PAIDDE: A Permutation-Archive Information Directed Differential Evolution Algorithm

Xiaosi Li\textsuperscript{1,2}, Kaiyu Wang\textsuperscript{2}, Haichuan Yang\textsuperscript{2}, Sichen Tao\textsuperscript{2}, Shuai Feng\textsuperscript{3}, and Shangce Gao\textsuperscript{2}, (Senior Member, IEEE)

\textsuperscript{1}Department of Engineering, Wesoft Company Ltd., Kawasaki-shi 210-0024, Japan.
\textsuperscript{2}Faculty of Engineering, University of Toyama, Toyama-shi 930-8555, Japan.
\textsuperscript{3}China Unicom Smart Connection Technology Limited, Beijing 100037, China.

Corresponding author: Shangce Gao (gaosc@eng.u-toyama.ac.jp).

This research was partially supported by the JSPS KAKENHI Grant Number JP22H03643 and JP19K22891.

\section*{ABSTRACT} Evolutionary algorithms have shown great successes in various real-world applications ranging in molecule to astronomy. As a mainstream evolutionary algorithm, differential evolution (DE) possesses the characteristics of simple algorithmic structure, easy implementation, and efficient search performance. Nevertheless, it still suffers from the issues of local optimal trapping and premature of evolution problems. In this study, we innovatively improve the performance of DE by incorporating a full utilization of information feedback, which includes the population's holistic information and the direction of differential vectors. The proposed permutation-archive information directed differential evolution (PAIDDE) algorithm is verified on a set of 29 benchmark numerical functions and 22 real-world optimization problems. Extensive experimental and statistical results show that PAIDDE can significantly outperform other 12 state-of-the-art algorithms in terms of solution qualities. Additionally, the computational complexity, solution distribution, convergence speed, search dynamics, and population diversity of PAIDDE are systematically analyzed. The source code of PAIDDE can be found at https://toyamaailab.github.io/sourcedata.html.

\section*{INDEX TERMS} Meta-heuristic algorithms, Differential evolution, Optimization, Population diversity, Evolutionary algorithms, Swarm intelligence

\section*{I. INTRODUCTION}

Evolutionary computation, inspired from the mechanisms of biological evolution, has achieved great success in various real-world tasks ranging in molecule to astronomy [1]–[7]. Representative algorithmic implementations of evolutionary computation include genetic programming [8], evolution strategies [9], evolutionary programming [10], genetic algorithms [11], and differential evolution (DE) [12]. In the past three decades, DE has attracted much interest due to its very promising performance in continuous optimization problems. Owing to its great efficiency, straightforward implementation, simple structure, and fast convergence [13], it was proved to be the most competitive algorithm in the Second International Contest on Evolutionary Computation. For example, IMODE which simultaneously used multiple differential search operators [14] ranked 1st in the IEEE CEC2020 competition. Till now, DE has been successfully applied on various scenarios, including image processing [15], manufacturing design [16], energy engineering [17], industry challenges [18], chemical engineering [19], electric vehicles [20], neural modeling optimization [21], [22], smart grid security [23], and deep neural network learning [24]. More other applications of DE can be referred in some survey researches [25]–[31].

However, DE as an algorithm par excellence does not mean that its edifice is complete [32], [33]. Considering the well-known “No-Free-Lunch Theorem” [34], different evolutionary computation have corresponding advantages and disadvantages in addressing different optimization problems. The search for a one-size-fits-all algorithm to solve all problems is unrealistic, which drives the emergence of new algorithms [35]–[39]. Although DE has a high local exploitation ability, which makes it converges quickly, it still lacks an excellent global exploration capability to escape from the local optima once being caught. Many attempts have been undertaken to address the issues of DE from various aspects. The following components have resulted in some significant improvements:

1) Improving search efficiency via an ensemble technique that mixes and executes multiple alternative mutation strate-
gies at the same time. Representative ones include EPSDE
[40], SEDE [41], SaDE [42], and IMODE [14].

2) Increasing the usage of each individual’s information in the crossover operator. OXDE [43], which incorporates an orthogonal crossover operator, shows excellent global exploration capacity.

3) Keeping the balance between exploration and exploitation. CIADE [44] and SCJADE [45] are two examples of algorithms that exploit the ergodicity properties of numerous chaotic maps to achieve such equilibrium. Additionally, population diversity maintenance [46] also aims to realize such exploration and exploitation balance.

