaFA, and CMA-ES and gives very competitive results when compared to INMDA and EEGWO. Moreover, the parameter calibration application of the Xinanjiang model shows the effectiveness of the DDS-POBL in the real optimization problem.

1. Introduction

The rapid development of productivity of human society has brought a great demand for optimization algorithms. Obtaining a good solution to the complex optimization problems in the real world becomes the specialized task for the optimization algorithms. Traditional optimization algorithms such as Newton's method and the gradient method, which are based on mathematical theory, can hardly solve these complex optimization problems due to the extreme computation burdens. Therefore, highly efficient optimization algorithms have become the focus of research in recent years. The metaheuristic algorithm inspired from various phenomena of nature is one of the prevailing highly efficient algorithms. The biggest characteristic of these algorithms is to continuously evaluate candidate solutions through multiple iterations and try to improve upon these solutions. These metaheuristic algorithms are usually classified into two main categories [1]: single solution-based heuristic global search algorithms and population-based heuristic algorithms. Some of the famous single solution-based heuristic global search algorithms are simulated annealing (SA) [2], threshold accepting method (TA) [3], microcanonical Annealing (MA) [4], tabu search (TS) [5], guided local search (GLS) [6], and dynamically dimensioned search (DDS) [7, 8]. Population-based ones include evolutionary algorithms (EA) [9], genetic algorithms (GA) [10], particle swarm optimization (PSO) [11], dragonfly algorithm

(DA) [12, 13], and shuffled complex evolution (SCE) algorithms [14].

According to the No Free Lunch (NFL) theorem [15], it is hard for researchers to propose a metaheuristic algorithm that is best suited for solving all optimization problems. That is to say, a particular algorithm may show very promising solutions only on certain problems but not on others. From this view, both the single solution-based heuristic global search algorithms and population-based heuristic algorithms have their respective strengths and weaknesses. The main trouble they all encounter is that the rates of convergence are very low, thus bringing them both a high computing burden and low results accuracy and limiting their applications in the real world. This study will focus on single solution-based heuristic global search algorithms especially the DDS algorithm and try to rectify its slow convergence speed and low solutions accuracy.

The dynamically dimensioned search (DDS) algorithm, introduced by Tolson and Shoemaker [7], provides a relatively new potential for the family of single solution-based heuristic global search algorithms. At the initial stages of iteration, the algorithm is mainly based on the global search and is converted to local search at the later stages of iteration. This special search mechanism of the DDS algorithm is achieved by dynamically and probabilistically reducing the number of dimensions in the neighborhood [7]. Different versions of DDS have been proposed and successfully applied to practical engineering optimization problems such as the hybrid discrete dynamically dimensioned search (HD-DDS) which was used to solve discrete, single-objective, constrained water distribution system (WDS) design problems [8], the modified dynamically dimensioned search (MDDS) which was presented to optimize the parameter for distributed hydrological model [16], the DDS algorithm which was used to automate the calibration process of an unsteady river flow model [17], the Pareto archived dynamically dimensioned search (PA-DDS) which was applied for multi-objective optimization [18], and the combining filter method and dynamically dimensioned search which was designed for constrained global optimization problems [19]. Although the DDS algorithm partly overcomes the common drawback of single solution-based search algorithms to some extent, it does not still provide an ideal solution to address the poor and slow convergence of the global optimum in the best case or an acceptable local optimum in the worst case completely.

The drawbacks of the DDS algorithm, by which the global optimal is obtained in the best case and a local optimum is achieved in the worst case, make DDS not to be a perfect algorithm. In practical applications, most optimization problems involve complex constraints. To overcome these drawbacks and obtain a solution that is globally optimal or close to it, DDS needs to be improved. Through the in-depth analysis of the potential solution update principle of the DDS algorithm, we found that DDS does not have a clear search direction when updating this potential solution. This makes us realize that the lack of clear search direction in DDS may be the main reason why it cannot converge to the global optimal vicinity in poor conditions. In literature [20], POBL is introduced to be an algorithm that can guide the search direction of each dimension. In this paper, we use POBL to guide the search direction of the DDS algorithm. In addition, the golden section (GS) search is used in conjunction with the POBL algorithm to guide DDS to quickly gather the potential solution near the global optimal solution and accelerate the convergence speed of the DDS. Hence, an improved version of dynamically dimensioned search algorithm named the "dynamically dimensioned search algorithm embedded with piecewise opposition-based learning" (DDS-POBL) algorithm is presented in this work. The proposed algorithm reconstructs the framework of the DDS algorithm by introducing piecewise opposition-based learning (POBL). The bounds of each variable will be updated dynamically along with the number of iterations and increased by adopting the golden ratio and the piecewise opposite number of the obtained best positions. A dynamically dimensioned search algorithm combined with dynamic bounds adjustment and opposition-based learning is one of the major contributions of this work.

The rest of the content of the paper can be listed as follows: in Section 2, a brief description to the DDS algorithm is presented. Section 3 contains description of motivation and the proposed DDS-POBL algorithm. Several numerical experiments and statistical analysis are reported in Section 4 and the comparison experiments between the proposed DDS-POBL and standard DDS and other state-of-the-art algorithms has also been carried out. And finally, a brief conclusion and future research direction are detailed in Section 5.

2. Dynamically Dimensioned Search Algorithm

The DDS algorithm is a greedy type of algorithm developed by Tolson and Shoemaker in 2007 [7]. The main purpose of the proposed DDS algorithm is to solve the calibration problems that exist in the context of watershed simulation models. Since it is easy for programming and based on a simple concept and takes into account the global and local search, it has attracted extensive attentions for research studies. The main difference between the DDS algorithm and existing optimization algorithms is that the neighborhood is dynamically adjusted by changing the dimension of the search and no algorithm parameter needs tuning. In the optimization process, the DDS algorithm obtains a good approximation of the globally optimal solution, rather than the precise global optimum within a specified maximum number of function evaluations. Thus, DDS is suited for computationally expensive optimization problems such as distributed watershed model calibration.

As mentioned above, the DDS algorithm takes into account the global and local search in its iteration because the algorithm searches globally in the initial iterations and becomes increasingly local when approaching the maximum allowable number of iterations [7, 16]. In each iteration, j is randomly selected with probability P from the decision variables D for inclusion in neighborhood I_{perturb}. The expression of probability P is as follows:

$$P(k) = 1 - \frac{\ln(k)}{\ln(k_{\max})}, \tag{1}$$

where k indicates the number of current iteration and $k_{\max}$ represents the maximum number of iteration.

At each iteration k, a new potential solution $x_{k,j}^{\text{trial}}$ is obtained by perturbing the current best solution $x_{k,j}^{\text{best}}$ only in the randomly selected dimensions. These perturbation magnitudes are sampled using a standard normal random variable $N(0,1)$, reflecting at decision variable bounds as follows:

$$x_{k,j}^{\text{trial}} = \begin{cases} x_{k,j}^{\text{best}}, & \text{if } I_{\text{perturb}}(j) \in \varnothing, \\ x_{k,j}^{\text{best}} + r \cdot \left(ub_j - lb_j\right) \cdot N, & \text{otherwise}, \end{cases} \tag{2}$$

where r is a scalar neighborhood size perturbation factor, ub_j and lb_j correspond to the upper and lower bounds of the jth dimension (variable), and N denotes a standard normal random number.

In order to accurately choose the best solution between the current best x_k^{best} and the trial potential x_k^{trial} for the next iteration, the greedy search method is employed. The current best x_k^{best} will be replaced by the trial x_k^{trial} if the objective function value of x_k^{trial} is smaller than x_k^{best} ($f(x_k^{\text{trial}}) < f(x_k^{\text{best}})$); otherwise, the current best position x_k^{best} is reserved for the next iteration. The pseudocode description of the DDS algorithm is presented in Algorithm 1.

3. Proposed DDS-POBL Algorithm

3.1. Analysis of the Low Searching Ability of DDS Algorithm. As described above, the DDS algorithm is not only a local optimization algorithm, but it also has strong ability to search globally. At the beginning, the probability that provokes the number of dimensions to be perturbed in the neighborhood is large, which makes the DDS algorithm searche globally. With the increase of iterations, the probability decreases gradually and becomes a local search. During this search, the trial candidate is achieved by perturbation operation on the current best point by using standard normal distribution, which brings more useful information to the search.

Based on the above analysis, it can be found that the DDS algorithm can be successfully transformed from the global search to the local search with the decrease of perturbation probability and that the diversity of trial candidate solution is also obtained from the current best solution through the disturbance of the standard normal distribution. But then two natural questions are raised: what is the effective search direction for the DDS algorithm, and which promising area is more likely that the trial candidate solution or the current best solution will approach the global optimum? Unfortunately, however, it is difficult to find reasonable answers from the existing literature. This is the main drawback of the DDS algorithm, and probably this is the reason why the final solution of DDS is always a good approximation of the globally optimal solution rather than the precise global optimum. It tells us that it is very important to adaptively adjust the search direction and search area in the iteration process. Therefore, an improved search mechanism for guiding the DDS algorithm search efficiently is needed in avoiding the problem of premature convergence and stagnation in local optima.

To accomplish this task, in this work, the dynamically dimensioned search algorithm embedded with piecewise opposition-based learning has been proposed, in which the search direction is determined by piecewise opposition-based learning strategy, and the promising areas are obtained by adjusting the boundaries of variables using the golden ratio.

3.2. Piecewise Opposition-Based Learning (POBL) Theory. Before the detailed description of the piecewise opposition-based learning theory, it is necessary to define the important concepts related to this algorithm.

Definition 1. Let $P(x_1, x_2, \ldots, x_n)$ be a point in n-dimensional space, where $x_1, x_2, \ldots, x_n \in \Omega$ and $x_i \in [lb_i, ub_i]$ $\forall i \in 1, 2, \cdots, n$. The piecewise opposite point of P is defined by $P^+(x_1^+, x_2^+, \cdots, x_n^+)$:

$$x_i^+ = lb_i + ub_i - x_i. \tag{3}$$

Let $P^+(x_1, x_2, \ldots, x_n)$ be a candidate solution in an n-dimensional solution space with $x_i \in [lb_i, ub_i]$ $\forall i \in 1, 2, \cdots, n$. The quality of a candidate solution is measured by calculating its fitness function value $f(x)$. If $f(P^+) \geq f(P)$ (for a maximization problem) or $f(P^+) \leq f(P)$ (for a minimization task), then replace P with P^+; otherwise, P is preserved in the next iteration. A formal description of the piecewise opposition-based learning algorithm is presented in Algorithm 2.