4) Adopting complex adaptive parameter control techniques. EL SHADE [47], LSHADE [48], SHADE [49], JADE [50], and jDE [51] are some of the most representative ones.

5) Using unique population structure to realize efficient communication among individuals. For example, scale-free-based population structures are used in [52], [53], distributed computation is implemented in [54], and two-stage realization is proposed in [55].

6) Other non-traditional concepts incorporated into DE-based approaches, e.g., opposition-based DE [56], memetic DE [57], and estimation distribution-based DE [58].

To summarize, the above algorithms have made significant progress in the field of DE, but there is still room to further improve the search efficiency of DE. In particular, the information arisen from the individuals during evolution has not been fully utilized, which motivates us to propose such an efficient DE variant from this aspect.

Accordingly, we improve the performance of DE in this study by taking full advantage of the useful information feedback while the algorithm is running. The direction of the differential vector in DE and the population’s holistic information during evolution are two kinds of information that can be regarded to be valuable in the search. Therefore, a permutation-archive information directed differential evolution (PAIDDE) algorithm is proposed. To be more explicit, two pieces of feedback information in this research are used: 1) the holistic search information of all individuals in the population, which is obtained by a randomly generated permutation in the external archive, and 2) the direction information of the differential vector. Because the holistic information is used, all individuals in PAIDDE may participate in the comparison of fitness, while most other DE variations rely on randomly picked individuals. As a result, the search efficiency will increase with aid of such extensive information feedback from the population. Furthermore, the direction information allows PAIDDE to conduct a comprehensive search for possible solution locations. By applying these two kinds of feedback information, the population diversity and the convergence speed can be increased. To verify the performance of PAIDDE, extensive experiments are conducted based on a set of 29 benchmark optimization functions taken from IEEE CEC2017 and 22 real-world optimization problems taken from IEEE CEC2011. Comparative results with other 12 state-of-the-art algorithms show the superiority of the proposed PAIDDE.

This work aims to make contributions summarized as:

1) We provide extensive experimental evidences for that combining a population feedback information and search direction utilization scheme into DE can significantly improve its search efficiency on small, medium, and large search dimensions.

2) The proposed PAIDDE is an efficient and effective optimizer for both numerical and real-world optimization problems. Based on a large number of trials, an exhaustive performance comparison of 12 algorithms, including five variations of DE algorithms and seven additional relevant meta-heuristics, confirms that PAIDDE statistically performs the best.

3) We provide sufficient analysis for PAIDDE in terms of solution distribution, search trajectory, convergence speed, population diversity, and computational complexity. These results give more insights into the search dynamics of PAIDDE, which may also be used to assist other practitioners in developing more powerful meta-heuristic algorithms, notably in the area of reuse of knowledge derived from population and search operations.

In brief, in comparison with other DE variants, PAIDDE is a state-of-the-art algorithm which not only sophisticatedly uses the feedback information from individuals during evolution, but also achieves a significant improvement in terms of optimization performance.

The remainder of the paper is structured as follows: Section 2 introduces the traditional DE. Section 3 delves into the specifics of the PAIDDE. The experimental findings and detailed analysis are presented in Section 4. Finally, Section 5 concludes this article and outlines potential future research topics.

II. DIFFERENTIAL EVOLUTION

DE’s main concept is to design operators by simulating biological evolution’s mutation, crossover, and selection. New individuals are generated by mutation and crossover, and the best ones are retained for the following generation through selection.

The population of the t-th iteration in DE is represented by \(\{X_1^t, X_2^t, ..., X_N^t\}\), and each individual can be expressed as \(X_i^t = (x_{i1}^t, x_{i2}^t, ..., x_{iD}^t)\), where \(D\) is the dimension of the optimization problem. To begin, \(N\) individuals are generated at random to form the population. Then, a mutation operator is used to create a mutated individual \(V_i^t = (v_{i1}^t, v_{i2}^t, ..., v_{iD}^t)\). The following are four regularly utilized mutation strategies:

1) \(DE/best/1:\)
\[
V_i^t = X_{best}^t + F(X_{r1}^t - X_{r2}^t) \tag{1}
\]

2) \(DE/rand/1\)
\[
V_i^t = X_{r1}^t + F(X_{r2}^t - X_{r3}^t) \tag{2}
\]