3.3. Piecewise Opposition-Based Learning Determines the Right Search Direction. For the current heuristic optimization algorithms, how to find its best search direction does not have a fixed operation mode. In general, researchers tend to use a greedy search method, namely, the trial-and-error method, to determine the search direction of candidate x. At each iteration, the algorithm tries to improve upon this candidate x until it eventually converges to the ideal optimal solution or meets certain predefined termination criteria. Therefore, the computational burden of these algorithms subjects to the quality of the candidate solution. However, the initial estimates are not always close to the actual solution. In some cases where they place on the opposite location of where the current candidate solution resides, it is always computationally expensive for the algorithm to converge [21]. According to Tizhoosh [21], searching for the candidate solution at all directions simultaneously is an available method to improve the poor quality of the initial guess. Fortunately, the POBL algorithm is proposed as the right way to guide the algorithm to search for the right direction based on this intuition, in which the opposite

Inputs: Scalar neighborhood size perturbation factor $r = 0.2$, maximum number of iterations $k_{\max}$, number of variables (dimension) n, upper bounds ub and lower bounds lb

Outputs: x^{best} and f^{best}

Initialization

  $x^0 = [x_1^0, x_2^0, \ldots, x_n^0]$, $x_i^0 = lb_j + \text{rand} \cdot (ub_j - lb_j)$

  Set $k = 1$, $x^{\text{best}} = x^0$, $f^{\text{best}} = f(x^{\text{best}})$, $I_{\text{perturb}} = \varnothing$

while $k \leq k_{\max}$ **do**

  Compute the probability of perturbing the decision variables P_k using equation (1)

  for $j = 1$ to n **do**

    Generate uniform random numbers, ω

    if $\omega < P_k$ **then**

      Set $I_{\text{perturb}}(j) = 1$

    end if

  end for

    Generate a standard normal random numbers, N

  for $j = 1$ to n **do**

      $x_j^{\text{trial}} = x_j^{\text{best}} + (ub_j - lb_j) \cdot I_{\text{perturb}}(j) \cdot r \cdot N$ //equation (2)

  end for

  for $j = 1$ to n **do**

    if $x_j^{\text{trial}} < lb_j$ **then**

      Set $x_j^{\text{trial}} = lb_j + (lb_j - x_j^{\text{trial}})$

      if $x_j^{\text{trial}} > ub_j$ **then**

        Set $x_j^{\text{trial}} = lb_j$

       end if

      end if

      if $x_j^{\text{trial}} > ub_j$ **then**

        Set $x_j^{\text{trial}} = ub_j - (x_j^{\text{trial}} - ub_j)$

        if $x_j^{\text{trial}} < lb_j$ **then**

          Set $x_j^{\text{trial}} = ub_j$

        end if

      end if

  end for

  Evaluate $f(x^{\text{trial}})$

  if $f(x^{\text{trial}}) < f^{\text{best}}$ **then**

    Set $x^{\text{best}} = x^{\text{trial}}$, $f^{\text{best}} = f(x^{\text{trial}})$

  end if

    Set $k = k + 1$

end while

Algorithm 1: DDS algorithm.

estimate x^+ of x is simultaneously taken into account with x at each iteration of the algorithm. Therefore, introducing the POBL theory into the DDS algorithm can guide the algorithm to search the right direction effectively. Meanwhile, in order to obtain a better initial guess, we set the current best solution of the standard DDS algorithm as the initial candidate solution of the POBL algorithm, as shown in Algorithm 2.

3.4. The Golden Section (GS) Can Guide the Solutions to Search for Promising Areas. The golden section (GS) is one of the most amazing phenomena present in nature. The successful applications of GS-based rule have been widely observed. One of them is the optimal method based on the GS. The core framework of this method is to divide the space into several sections or groups. The main goal is to separate the space using this region division method in a way that allows for better functionality, spacing, and distribution [22]. It is well known that the feasible or theoretical optimal solution of the optimization problem is distributed in a certain interval of its solution space. Therefore, if we can find such a promising interval quickly, the algorithm will quickly find the optimal solution. At present, the bisection method and GS are two main fast search methods that gradually reduce the interval to find the minimum. However, the search efficiency of the GS is higher than that of the bisection method. It has been proved in practice that the GS method can achieve the results obtained through 2,500 experiments by the bisection method only with 16 trials. Therefore, the GS search is an efficient method to gradually reduce the interval where the minimum value is located. The key is to keep that no matter how many points have been evaluated, the minimum value is within the interval defined by the two points adjacent to the point with the lowest evaluated value so far [23]. According to [23], its specific implementation steps and principle are as follows:

Figure 1 shows a simple step in the technique of finding a minimum in one-dimensional solution space of a unimodal function. As described in Figure 1, the functional values of $f(x)$

Begin: Input x^{best} and its fitness value $f(x^{\text{best}})$, golden section ratio $g = 0.618$
(1) Set $P = x^{\text{best}}$, $f(P) = f(x^{\text{best}})$
(2) **for** $j = 1$ to n **do**
(3) $P_j^{+} = lb_j + ub_j - x_j^{\text{best}}$
(4) **end for**
(5) Calculate the fitness value of P^{+}, $f(P^{+})$
(6) **if** $f(P^{+}) > f(P)$ **then**
(7) $P^{+} = P$
(8) **end if**
(9) Set $x^{\text{best}} = P^{+}$
Update the search interval $[lb_j, ub_j]$ for next iteration using GS method
(10) **for** $j = 1$ to n **do**
(11) Set $X_1 = lb_j + g \cdot (ub_j - lb_j)$, $X_2 = ub_j - g \cdot (ub_j - lb_j)$
(12) Calculate the fitness values of X_1 and X_2, f_1 and f_2
(13) **if** $f(X_1) > f(X_2)$ **then**
(14) Set $lb_j = X_1$
(15) **else**
(16) Set $ub_j = X_2$
(17) **end if**
(18) **end for**

Algorithm 2: POBL and GS algorithm [21].

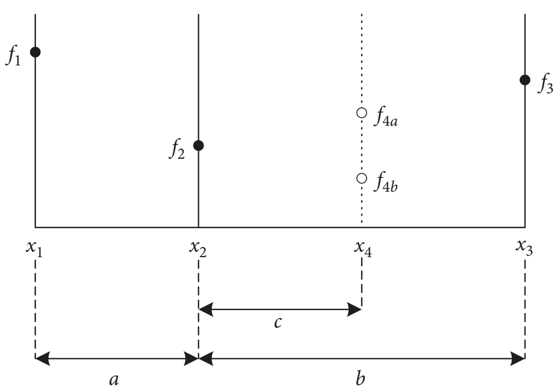


Figure 1: Schematic diagram of the golden section.

are represented on the vertical axis, and the horizontal axis is the range of values of x parameter. First, the functional values of $f(x)$ at the three points of x_1, x_2, and x_3 are calculated, and the results that $f_1 > f_3 > f_2$ are known. Since f_2 is smaller than either f_1 or f_3, it is clear that the interval from x_1 to x_3 is a promising area that can be used to find a minimum.

The next operation in the minimization process is to "probe" the function by evaluating a new value x, namely, x_4. Since the unimodal functional values satisfy $f_2 < f_3 < f_1$, a function value smaller or bigger than f_2 can be find in the interval between x_2 and x_3. Therefore, it is most effective to select x_4 in the largest interval, i.e., between x_2 and x_3. From Figure 1, in the case where the function yields f_{4a}, a minimum lies between x_1 and x_4, and the three points x_1,x_2, and x_4 will be the new triplet points. However, if the function yields the value of f_{4b}, then a minimum lies between x_2 and x_3, and the new triplet of points will be x_2, x_4, and x_3. Based on this analysis, in either case, we can find a narrower promising search area guaranteed to contain the minimum value of function.

3.5. Design of Proposed DDS Embedded with Piecewise Opposition-Based Learning: DDS-POBL. Based on the analysis of the POBL algorithm, it can be seen that a good choice of the initial guess is crucial for the algorithm to quickly find the right search direction. To apply the POBL algorithm correctly to the DDS algorithm and to help it find the right search direction, we use the current best solution obtained from the DDS algorithm as the initial guess for POBL. After using the POBL algorithm to determine the optimal search direction for the DDS algorithm, the next step is to use the GS search method to guide the DDS algorithm to search for the most promising interval. By determining the correct search direction of the DDS algorithm and guiding it search to the most promising area, the present DDS-POBL algorithm has greatly improved its global and local optimization performance compared with the standard version of the DDS algorithm, and its applications have also been greatly expanded.

The pseudocode of the proposed DDS-POBL algorithm is described as Algorithm 3. In Algorithm 3, we first initialize the parameters of the DDS algorithm and set $r = 0.2$, $g = 0.618$, and $I_{\text{perturb}} = 0$. Then, we run the DDS algorithm using pseudocode of lines 1 through 31 to get a better solution x^{best} and its fitness value $f(x^{\text{best}})$. Next, we invoke Algorithm 2 to find the right search direction and use the x^{best} as the initial input for the POBL algorithm. A potential optimal solution P^{+} is found by executing the POBL algorithm, i.e., executing the first to the ninth lines of Algorithm 2. Finally, the GS method is used to search the most promising interval, i.e., the 10th through 18th lines of Algorithm 2 are executed.

4. Numerical Experiments and Results

4.1. Benchmark Functions and Parameter Settings. A set of benchmark functions, including 23 test problems, were collected from several literature studies [13, 24–26] and applied

Inputs: Scalar neighborhood size perturbation factor $r = 0.2$, maximum number of iterations $k_{\max}$, number of variables (dimension) n, upper bounds ub and lower bounds lb
Outputs: x^{best} and f^{best}
Initialization
    $x^0 = [x_1^0, x_2^0, \ldots, x_n^0]$, $x_i^0 = lb_j + \text{rand} \cdot (ub_j - lb_j)$
    Set $k = 1$, $x^{\text{best}} = x^0$, $f^{\text{best}} = f(x^{\text{best}})$, $I_{\text{perturb}} = \varnothing$
while $k \leq k_{\max}$ **do**
    Compute the probability of perturbing the decision variables P_k using equation (1)
    for $j = 1$ to n **do**
        Generate uniform random numbers, ω
        if $\omega < P_k$ **then**
            Set $I_{\text{perturb}}(j) = 1$
        end if
    end for
    Generate a standard normal random numbers, r
    for $j = 1$ to n **do**
        $x_j^{\text{trial}} = x_j^{\text{best}} + (ub_j - lb_j) \cdot I_{\text{perturb}}(j) \cdot r$
    end for
    for $j = 1$ to n **do**
        if $x_j^{\text{trial}} < lb_j$ **then**
            Set $x_j^{\text{trial}} = lb_j + (lb_j - x_j^{\text{trial}})$
            if $x_j^{\text{trial}} > ub_j$ **then**
                Set $x_j^{\text{trial}} = lb_j$
            end if
        end if
        if $x_j^{\text{trial}} > ub_j$ **then**
            Set $x_j^{\text{trial}} = ub_j - (x_j^{\text{trial}} - ub_j)$
            if $x_j^{\text{trial}} < lb_j$ **then**
                Set $x_j^{\text{trial}} = ub_j$
            end if
        end if
    end for
    Evaluate $f(x^{\text{trial}})$
    if $f(x^{\text{trial}}) < f^{\text{best}}$ **then**
        Set $x^{\text{best}} = x^{\text{trial}}$, $f^{\text{best}} = f(x^{\text{trial}})$
    end if
End of DDS and Invoking Algorithm 2
    Invoking Algorithm 2
    Set $k = k + 1$
end while
return x^{best} and f^{best}

Algorithm 3: DDS-POBL algorithm.

to evaluate the performance of the proposed algorithm. The selected test functions are classified into two categories: unimodal and multimodal benchmark test functions. The unimodal problems with only one global optimum and no local optima are suitable for investigating the exploitation of the DDS-POBL algorithm. However, multimodal problems are often used to investigate the exploration performance of the algorithm and the local optimal avoidance ability since it has more than one local optimal solution [27]. Detailed description of the selection of test functions and their respective global optimums are summarized in Table 1. In Table 1, $f_{\min}$ represents the global optimal value.

In order to obtain a fair comparison, we set the neighborhood disturbance parameters and the maximum number of iterations of the DDS and DDS-POBL algorithms to 0.2 and 500, respectively, i.e., $r = 0.2$ and $k_{\max} = 500$. The DDS-POBL and conventional DDS algorithms were coded in MATLAB R2015a, and all experiments were performed on a personal computer (Core i5@ 2.5 GHz, 2.70 GHz, and 64 GB RAM).

4.2. The Evaluation Metrics of Function Optimization Performance. When evaluating the optimization performance of an algorithm, researchers often prefer to investigate its efficiency through metrics such as the *acceleration rate* (AR) and *success rate* (SR). According to [21, 27], AR is a metric related to the convergence speed of the algorithm. In the present work, AR is employed to compare the convergence rate of the DSS-POBL algorithm against the convergence rate of the DDS algorithm. It is defined as follows:

$$\text{AR}_{\text{DDS-POBL}} = \frac{\text{NFC}_{\text{DDS}}}{\text{NFC}_{\text{DDS-POBL}}}, \tag{4}$$

Table 1: Summary of the benchmark test functions used in the experiment of this work.

Function type	Function name	Definition	Search interval	$f_{\min}$
Unimodal functions	Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[−100, 100]	0
	Sum-square	$f_2(x) = \sum_{i=1}^{n} i x_i^2$	[−10, 10]	0
	Schwefel's 2.22	$f_3(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	[−10, 10]	0
	Rotated hyperellipsoid	$f_4(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[−100, 100]	0
	Schwefel's 2.21	$f_5(x) = \max\{\lvert x_i\rvert, 1 \le x_i \le n\}$	[−100, 100]	0
	Rosenbrock	$f_6(x) = \sum_{i=1}^{n} [100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[−30, 30]	0
	Step	$f_7(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	[−100, 100]	0
	Quartic	$f_8(x) = \sum_{i=1}^{n} i x_i^4$	[−1.28, 1.28]	0
	Noise	$f_9(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	[−1.28, 1.28]	0
	Sum-power	$f_{10}(x) = \sum_{i=1}^{n} \lvert x_i\rvert^{(i+1)}$	[−1, 1]	0
Multimodal functions	Rastrigin	$f_{11}(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[−5.12, 5.12]	0
	Ackley	$f_{12}(x) = -20\exp(-0.2\sqrt{(1/n)\sum_{i=1}^{n} x_i^2}) - \exp((1/n)\sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$	[−32, 32]	0
	Griewank	$f_{13}(x) = (1/4000)\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	[−600, 600]	0
	Levy	$f_{14}(x) = \sum_{i=1}^{n} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \sin^2(3\pi x_1) + \lvert x_n - 1\rvert [1 + \sin^2(3\pi x_n)]$	[−10, 10]	0
	Alpine	$f_{15}(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1 x_i\rvert$	[−10, 10]	0
	Inverted cosine mixture	$f_{16}(x) = 0.1n - (0.1\sum_{i=1}^{n} \cos(5\pi x_i) - \sum_{i=1}^{n} x_i^2)$	[−1, 1]	0
	Zakharov	$f_{17}(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5 i x_i)^2 + (\sum_{i=1}^{n} 0.5 i x_i)^4$	[−5, 10]	0
	Pathological	$f_{18}(x) = \sum_{i=2}^{n} 0.5 + ((\sin^2(\sqrt{100 x_{i-1}^2 + x_i^2}) - 0.5) / (1 + 0.001 (x_{i-1}^2 - 2x_{i-1}x_i + x_i^2)^2))$	[−100, 100]	0
	Levy and montalo	$f_{19}(x) = 0.1 (\sin^2(3\pi x_1)) + \sum_{i=1}^{n-1} (x_i - 1)^2 (1 + \sin 2(3\pi x_{i+1})) + (x_n - 1)^2 (1 + \sin^2(2\pi x_n))$	[−5, 5]	0
	Elliptic	$f_{20}(x) = \sum_{i=1}^{n} (10^6)^{(i-1)/(n-1)} x_i^2$	[−100, 100]	0
	Easom	$f_{21}(x) = (-1)^{n+1} \prod_{i=1}^{n} \cos(x_i) \cdot \exp[-\sum_{i=1}^{n} (x_i - \pi)^2]$	[−100, 100]	0
	Salomon	$f_{22}(x) = 1 - \cos(2\pi\sqrt{\sum_{i=1}^{n} x_i^2}) + 0.1\sqrt{\sum_{i=1}^{n} x_i^2}$	[−100, 100]	0
	Schaffer	$f_{23}(x) = 0.5 + (\sin^2(\sqrt{\sum_{i=1}^{n} x_i^2}) - 0.5/(1 + 0.001(\sum_{i=1}^{n} x_i^2))^2)$	[−100, 100]	0

where NFC indicates the number of function calls and the reported NFCs are used in the present experiments for each function. For the average over 50 trials, $\mathrm{AR} > 1$ means that the convergence rate of DDS-POBL is faster than DDS, whereas $\mathrm{AR} = 1$ means that DDS-POBL has the same convergence rate to DDS; $\mathrm{AR} < 1$ means that the convergence rate of DDS-POBL is slower than DDS.

The *success rate* (SR) is an important metric to evaluate the performance for the optimization algorithm, which is defined as the ratio of the number of trials that the algorithm successfully reaches the desired value before reaching the maximum number of function calls and the total number of experimental trails. Its expression is described as follows:

$$\mathrm{SR} = \frac{\text{number of trials reaching the desired value}}{\text{total number of trials}}. \quad (5)$$

Based on the description of AR and SR, the average AR ($\mathrm{AR}_{\mathrm{ave}}$) and SR ($\mathrm{SR}_{\mathrm{ave}}$) can be calculated over the n benchmark functions as follows:

$$\mathrm{AR}_{\mathrm{ave}} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{AR}_i,$$
$$\mathrm{SR}_{\mathrm{ave}} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{SR}_i. \quad (6)$$

4.3. Comparison with Conventional DDS Algorithm. A comparison result of optimization performance between the proposed DDS-POBL algorithm and the conventional dynamically dimensioned search (DDS) algorithm for 23 test problems is presented in Table 2. In this experiment, each test function was tested 50 times for 30 and 60 dimensions to observe the performance of both algorithms. Four common criteria were introduced to compare the algorithms [25], i.e., Best, Mean, Worst, and Standard deviation.

From Table 2, for the 30 and 60 dimensions, the DDS-POBL algorithm has better "Best," "Mean," "Worst," and "St. dev" values for all test functions except f_{21} and f_{23} than the conventional DDS algorithm. For the four test functions (i.e. f_7, f_8, f_{13}, and f_{21}), the DDS-POBL algorithm could achieve theoretical optima (0). Moreover, on the test functions f_1–f_5, f_{14}–f_{17}, f_{19}, and f_{20}, the DDS-POBL algorithm provided much closer results to the global optimum. Both DDS-POBL and DDS could obtain global optimum in test function f_{21}. DDS had better "St. dev" value than DDS-POBL in function f_{23} and better "Best" value in function f_{18} for dimension 30. In addition, the DDS algorithm provided solutions close to the solution of the DDS-POBL algorithm in the test functions f_9, f_{11}, f_{12}, f_{18}, f_{22}, and f_{23}.

To analyze the convergence of the algorithms, the convergence curves of the DDS-POBL and DDS on some typical test functions have been plotted for 30 and 60

Table 2: Best, Mean, Worst, and Standard deviation value obtained by the DDS and DDS-POBL algorithm for 23 test functions. The best results are highlighted in italic.

Function	Dim	DDS				DDS-POBL			
		Best	Mean	Worst	St. dev	Best	Mean	Worst	St. dev
f_1	30	3.08E+004	4.93E+004	6.52E+004	7.73E+003	*1.82E−203*	*1.74E−202*	*1.48E−201*	*0.00E+00*
	60	1.34E+005	1.63E+005	2.14E+005	1.82E+004	*6.78E−203*	*3.92E−202*	*1.76E−201*	*0.00E+00*
f_2	30	4.17E+003	6.36E+003	8.93E+003	1.07E+003	*3.71E−204*	*9.25E−203*	*3.39E−201*	*0.00E+00*
	60	2.63E+004	2.45E+004	5.99E+004	5.52E+003	*1.70E−203*	*3.22E−201*	*1.52E−199*	*0.00E+00*
f_3	30	8.92E+002	4.57E+009	1.09E+011	1.67E+010	*2.28E−102*	*3.76E−102*	*1.09E−101*	*1.77E−102*
	60	1.36E+018	1.14E+025	2.96E+026	4.89E+025	*4.89E−102*	*7.44E−102*	*1.54E−101*	*2.09E−102*
f_4	30	3.96E+004	7.61E+004	1.15E+005	1.65E+004	*1.21E−202*	*4.95E−201*	*1.01E−199*	*0.00E+00*
	60	1.93E+005	3.07E+005	4.22E+005	5.00E+005	*1.48E−202*	*1.65E−200*	*2.73E−199*	*0.00E+00*
f_5	30	7.34E+001	9.19E+001	9.98E+001	6.98E+000	*1.29E−102*	*1.16E−101*	*4.35E−101*	*9.06E−102*
	60	9.41E+001	9.86E+001	9.99E+001	1.32E+000	*3.04E−102*	*2.37E−101*	*1.10E−100*	*2.37E−101*
f_6	30	1.09E+008	1.97E+008	3.77E+008	5.80E+007	*0.00E+00*	*1.86E−029*	*5.93E−029*	*1.94E−029*
	60	5.64E+008	9.31E+008	1.31E+009	1.74E+008	*0.00E+00*	*3.81E−029*	*1.53E−028*	*3.17E−029*
f_7	30	2.79E+004	4.71E+004	7.09E+004	9.69E+003	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
	60	1.16E+005	1.56E+005	1.93E+005	1.72E+004	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_8	30	3.48E+001	8.29E+001	1.45E+002	2.71E+001	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
	60	3.31E+002	8.29E+002	1.39E+003	2.03E+002	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_9	30	2.87E+001	8.81E+001	1.89E+002	3.39E+001	*2.00E−002*	*7.05E−002*	*1.55E−001*	*3.40E−002*
	60	5.14E+002	8.79E+002	1.32E+003	1.74E+002	*1.83E−002*	*5.55E−002*	*9.08E−002*	*1.58E−002*
f_{10}	30	2.79E−001	1.36E+000	4.47E+000	9.01E−001	*1.33E−212*	*8.37E−209*	*1.94E−207*	*0.00E+00*
	60	1.05E+000	3.42E+000	6.46E+000	1.24E+000	*2.68E−216*	*4.68E−208*	*1.24E−206*	*0.00E+00*
f_{11}	30	3.42E+002	4.11E+002	4.72E+002	2.69E+001	*29.8488*	*29.8488*	*29.8488*	*3.59E−014*
	60	8.28E+002	9.79E+002	1.15E+003	5.52E+001	*59.6975*	*59.6975*	*59.6975*	*7.18E−014*
f_{12}	30	1.96E+001	2.04E+001	2.09E+001	2.96E−001	*3.5745*	*3.5745*	*3.5745*	*1.65E−015*
	60	2.05E+001	2.09E+001	2.11E+001	1.12E−001	*3.5745*	*3.5745*	*3.5745*	*2.52E−015*
f_{13}	30	2.56E+002	4.62E+002	6.25E+002	6.54E+001	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
	60	1.12E+003	1.42E+003	1.88E+003	1.70E+002	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{14}	30	4.69E+002	7.15E+002	9.70E+002	1.09E+002	*0.00E+00*	*1.06E−031*	*2.96E−031*	*7.25E−032*
	60	1.78E+003	2.32E+003	3.18E+003	2.72E+002	*9.86E−032*	*4.44E−018*	*2.22E−016*	*3.14E−017*
f_{15}	30	4.86E+001	6.25E+001	7.83E+001	7.44E+000	*7.63E−017*	*1.17E−016*	*1.61E−016*	*1.88E−017*
	60	1.27E+002	1.51E+002	1.76E+002	1.09E+001	*1.87E−016*	*2.42E−016*	*3.99E−016*	*3.19E−017*
f_{16}	30	5.57E+000	8.08E+000	1.03E+001	9.41E−001	*2.55E−207*	*4.04E−205*	*1.86E−203*	*0.00E+00*
	60	1.73E+000	2.19E+000	2.53E+000	1.58E+000	*4.98E−207*	*1.25E−205*	*3.50E−204*	*0.00E+00*
f_{17}	30	3.26E+004	3.69E+005	9.01E+005	2.44E+005	*2.49E−205*	*7.05E−203*	*7.03E−202*	*0.00E+00*
	60	3.38E+005	5.81E+006	1.47E+007	2.71E+006	*3.23E−205*	*1.40E−201*	*2.97E−200*	*0.00E+00*
f_{18}	30	*1.41E+001*	1.45E+001	1.46E+001	8.45E−002	*1.40E+001*	*1.40E+001*	*1.40E+001*	*7.69E−011*
	60	*2.92E+001*	2.95E+001	2.99E+001	1.05E−001	*2.90E+001*	*2.90E+001*	*2.90E+001*	*1.84E−012*
f_{19}	30	5.69E+001	1.10E+002	1.72E+002	2.71E+001	*1.35E−032*	*1.19E−031*	*4.94E−031*	*9.72E−032*
	60	2.56E+002	3.62E+002	4.98E+002	5.53E+001	*1.35E−032*	*2.19E−031*	*5.93E−031*	*1.35E−031*
f_{20}	30	2.22E+008	4.27E+008	6.90E+008	1.06E+008	*8.31E−199*	*2.92E−197*	*7.56E−196*	*0.00E+00*
	60	1.45E+009	2.54E+009	3.99E+009	5.79E+008	*1.91E−198*	*3.77E−197*	*4.59E−196*	*0.00E+00*
f_{21}	30	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
	60	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*	*0.00E+00*
f_{22}	30	1.86E+001	2.35E+001	2.69E+001	1.95E+000	*4.99E−001*	*4.99E−001*	*4.99E−001*	*3.88E−016*
	60	3.58E+001	4.07E+001	4.47E+001	2.13E+000	*1.2776*	*1.4578*	*1.4999*	*6.53E−002*
f_{23}	30	4.99E−001	4.99E−001	4.99E−001	*3.42E−005*	*4.92E−001*	*4.95E−001*	*4.96E−001*	1.40E−003
	60	5.00E−001	5.00E−001	5.00E−001	*2.89E−006*	*4.67E−001*	*4.79E−001*	*4.83E−001*	4.40E−003