3) \(DE/current-to-best/1\)
\[
V_i^t = X_i^t + F(X_{best}^t - X_i^t) + F(X_{r1}^t - X_{r2}^t) \tag{3}
\]
4) DE/best/2

\[ V^t_i = X^t_{\text{best}} + F(X^t_{r1} - X^t_{r2}) + F(X^t_{r3} - X^t_{r4}) \] (4)

where \( V^t_i \) is the \( i \)-th mutated individual created by a mutation operator at the \( t \)-th iteration. \( X^t_{\text{best}} \) is the \( i \)-th individual and \( X^t_{\text{best}} \) is the best individual at the \( t \)-th iteration. \( r_1, r_2, r_3, r_4 \) and \( i \), which are uniformly generated from the set \( \{1, 2, ..., NP\} \), are mutually different integers.

To create a trial vector \( U^t_i = (u^t_{1i}, u^t_{2i}, ..., u^t_{Di}) \), a crossover strategy is activated. This strategy uses individuals in \( V^t_i \) to crossover with the individuals in the paternal populations \( X^t_i \) which is the \( i \)-th individual in the \( t \)-th iteration, formulated as follows:

\[ u^t_{ij} = \begin{cases} v^t_{ij}, & j = j_{\text{rand}} \text{ or } \text{rand}(0, 1) < CR \\ x^t_{ij}, & \text{otherwise} \end{cases} \] (5)

where \( u^t_{ij} \) is the \( i \)-th trial vector at the \( t \)-th iteration, \( j = 1, 2, ..., D \) denotes the selected dimension. \( D \) is the dimensional size of the optimization problem. \( v^t_{ij}(t) \) is a mutant vector at the \( j \)-th dimension, and \( x^t_{ij}(t) \) is a target vector at the \( j \)-th dimension. \( \text{rand}(0, 1) \) is a random value uniformly generated in the interval of \((0, 1)\). \( CR \) is the parameter that controls the crossover probability that specifies the extent to which elements of the population individuals are replaced by elements of the mutant individuals. In many DE variants, \( CR = 0.9 \), \( j_{\text{rand}} \) is a random integer generated in the interval of \((1, D)\).

Finally, the fitness of all individuals \( U^t_i \) in the population are calculated. In the selection operation, \( U^t_i \) is compared with the individuals in the paternal population \( X^t_i \) and the better individual is selected to be survived into the next iteration. Without loss of generality, for a minimization problem, this process can be formulated as:

\[ X^{t+1}_i = \begin{cases} U^t_i, & f(U^t_i) < f(X^t_i) \\ X^t_i, & \text{otherwise} \end{cases} \] (6)

where \( f() \) denotes the fitness function. The processes stated above are repeated until the stop requirement is met.

III. PAIDDE

A. MOTIVATION

In the field of evolutionary computation, the utilization of feedback information has an important impact on the performance of search algorithms [59]–[62]. Based on the development process of DE, it is widely accepted that the more extensive and comprehensive the use of information, the superior the performance of DE variants [25], [28]. For example, the mutation operator \textit{current-to-pbest}/1, which is widely used in JADE [50], SHADE [49], and LSHADE [48], adds the use of information about the best individuals in the population in comparison with \textit{DE/rand}/1 applied in the original DE, thus greatly improving the algorithm search efficiency.

The mutation strategy \textit{current-to-pbest}/1 can be formulated as:

\[ V^t_i = X^t_i + F_i(X^t_{\text{pbest}} - X^t_i) + F_i(X^t_{r1} - X^t_{r2}) \] (7)

where the individual \( X^t_{\text{pbest}} \) is picked at random from the top \( NP \times p \) individuals in terms of the calculated fitness in the \( t \)-th iteration (in previous studies, \( p = 0.11 \) [49], [50]). \( r_1 \) and \( r_2 \) are two randomly selected values from the set \( \{1, 2, ..., NP\} \) and \( r_1, r_2 \) and \( i \) are not equal. \( F_i \) is a scale parameter which is used to control the search range of the population. Therefore, in this study, we try to reuse the information feedback during the operation of the algorithm, so as to further improve the algorithm’s search performance. The information is utilized in two main ways: 1) making use of the population’s holistic information, and 2) making use of the differential vector’s direction.