dimensions in Figure 2. All the convergence curves are plotted by the mean result achieved in the 50 runs. In the figure, the number of iterations is indicated on the horizontal axis and the vertical axis represents the mean of logarithms (log) of objective function values. As displayed in Figure 2, the DDS-POBL algorithm has faster convergence than the classical DDS algorithm for all of the benchmark test functions. This means that the guide on the search direction of algorithm is right and that the boundaries updated by the GS can make the search lead to more promising area.

To better evaluate single solution-based search algorithms, we calculate the *acceleration rate* (AR) and *success*

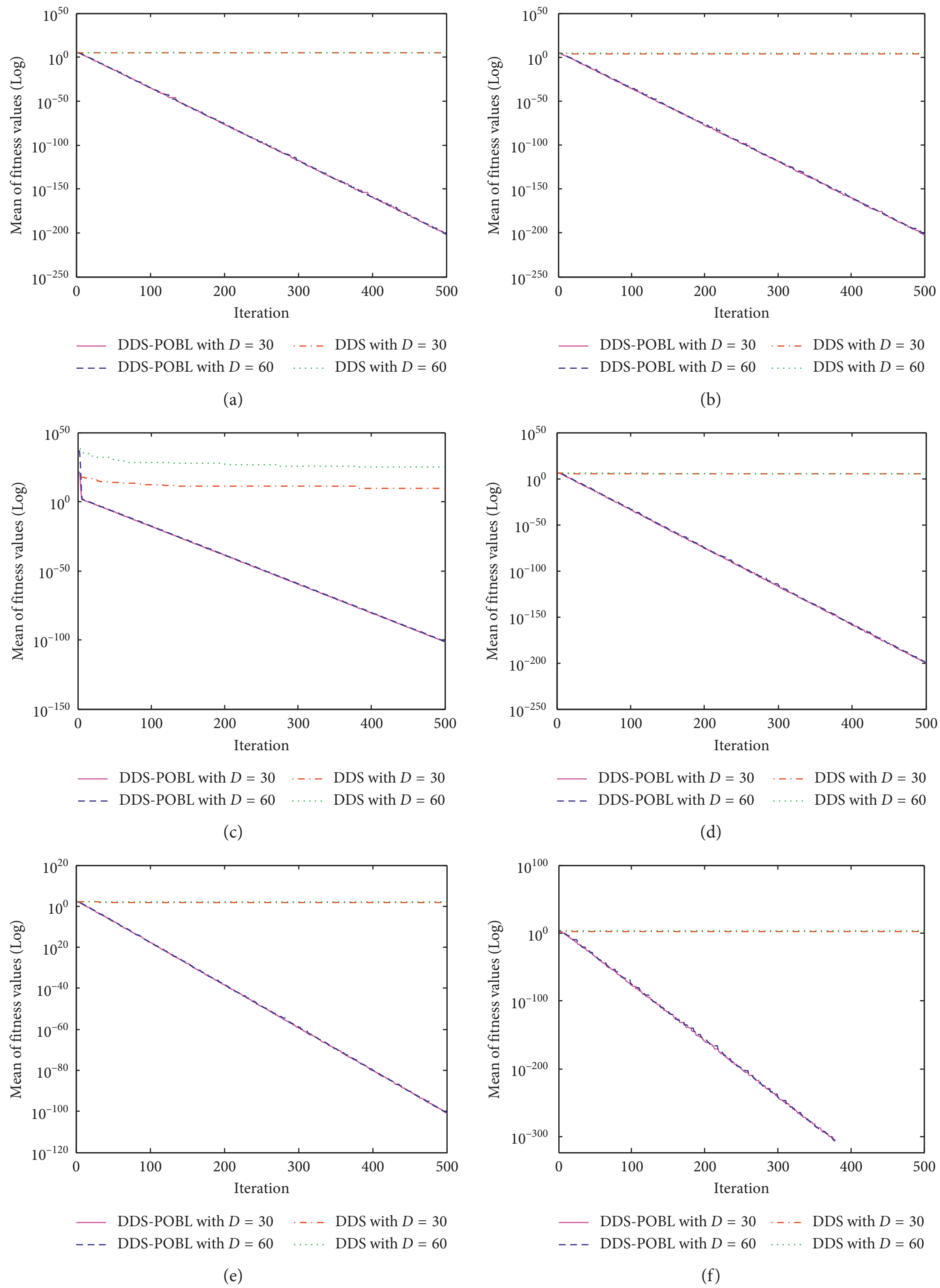


Figure 2: Continued.

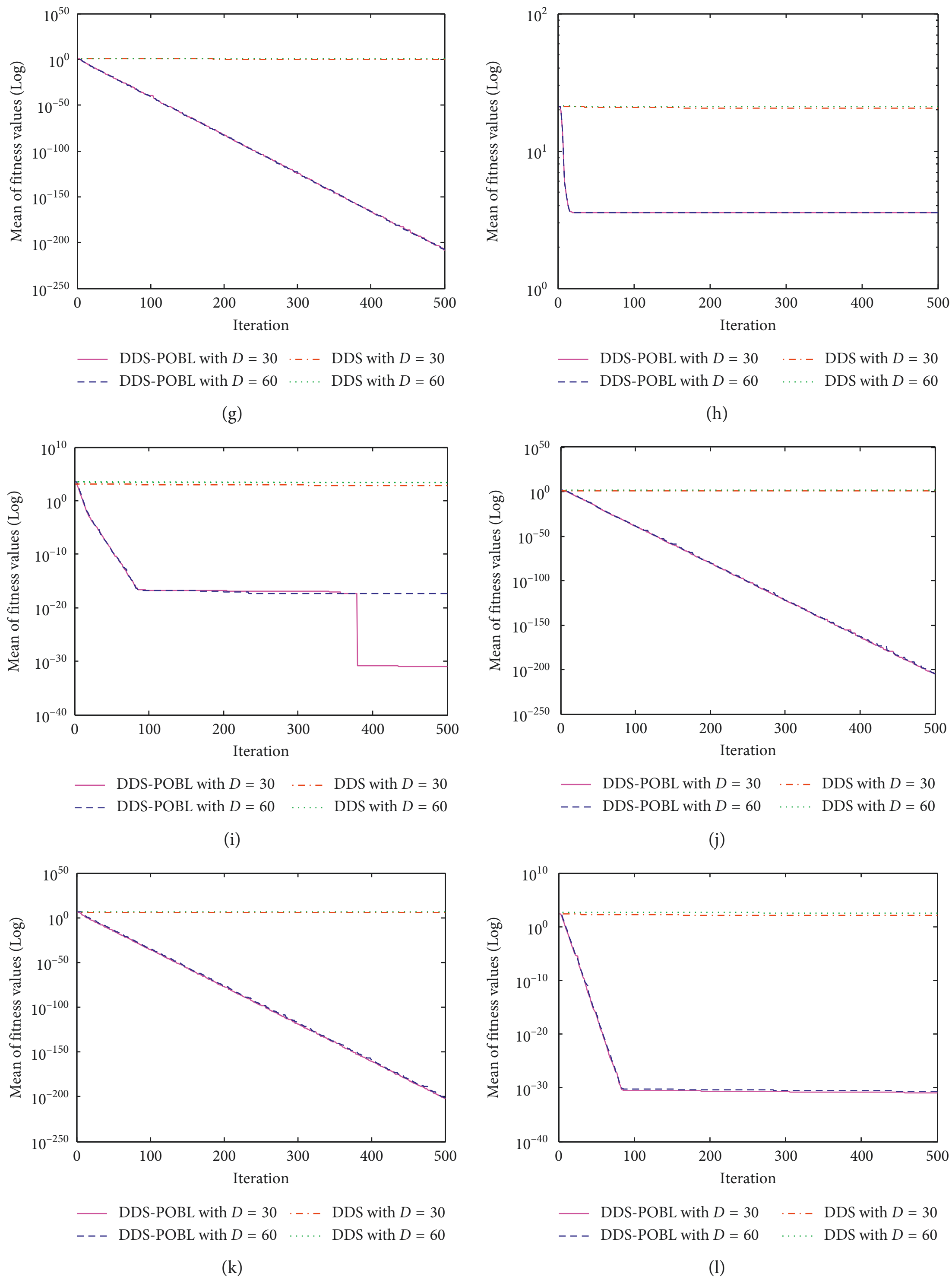
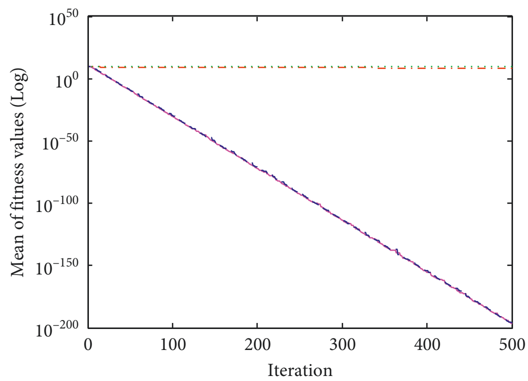
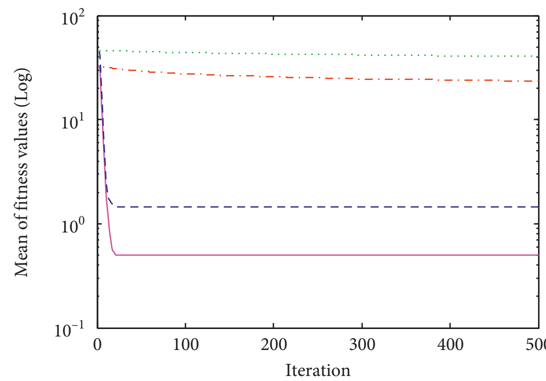


Figure 2: Continued.



(m)



(n)

Figure 2: Convergence curves for $D = 30$ and $D = 60$ of certain typical functions. (a) Function f1. (b) Function f2. (c) Function f3. (d) Function f4. (e) Function f5. (f) Function f8. (g) Function f10. (h) Function f12. (i) Function f14. (j) Function f16. (k) Function f17. (l) Function f19. (m) Function f20. (n) Function f22.

Table 3: Comparison results of acceleration rates and success rates between the DDS-POBL algorithm and DDS algorithm.

Function	Success rate (*SR*) DDS-POBL		DDS		Acceleration rate ($AR_{DDS-POBL}$)	
	Dim = 30	Dim = 60	Dim = 30	Dim = 60	Dim = 30	Dim = 60
f_1	1	1	0	0	5.68	5.62
f_2	1	1	0	0	5.75	5.60
f_3	1	1	0	0	3.09	3.06
f_4	1	1	0	0	5.44	5.33
f_5	1	1	0	0	3.07	3.05
f_6	1	1	0	0	1.25	1.10
f_7	1	1	0	0	5.58	5.53
f_8	1	1	0	0	11.51	11.23
f_9	0	0	0	0	1	1
f_{10}	1	1	0	0	6.82	6.81
f_{11}	0	0	0	0	1	1
f_{12}	0	0	0	0	1	1
f_{13}	1	1	0	0	9.32	9.20
f_{14}	1	0.94	0	0	1.27	1.02
f_{15}	0	0	0	0	1	1
f_{16}	1	1	0	0	6.26	6.19
f_{17}	1	1	0	0	5.73	5.61
f_{18}	0	0	0	0	1	1
f_{19}	1	1	0	0	1.26	1.05
f_{20}	1	1	0	0	4.96	4.90
f_{21}	1	1	1	1	1	1
f_{22}	0	0	0	0	1	1
f_{23}	0	0	0	0	1	1
Average	0.7391	0.7365	0.0435	0.0435	3.70	3.62

rate (SR) (as described in Section 4.1) for several benchmark functions. These results are recorded in Table 3. As described in Table 3, the proposed DDS-POBL algorithm converges to the desired result faster than the conventional DDS algorithm for 15 out of 23 test functions. For the other 8 test functions, the convergence speed of the DDS-POBL algorithm is equal to that of the DDS algorithm. From the value of metrics of the *average acceleration rate* (AR_{ave}) in Table 3, it can be observed that the DDS-POBL algorithm has a more superior average convergence rate than DDS. Overall, in terms of achieving global optimum and avoiding local optima, the DDS-POBL algorithm outperforms DDS.

4.4. Comparison with Other State-of-the Art Algorithm. To further show the significant superiority of DDS-POBL, we compared it with eight state-of-the-art optimization algorithms, i.e., the particle swarm optimizer (PSO) [28], the piecewise opposition-based learning (POHS) [20], the completely derandomized self-adaptation evolution strategies (CMA-ES) [29], the composite differential evolution (CoDE) [30], the memory-based hybrid dragonfly algorithm (MHDA) [28], the exploration-enhanced grey wolf optimizer (EEGWO) [24], the hybrid method based on the DA and the improved NM simplex algorithm (INMDA) [13], and the firefly algorithm with neighborhood attraction (NaFA) [31]. In this experiment, the parameter settings of EEGWO, INMDA, and DDS-POBL are as follows: the population size is 30, the maximum number of iterations is 500, the number of independent experiments is 50, and the other parameters related to the algorithm are consistent with its original literature. Since the function values for each test function of the POHS, CMA-ES, MHDA, and NaFA algorithm were taken directly from their original papers, the parameter settings of these algorithms remain the same as the original papers.

In this experiment, the best fitness values are averaged (represented by "Mean"), and the corresponding standard deviation values (indicated by "St. dev") are computed. Table 4 records the experimental results obtained by DDS-POBL and other eight algorithms for $D = 30$. Please note that the best results achieved for each test function is highlighted in italic.

The comparison results presented in Table 4 show that the proposed DDS-POBL algorithm yields better results than PSO in 7 test functions ($f_1, f_3, f_4, f_5, f_6, f_9$, and f_{13}) except for 2 test functions (f_{11} and f_{12}). In terms of functions f_{11} and f_{12}, POHS is able to achieve better results than DDS-POBL, but DDS-POBL achieves better results on f_1, f_5, f_6, and f_{13}. The DDS-POBL algorithm provides better optimization values than the CMA-ES algorithm for 8 out of 9 benchmark test functions, while CMA-ES provides better "Mean" result on function f_{11}. The DDS-POBL algorithm shows poorer optimization results than CoDE only in function f_{12} for 9 test problems. With respect to MHDA, DDS-POBL obtains better "Mean" and "St. dev" results on functions $f_1, f_3, f_4, f_5, f_6, f_9$, and f_{13}, but MHDA has better performance in functions f_{11} and f_{12}. In terms of functions f_{11} and f_{12}, POHS is able to achieve better results than DDS-POBL, and DDS-POBL achieves better results on f_1, f_5, f_6, and f_{13}. When compared to INMDA, DDS-POBL provides very competitive results on all functions except functions f_9, f_{11}, and f_{12}. INMDA gets better results than DDS-POBL on functions f_{11} and f_{12}, on the contrary, worse on f_9. NaFA obtains better results than DDS-POBL on functions f_{11} and f_{12} for "Mean" value. Compared with EEGWO, DDS-POBL gets competitive results for five functions (f_1, f_3, f_4, f_5, and f_9) and similar for one function (f_{13}). In addition, DDS-POBL obtains better results than EEGWO for one function (f_6) but worse for two functions (f_{11} and f_{12}).

For the Rastrigin function, not all optimization algorithms can converge to the global optimal value of zero. This can be easily found in the existing literature. For example, the POHS algorithm in literature [20] has an optimized value of $2.08E + 01$ for the Rastrigin function, while the optimized value is $5.90E - 07$ when using the MHDA in literature [28]. In the literature [31], the optimized value obtained by NaFA is $2.09E + 01$ for Rastrigin function and that by CoDE [30] is $3.41E + 01$. In this paper, the optimal solution of Rastrigin function obtained by the DDS and DDS-POBL algorithm is not a global solution. Although the value 29.8488 obtained by DDS-POBL is smaller than that by CoDE, it is still far from the zero. Maybe this is one of the drawbacks of the DDS and DDS-POBL algorithm.

As can be seen from Table 4, the optimization performance of DDS-POBL is very close to that of EEGWO and INMDA, and it is difficult to determine which algorithm is better. Therefore, the comparison of algorithms should be done using some statistical analysis. To conduct statistical analysis of the comparison results scientifically, we adopted two statistical methods, namely, the sign test and Wilcoxon signed-rank test, which are taken from reference [32]. The statistical analysis results are given in Tables 5 and 6. In the sign test shown in Table 5, DDS-POBL shows a significant improvement over CMA-ES and CoDE with a level of significance $\alpha = 0.05$ and over PSO and MHDA with a level of significance $\alpha = 0.1$. From the Wilcoxon signed-rank test results recorded in Table 6, we can see that DDS-POBL outperforms CMA-ES with a level of significance $\alpha = 0.01$ and outperforms CoDE, NaFA, and MHDA with $\alpha = 0.1$. In addition, DDS-POBL is inferior to INMDA, with a level of significance $\alpha = 0.05$, not significantly better than PSO and not significantly worse than EEGWO.

In order to reveal which algorithm reaches to vicinity of global solution fast, Figure 3 displays the convergence curves of DDS-POBL, EEGWO, and INMDA on functions f_1, f_5, f_{11}, and f_{13}. As can be seen in Figure 3, DDS-POBL and EEGWO showed almost the same convergence rate and both were inferior to INMDA on functions f_1 and f_5. On function f_{11}, EEGWO presented the fastest convergence rate, followed by INMDA, and DDS-POBL was the worst. For function f_{13}, DDS-POBL and EEGWO showed the fastest convergence rate, while INMDA was slower than the former two, but all of them converged to the global optimal.

4.5. Robustness of the Proposed Algorithm. For the DDS algorithm, performance is often determined by a scalar neighborhood size perturbation factor r. The author of paper [7] suggested that the optimization performance of DDS is best when $r = 0.2$. In the present work, in order to investigate the effect of different values of r on the performance of the DDS-POBL algorithm, the parameter r is varied at five different values i.e. 0.1, 0.2, 0.3, 0.5, and 0.9, with the rest of the parameters kept the same as mentioned in the previous subsection. All of the 23 test functions listed in Table 1 were run 20 times independently. Table 7 records the experimental results obtained by DDS-POBL with five different values of a scalar neighborhood size perturbation factor r for $D = 30$, where "Mean" indicates the mean best objective function value and "St. dev" represents the corresponding standard deviation value. To quickly recognize the best results, the best results for each function are marked in italic.

As it is described in Table 7, the total optimization performance of the DDS-POBL algorithm with $r = 0.2$ was superior to other cases. The specific optimization results of DDS-POBL are as follows: when r is 0.1 and 0.5, 8 optimal results are obtained; when r is 0.2, 13 optimal results are obtained; when r is 0.3, 7 optimal results are obtained; and when r is 0.9, 6 optimal results are obtained. However, the optimization results of different values of r are not significantly different

Table 4: Comparison of the DDS-POBL algorithm with other state-of-the-art algorithms ($D = 30$).

Function	Index	PSO	POHS	CMA-ES	CoDE	EEGWO	MHDA	INMDA	NaFA	DDS-POBL
f_1	Mean	2.70E − 09	1.39E − 23	6.15E − 29	2.36E − 10	1.58E − 203	4.07E − 42	0	4.43E − 29	1.74E − 202
	St. dev	1.00E − 09	5.10E − 01	1.72E − 29	1.69E − 10	0.00E + 00	2.22E − 41	0	4.06E − 30	0.00E + 00
f_3	Mean	7.15E − 05	—	1.8127	6.64E − 06	1.11E − 102	6.62E − 15	0	2.98E − 15	3.76E − 102
	St. dev	2.26E − 05	—	6.9028	3.41E − 06	8.56E − 103	3.61E − 14	0	2.80E − 15	1.77E − 102
f_4	Mean	4.71E − 06	—	3.12E − 27	8.10E − 01	2.54E − 200	2.55E − 50	0	2.60E − 28	4.95E − 201
	St. dev	1.49E − 06	—	8.17E − 28	5.39E − 01	0.00E − 00	1.30E − 49	0	2.37E − 29	0.00E + 00
f_5	Mean	3.25E − 07	5.49E − 21	3.84E − 15	2.95E − 02	2.10E − 102	4.99E − 05	0	3.43E − 15	1.16E − 101
	St. dev	1.02E − 08	8.5E − 01	4.93E − 16	1.10E − 02	2.64E + 102	2.73E − 04	0	2.89E − 17	9.06E − 102
f_6	Mean	1.23E − 01	5.04E + 03	0.7125	2.79E + 01	2.89E + 01	3.34E − 22	0	2.39E + 01	1.86E − 029
	St. dev	2.16E − 01	2.50E + 01	1.9802	1.75E + 01	2.42E − 02	5.67E − 22	0	8.96E − 01	1.94E − 029
f_9	Mean	1.39E + 00	—	0.2101	1.38E − 02	5.96E − 05	5.25E − 05	7.77E − 05	2.91E − 02	8.37E − 209
	St. dev	0.001269	—	0.0555	5.70E − 03	4.72E − 05	5.02E − 05	2.92E − 05	1.56E − 02	0.00E + 00
f_{11}	Mean	2.78E − 01	2.08E + 01	2.32E + 02	3.41E + 01	0.00E + 00	5.90E − 07	0	2.09E + 01	29.8488
	St. dev	2.18E − 01	0.90E + 01	5.54E + 01	5.70E + 00	0.00E + 00	3.23E − 06	0	6.96E + 00	3.59E − 014
f_{12}	Mean	1.11E − 09	0.15E + 01	19.4830	3.89E − 06	8.88E − 16	6.34E − 15	8.88E − 16	3.02E − 14	3.5745
	St. dev	2.39E − 11	7.80E − 02	0.1369	1.51E − 06	0.00E + 00	2.72E − 14	0.00E + 00	8.94E − 15	1.65E − 015
f_{13}	Mean	2.73E − 01	6.76E − 01	1.40E − 03	5.14E − 05	0.00E + 00	2.40E − 04	0	0.00E + 00	0.00E + 00
	St. dev	2.04E − 01	0.60E − 02	3.30E − 03	2.81E − 04	0.00E + 00	2.25E − 02	0	0.00E + 00	0.00E + 00

Table 5: Summary of sign test results on six state-of-the-art algorithms.

DDS-POBL	PSO	POHS	CMA-ES	CoDE	EEGWO	MHDA	INMDA	NaFA
Wins (+)	7	≥ 4	9	8	2	7	1	6
Approximates (≈)	0	NaN	0	0	5	0	5	1
Loses (_)	2	NaN	0	1	2	2	3	2
Detected differences	$\alpha = 0.1$	NaN	$\alpha = 0.05$	$\alpha = 0.05$	–	$\alpha = 0.1$	–	–

Note. The symbol “NaN” in this paper indicates the vacancy value, which cannot be calculated since the corresponding value in the reference is missing.

Table 6: Wilcoxon signed-rank test results on six state-of-the-art algorithms.

Comparison	R^+	R^-	p value	Comparison	R^+	R^-	p value
DDS-POBL versus PSO	28	17	0.5703	DDS-POBL versus POHS	NaN	NaN	NaN
DDS-POBL versus CMA-ES	45	0	0.007	DDS-POBL versus CoDE	38	7	0.0742
DDS-POBL versus EEGWO	13	23	0.5469	DDS-POBL versus INMDA	3	33	0.0391
DDS-POBL versus MHDA	42	3	0.0195	DDS-POBL versus NaFA	31	5	0.0781

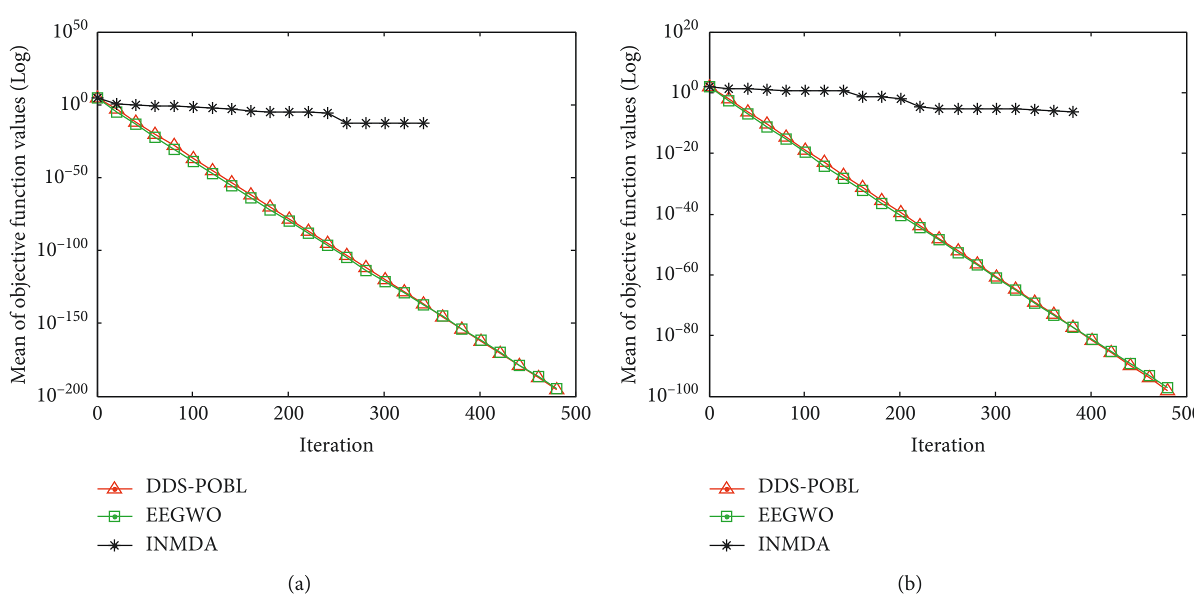


Figure 3: Continued.

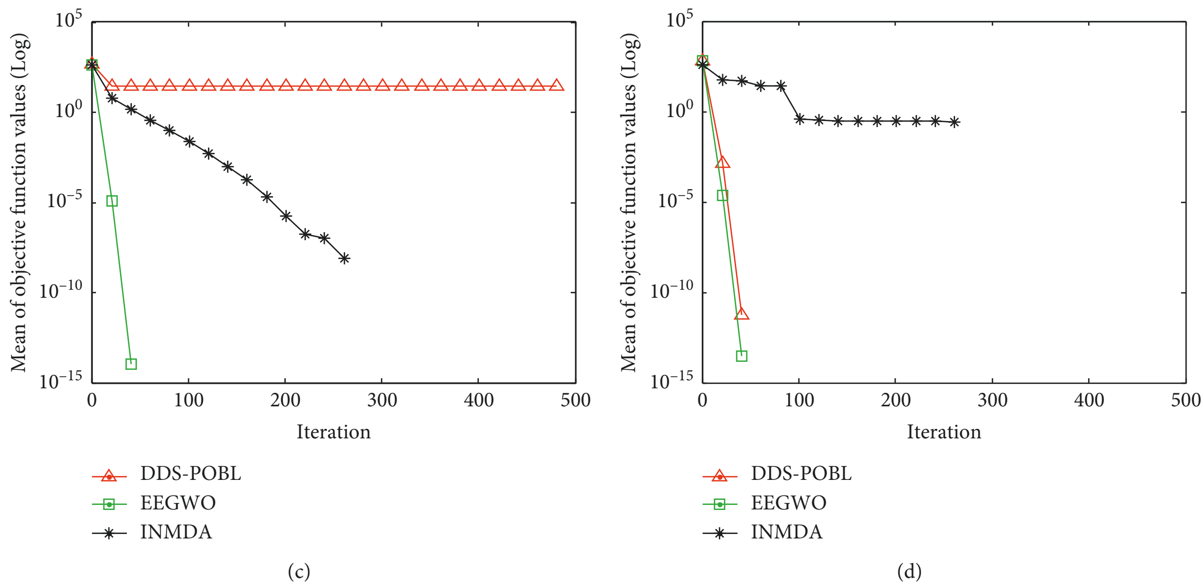


Figure 3: Convergence curves of DDS-POBL, EEGWO, and INMDA with $D = 30$. (a) Function f1. (b) Function f5. (c) Function f11. (d) Function f13.

except for one function (f_{13}). Figure 4 clearly shows the average convergence speed and the performance of the DDS-POBL algorithm when different r values are taken. Therefore, the proposed DDS-POBL algorithm has good robustness.

4.6. Scalability of the Proposed Algorithm. To further investigate the scalability and optimization performance of the proposed algorithm in high dimensional space, experiments are repeated on some typical test functions (f_1, f_2, f_5–f_8, f_{10}, f_{13}, f_{16}, f_{17}, and f_{20}–f_{21}) with dimensions 100, 300, and 500. Each test function is run 30 times independently, and the rest of the parameters are the same setting as Section 4.2. Table 8 records this experiment results.

As can be seen from Table 8, the greatly increased dimension of the test functions dose not result in degrade of optimization performance of the DDS-POBL algorithm too much. Although the optimization performance of the DDS-POBL algorithm can be reduced as the dimension increases on several test functions (i.e., f_1, f_2, f_5, f_6, f_{10}, f_{16}, f_{17}, and f_{20}), the reduction is too small to be ignored. In addition, the optimization performance of DDS-POBL on several test functions (f_7, f_8, f_{13}, and f_{21}) does not change with increasing dimension, and all of them obtain global optima 0.

4.7. Application of the Proposed Method to Parameter Calibration of Xinanjiang Model (Day Model). The Xinanjiang model is a well-known hydrological model put forward by Zhao R. J. of Ho-hai University, which is a concept with decentralized parameters watershed hydrological model. Its detailed information can be found in reference [33]. There are many studies on parameter optimization of this model, mainly including the surrogate modeling approach [34], genetic algorithm (GA) [35], and SEC-UA [36]. In this subsection, we choose this model as a real optimization problem to test the efficiency of the proposed method and select the Yanduhe catchment of the Three Gorges, with a drainage area of 601 km^2 [37] as the study area. This area has a humid climate, good vegetation, and loose soil and is divided into 59 basic units, including 30 outer units and 29 inner units. The area, chain length, and slope length of the inner and outer units, respectively, are surveyed, and the results are shown in Table 9. The historical runoff data used was from January 1, 1981, to December 30, 1981.

The parameters of the Xinanjiang model are divided into four categories, including 15 parameters [31]. The first category "evapotranspiration parameters" includes K, WUM, WLM, and C. The second category "runoff production parameters" consists of WM, B, and IMP. The third category "parameters of runoff separation" contains SM, EX, KSS, and KG. The last category "runoff concentration parameters" includes KKSS, KKG, CS, and L. Among these parameters, WM, WUM, WLM, B, C, and IMP are insensitive parameters and are generally valued by experience, and their value is recorded in Table 10. On the contrary, the other 9 parameters are sensitive parameters. In these sensitive parameters, K, EX, and L do not need to be calibrated and are determined by experience as shown in Table 10. The other six sensitive parameters need to be calibrated, and their range is described in Table 11. To further verify the effectiveness of the proposed algorithm, we used the DDS-POBL and DDS to calibrate these six sensitive parameters. The results are shown in Table 11. Table 12 listed the proportion of different intervals of the runoff calculation relative error. Figure 5 plotted the fit of the calculation flow and observed flow of the DDS-POBL and DDS algorithm after calibrating the nine sensitive parameters.

It can be seen from Table 12 that, in the runoff calculation, relative error intervals are [0, 20] and (30,100], the proportion of the DDS is larger than DDS-POBL, while in the interval (0, 30], DDS-POBL is much larger than DDS. In addition, the proportion of DDS-POBL in runoff calculation

Table 7: Descriptive results of DDS-POBL using five different r values for 23 functions ($D = 30$).

Function	$r = 0.1$		$r = 0.2$		$r = 0.3$		$r = 0.5$		$r = 0.9$	
	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev	Mean	St. dev
f_1	1.95E − 201	0.00E − 00	*1.74E − 202*	*0.00E − 00*	9.21E − 202	0.00E − 00	1.38E − 200	0.00E − 00	2.51E − 201	0.00E − 00
f_2	1.38E − 202	0.00E − 00	9.25E − 203	0.00E − 00	1.65E − 202	0.00E − 00	*8.47E − 203*	*0.00E − 00*	4.77E − 202	0.00E − 00
f_3	4.99E − 102	6.43E − 102	*3.76E − 102*	*1.77E − 102*	4.10E − 102	1.79E − 102	6.90E − 102	1.49E − 101	6.34E − 102	4.81E − 102
f_4	3.65E − 201	0.00E − 00	4.95E − 201	0.00E − 00	*2.64E − 201*	*0.00E − 00*	1.30E − 200	0.00E − 00	1.01E − 200	0.00E − 00
f_5	2.49E − 101	7.94E − 101	*1.16E − 101*	*9.06E − 102*	1.35E − 101	1.45E − 101	1.22E − 101	1.59E − 101	1.88E − 101	2.11E − 101
f_6	6.11E − 029	3.27E − 029	1.86E − 029	1.94E − 029	*2.17E − 030*	*6.86E − 030*	3.53E − 029	1.84E − 029	8.85E − 029	1.89E − 029
f_7	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*
f_8	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*
f_9	7.18E − 002	3.40E − 002	7.05E − 002	3.40E − 002	7.67E − 002	3.28E − 002	*6.50E − 002*	*2.97E − 002*	6.73E − 002	2.68E − 002
f_{10}	*5.55E − 209*	*0.00E − 00*	8.37E − 209	0.00E + 00	1.58E − 208	0.00E − 00	3.22E − 208	0.00E − 00	8.55E − 208	0.00E − 00
f_{11}	29.8488	3.92E − 014	*29.8488*	*3.59E − 014*	29.8488	3.92E − 014	*29.8488*	*3.59E − 014*	*29.8488*	*3.59E − 014*
f_{12}	3.5745	1.79E − 015	*3.5745*	1.65E − 015	3.5745	1.79E − 015	3.5745	1.79E − 015	3.5745	2.25E − 015
f_{13}	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*
f_{14}	1.33E − 017	5.33E − 017	1.06E − 031	7.25E − 032	*7.89E − 033*	*2.08E − 032*	8.48E − 032	4.46E − 032	2.88E − 031	7.48E − 032
f_{15}	*1.12E − 016*	*1.86E − 017*	1.17E − 016	1.88E − 017	1.18E − 016	2.22E − 017	1.28E − 016	2.09E − 017	1.45E − 016	2.47E − 017
f_{16}	7.38E − 206	0.00E + 00	4.04E − 205	0.00E + 00	4.58E − 206	0.00E + 00	*1.71E − 206*	*0.00E + 00*	9.90E − 206	0.00E + 00
f_{17}	8.65E − 202	0.00E + 00	*7.05E − 203*	*0.00E + 00*	6.79E − 202	0.00E + 00	3.64E − 202	0.00E + 00	4.21E − 202	0.00E + 00
f_{18}	1.40E + 01	3.63E − 013	1.40E + 01	7.69E − 011	1.40E + 01	1.29E − 010	1.40E + 01	3.47E − 011	*1.40E + 01*	*3.09E − 013*
f_{19}	3.52E − 031	1.80E − 031	*1.19E − 031*	*9.72E − 032*	2.04E − 032	1.99E − 032	9.53E − 032	4.94E − 032	2.57E − 031	6.33E − 032
f_{20}	6.39E − 197	0.00E + 00	*2.92E − 197*	*0.00E + 00*	8.58E − 198	0.00E + 00	2.50E − 197	0.00E + 00	4.45E − 197	0.00E + 00
f_{21}	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*	*0.00E + 00*
f_{22}	*4.96E − 001*	*1.83E − 002*	*4.99E − 001*	3.88E − 016	4.99E − 001	3.84E − 016	4.99E − 001	3.88E − 016	4.99E − 001	3.87E − 016
f_{23}	*0.4996*	*3.44E − 04*	0.4999	2.51E − 05	0.4999	1.80E − 05	0.4999	1.55E − 05	0.4999	2.07E − 05

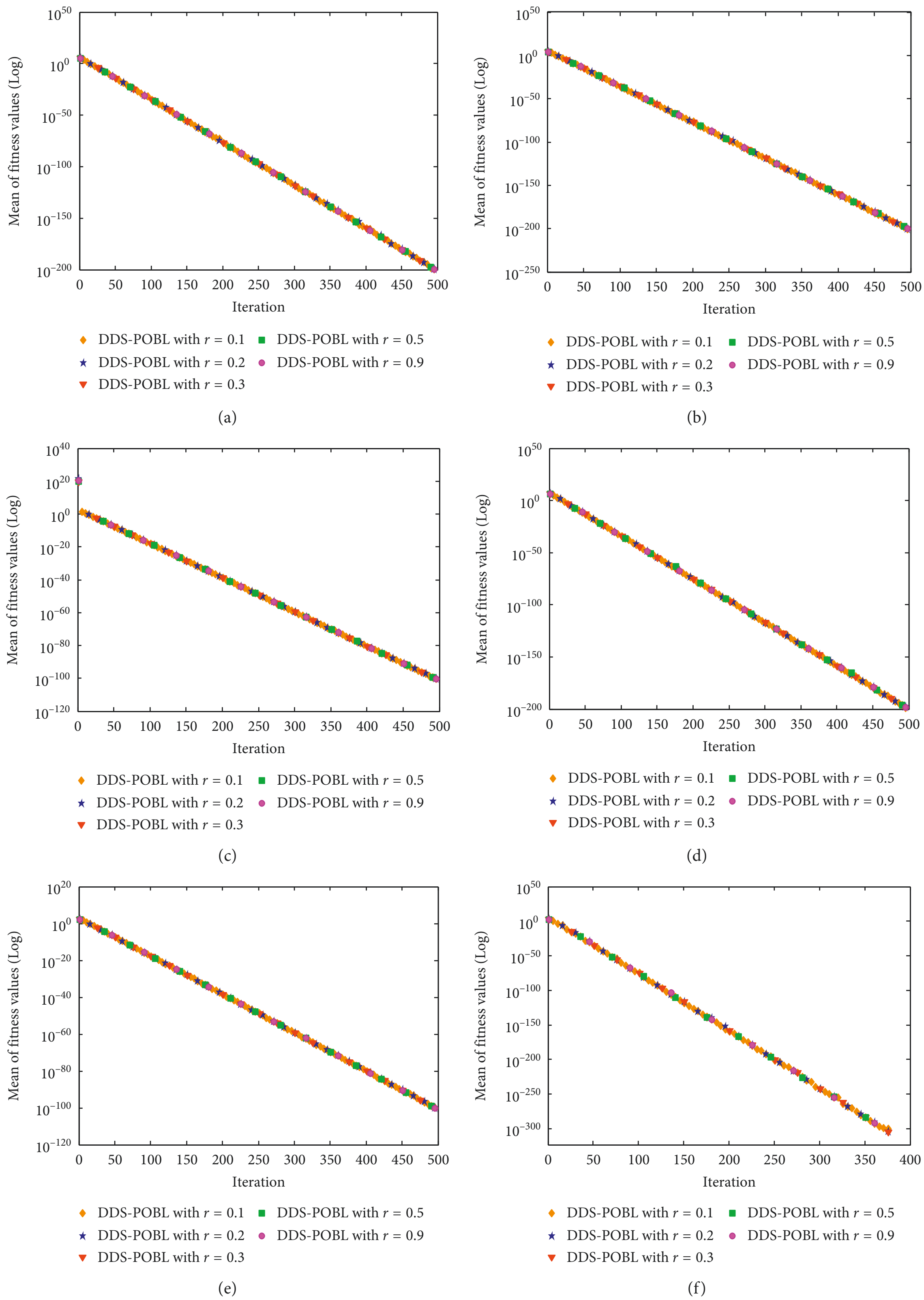


Figure 4: Continued.

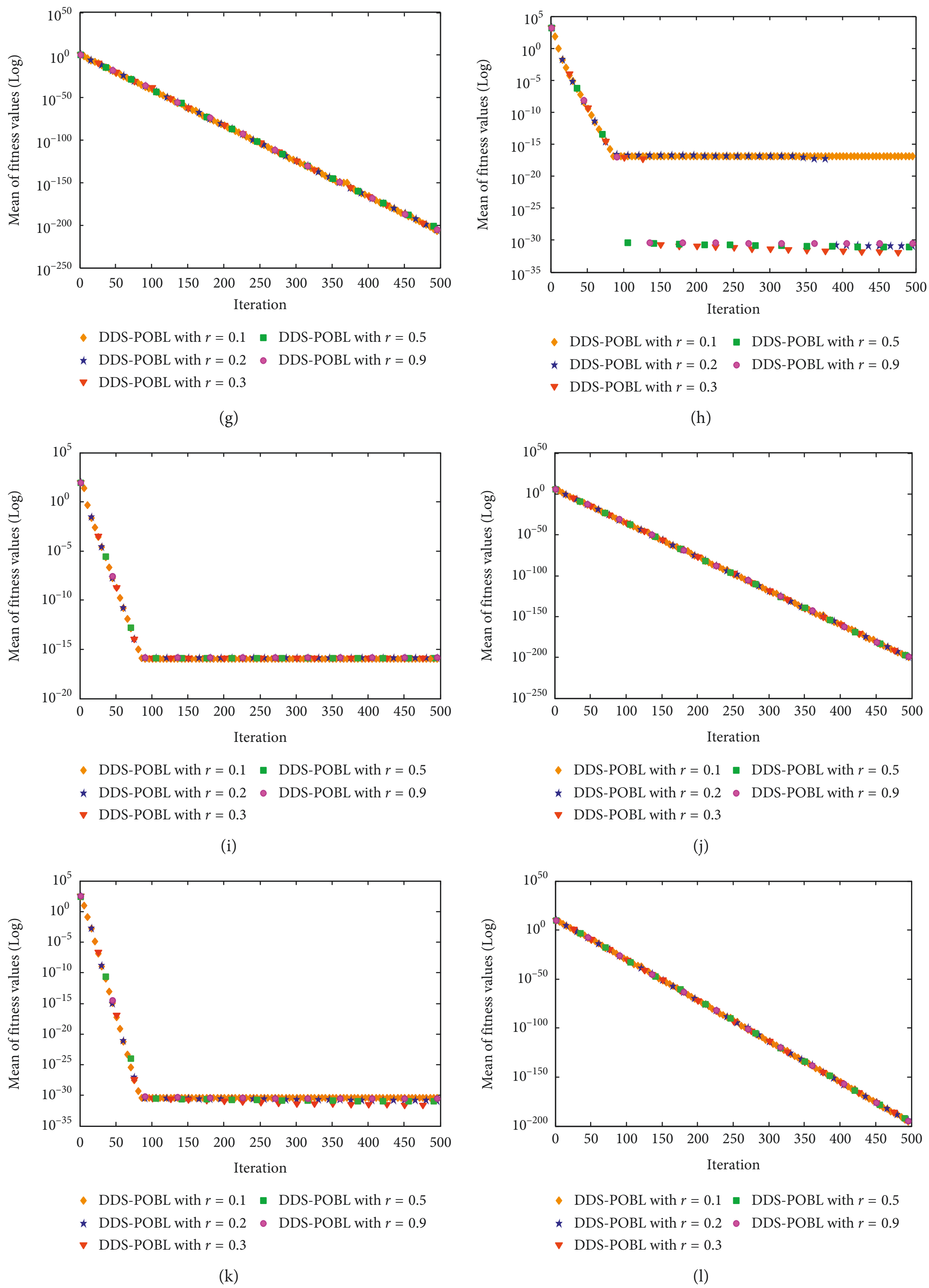


Figure 4: Convergence curves for $D = 30$ with different r values on some typical functions. (a) f1. (b) f2. (c) f3. (d) f4. (e) f5. (f) f8. (g) f10. (h) f14. (i) f15. (j) f17. (k) f9. (l) f20.

relative error interval (20, 30] is greater than the sum of DDS in the interval [0, 50]. From Figure 5, DDS-POBL has a slightly poor fitting effect compared to DDS during the relatively smooth period of runoff variation, but the DDS fitting effect is significantly inferior to DDS-POBL when the runoff changes drastically.

Table 8: Evaluation of the performance of the DDS-POBL algorithm over 30 independent runs for dimensions 100, 300, and 500.

Function	Dim = 100		Dim = 300		Dim = 500	
	Mean	St. dev	Mean	St. dev	Mean	St. dev
f_1	*2.33E − 201*	*0.00E + 000*	5.81E − 201	0.00E + 000	3.86E − 200	0.00E + 000
f_2	*6.77E − 202*	*0.00E + 000*	4.33E − 201	0.00E + 000	1.21E − 200	0.00E + 000
f_5	*3.01E − 101*	*4.05E − 101*	9.65E − 101	1.34E − 100	6.75E − 101	5.44E − 101
f_6	*8.86E − 029*	*4.97E − 029*	3.83E − 028	1.52E − 028	7.97E − 028	3.36E − 028
f_7	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*
f_8	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*
f_{10}	7.83E − 208	0.00E + 000	*6.62E − 209*	*0.00E + 000*	2.32E − 208	0.00E + 000
f_{13}	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*
f_{16}	*6.16E − 205*	*0.00E + 000*	7.11E − 205	0.00E + 000	7.51E − 205	0.00E + 000
f_{17}	*3.81E − 201*	*0.00E + 000*	1.07E − 199	0.00E + 000	1.03E − 198	0.00E + 000
f_{20}	*2.92E − 197*	*0.00E + 000*	1.83E − 196	0.00E + 000	2.88E − 195	0.00E + 000
f_{21}	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*	*0.00E + 000*

Table 9: Result of basic cell survey.

Unit	Amount	Average area (km^2)	Average chain length (km^2)	Average slope length (km)
Inner unit	29	6.081	2.846	1.069
Outer unit	30	14.164	3.118	1.136

Table 10: Empirical values of 6 parameters.

Parameter	Definition	Empirical value
WM	Areal mean tension water capacity	120
WUM	Tension water capacity of the upper layer	20
WLM	Tension water capacity of the lower layer	60
K	Ratio of potential evapotranspiration to pan evaporation	0.55
B	Exponent of the tension water capacity curve	0.15
C	Coefficient of deep evapotranspiration	0.15
IMP	Ratio of the impervious to the total area of the basin	0.01
EX	Exponent of the free water capacity curve influences the development of the saturated area	2.2
L	Corresponding "lag"	0

Table 11: Calibration results for 6 sensitive parameters.

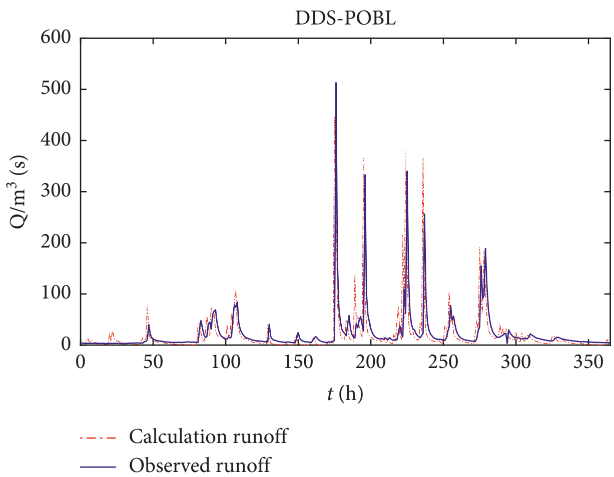
Parameter	Definition	DDS	DDS-POBL	Range
SM	Areal mean of the free water capacity of the surface soil layer	30.561	50	[10, 50]
KSS	Outflow coefficients of the free water storage to groundwater and interflow relationships.	0.39	0.382	[0.3, 0.4]
KG	Outflow coefficients of the free water storage to groundwater and interflow relationships.	0.491	0.397	[0.3, 0.5]
CS	Recession constant in the "lag and route" method for routing through the channel system within each sub-basin	0.655	0.63	[0.2, 0.7]
KKSS	Recession constant of the lower interflow storage	0.8	0.85	[0.8, 0.9]
KKG	Recession constant of groundwater storage	0.998	0.91	[0.9 0.999]

5. Conclusions

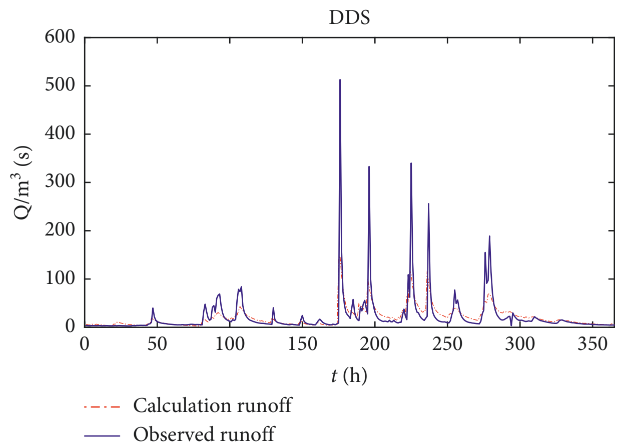
An improved version of DDS combined with hybrid piecewise opposition-based learning, called DDS-POBL, has been proposed in this work. Compared with DDS, DDS-PBOL has been significantly improved in the two following aspects. One is to introduce the piecewise oppositional learning strategies to help the DDS algorithm to search the correct direction; the other is to use the golden section method to guide potential solutions to search more promising areas. Several different numerical experiments were performed to verify the advantages of the proposed

Table 12: The proportion of different intervals of runoff calculation relative error.

Runoff calculation relative error interval (%)		[0, 20]	(20, 30]	(30, 40]	(40, 50]	(50, 100]
Proportion (%)	DDS	30.14	15.07	7.40	10.41	36.99
	DDS-POBL	7.39	77.53	3.29	3.84	7.95



(a)



(b)

Figure 5: Fitting of calculation runoff and observed runoff of DDS-POBL and DDS.

approach in optimizing performance. Firstly, tests were done on 23 benchmark test functions. The simulation results showed that on the 15 out of 23 test functions, DDS-POBL had better optimization performance than traditional DDS. Secondly, nine typical test functions were chosen to verify the performance of DDS-POBL compared to other state-of-the-art algorithms. Experimental results revealed that the proposed DDS-POBL algorithm could offer highly competitive results compared to other state-of-the-art algorithms in most cases. Thirdly, an experiment on a scalar neighborhood size perturbation factor was performed with different values of DDS-POBL to investigate its robustness. The optimization results showed that the proposed DDS-POBL algorithm could provide optimization results with little difference. Furthermore, the parameter calibration application of the Xinanjiang model reveals the superiority of DDS-POBL over DDS in practical optimization problems.

In addition, several representative large-scale test functions were selected as experimental objects to verify the scalability of the DDS-POBL algorithm, and the results of large-scale test problems proved that the proposed method has good scalability. However, for functions f_{11} and f_{12}, the proposed DDS-POBL algorithm could not find a satisfactory result. Therefore, the proposed DDS-POBL algorithm is still not an ideal algorithm. In our future work, the exciting research and applications can be explored further.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (Grant no. 71841054), Soft Science Foundation of Heilongjiang Province (Grant no. GC16D102), Key Program of Economic and Social of Heilongjiang Province (Grant no. KY10900170004), Philosophy and Social Science Research Planning Program of Heilongjiang Province (Grant no. 17JYH49), Research on Supply Potential and Business Decision of Larch Carbon potential in Heilongjiang Province (Grant no. 2572018BM07), and Research on Forest Carbon Sink Supply Potential and Policy Tools in State-Owned Forest Area of Heilongjiang province (Grant no. 18GLD289).

References

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

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

[3] G. Dueck and T. Scheuer, "Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing," *Journal of Computational Physics*, vol. 90, no. 1, pp. 161–175, 1990.

[4] M. Creutz, "Microcanonical Monte Carlo simulation," *Physical Review Letters*, vol. 50, no. 19, pp. 1411–1414, 1983.

[5] F. S. Wen and C. S. Chang, "Tabu search approach to alarm processing in power systems," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 144, no. 1, pp. 31–38, 1997.

[6] C. Voudouris, E. P. K. Tsang, and A. Alsheddy, "Guided local search," in *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds., pp. 321–361, Springer, Berlin, Germany, 2010.

[7] B. A. Tolson and C. A. Shoemaker, "Dynamically dimensioned search algorithm for computationally efficient watershed model calibration," *Water Resources Research*, vol. 43, no. 1-6, p. W01413, 2007.

[8] B. A. Tolson, M. Asadzadeh, H. R. Maier, and A. Zecchin, "Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization," *Water Resources Research*, vol. 45, no. 12, article W12416, 2009.

[9] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin, Germany, 3rd edition, 1996.

[10] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, 1992.

[11] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, November-December 1995.

[12] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing and Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

[13] J. Xu and F. Yan, "Hybrid Nelder-Mead algorithm and dragonfly algorithm for function optimization and the training of a multilayer perceptron," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3473–3487, 2019.

[14] Q. Y. Duan, V. K. Gupta, and S. Sorooshian, "Shuffled complex evolution approach for effective and efficient global minimization," *Journal of Optimization Theory and Applications*, vol. 76, no. 3, pp. 501–521, 1993.

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

[16] X. Huang, W. Liao, X. Lei et al., "Parameter optimization of distributed hydrological model with a modified dynamically dimensioned search algorithm," *Environmental Modelling and Software*, vol. 52, pp. 98–110, 2014.

[17] F.-R. Lin, N.-J. Wu, C.-H. Tu, and T.-K. Tsay, "Automatic calibration of an unsteady river flow model by using dynamically dimensioned search algorithm," *Mathematical Problems In Engineering*, vol. 2017, Article ID 7919324, 19 pages, 2017.

[18] A. Masoud and T. Bryan, "Pareto archived dynamically dimensioned search with hypervolume-based selection for multi-objective optimization," *Engineering Optimization*, vol. 45, no. 12, pp. 1489–1509, 2013.

[19] M. J. F. G. Macêdo, M. F. P. Costa, A. M. A. C. Rocha, E. W. Karas, A. M. A. C. Rocha, and E. W. Karas, "Combining filter method and dynamically dimensioned search for constrained global optimization," in *Computational Science and Its Applications-ICCSA 2017*, pp. 119–134, Springer, Cham, Switzerland, 2017.

[20] R. Sarkhel, N. Das, A. K. Saha, and M. Nasipuri, "An improved Harmony Search Algorithm embedded with a novel piecewise opposition based learning algorithm," *Engineering Applications of Artificial Intelligence*, vol. 67, pp. 317–330, 2018.

[21] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 1, pp. 695–701, Vienna, Austria, November 2005.

[22] E. Cuevas, L. Enríquez, D. Zaldívar, and M. Pérez-Cisneros, "A selection method for evolutionary algorithms based on the Golden Section," *Expert Systems With Applications*, vol. 106, pp. 183–196, 2018.

[23] J. A. Koupaei, S. M. M. Hosseini, and F. M. M. Ghaini, "A new optimization algorithm based on chaotic maps and golden section search method," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 201–214, 2016.

[24] W. Long, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization," *Engineering Applications of Artificial Intelligence*, vol. 68, pp. 63–80, 2018.

[25] W. Gao, S. Liu, and L. Huang, "A novel artificial bee colony algorithm based on modified search equation and orthogonal learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1011–1024, 2013.

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

[27] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

[28] S. R. K.S. and S. Murugan, "Memory based hybrid dragonfly algorithm for numerical optimization problems," *Expert Systems with Applications*, vol. 83, pp. 63–78, 2017.

[29] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[30] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 55–66, 2011.

[31] H. Wang, W. Wang, X. Zhou et al., "Firefly algorithm with neighborhood attraction," *Information Sciences*, vol. 382-383, pp. 374–387, 2017.

[32] J. Derrac, J. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[33] R. J. Zhao, "The Xinanjiang model applied in China," *Journal of Hydrology*, vol. 135, no. 1–4, pp. 371–381, 1992.

[34] Y. Ye, X. Song, J. Zhang, F. Kong, and G. Ma, "Parameter identification and calibration of the Xin'anjiang model using the surrogate modeling approach," *Frontiers of Earth Science*, vol. 8, no. 2, pp. 264–281, 2014.

[35] Q. J. Wang, "The genetic algorithm and its application to calibrating conceptual rainfall-runoff models," *Water Resources Research*, vol. 27, no. 9, pp. 2467–2471, 1991.

[36] Q. Duan, S. Sorooshian, and V. Gupta, "Effective and efficient global optimization for conceptual rainfall-runoff models," *Water Resources Research*, vol. 28, no. 4, pp. 1015–1031, 1992.

[37] X. Song, F. Kong, C. Zhan, J. Han, and X. Zhang, "Parameter identification and global sensitivity analysis of Xinanjiang model using meta-modeling approach," *Water Science and Engineering*, vol. 6, no. 1, pp. 1–17, 2013.