B. REUSING OF POPULATION’S HOLISTIC INFORMATION

According to Eq. (7), the differential vector \( V^t_i \) produced by two randomly chosen individuals in the population, i.e., \( X^t_{r1} \) and \( X^t_{r2} \), can provide a flexible movement toward the target vector. Nevertheless, because a percentage of the population will not be selected, such a random selection operation will not be able to completely utilize the population’s information. This leads to a tendency for the algorithm to converge to a local optimal solution prematurely. To solve this issue, we propose a novel holistic information utilization strategy that can be implemented as follows:

\[ V^t_i = X^t_i + F_i(X^t_{\text{pbest}} - X^t_i) + F_i(X^t_{r1} - Y^t_i) \] (8)

where \( Y^t_i \) comes from an external archive which is used to keep the diversity of population, and \( X^t_{r1} \), \( X^t_{\text{pbest}} \), and \( X^t_{r1} \) come from the primary evolution population. It is notable that the external archive with \( NP \) individuals is also initialized at random at first. The following is the mutation and crossover operators. The archive is updated via:

\[ Y^{t+1}_i = \begin{cases} U^t_{ri}, & \text{if } f(U^t_{ri}) < f(Y^t_i) \\ Y^t_i, & \text{otherwise} \end{cases} \] (9)

where \( r_i \) is chosen at random from the set \( \{1, 2, ..., NP\} \).

Remark 1: Because all \( r_i \) \( (i = 1, 2, ..., NP) \) in the external archive is a permutation rather than random integers, all individuals in the population are given the opportunity to participate in the iteration. As a result, it is possible to make use of the population’s holistic information.

Remark 2: Because all \( r_i \) are random permutations, the trail vector \( U^t_{ri} \) is not guaranteed to be compared with its corresponding target vector \( Y^t_{ri} \). Instead, it is compared to a population-wide goal vector. The demographic diversity could be preserved as a result of this.

C. REUSING OF DIFFERENTIAL VECTOR’S DIRECTION

Because earlier DE variants have rarely explored the differential vector’s direction information, which is also thought to be vital to its search performance [63], this research provides
a straight-forward but effective utilization strategy for the direction information, as described in Eqs. (10)(11):

$$V^t_i = X^t_i + F_i(X^t_{r1} - X^t_i) + G(α, β, ζ)F_i(X^t_{pbest} - Y^t_i)$$

(10)

$$G(α, β, ζ) = g(f(X^t_{r1}), f(X^t_{pbest}), f(Y^t_i))$$

(11)

The direction control function $G(α, β, ζ)$ is defined by the relationship between its variables, which is shown as:

$$G() = \begin{cases} 
1, & \text{if } α ≤ β ≤ ζ \text{ or } β ≤ α ≤ ζ \\
1, & \text{if } rand(0, 1) ≥ Q \text{ and } \\
-1, & \text{if } α ≤ β ≤ ζ \text{ or } ζ ≤ α ≤ β \text{ or } ζ ≤ β ≤ α \\
-1, & \text{if } rand(0, 1) ≥ Q \text{ and } β ≤ ζ ≤ α \\
1, & \text{if } rand(0, 1) < Q \text{ and } β ≤ ζ ≤ α 
\end{cases}$$

(12)

where $T_{max}$ denotes the maximum function evaluation number, $T$ is the current function evaluation number, and $Q = \frac{α + β + ζ}{T_{max}}$ is a threshold.

According to Eqs. (10), (11), and (12), the mutation vector $V^t_i$ generated upon the target vector $X^t_i$ is based on the relationship among randomly chosen individual $X^t_{r1}$, the current promising individual $X^t_{pbest}$, and the individual $Y^t_i$ from the external archive. The implementation of Eq. (12) can be paraphrased as follows:

- **Case 1**: $f(X^t_{r1}) ≤ f(X^t_{pbest}) ≤ f(Y^t_i)$ and $f(X^t_{pbest}) ≤ f(Y^t_i)$ and $f(Y^t_i) ≤ f(Y^t_i)$. In this case, the fitness of $X^t_{pbest}$ is better than $Y^t_i$, which means that the differential vector $(X^t_{pbest} - Y^t_i)$ can improve the solution. As a result, such movement patterns are more likely to lead to a promising search location and are likely to make $X^t_i$ better. Thus the direction of $(X^t_{pbest} - Y^t_i)$ remains unchanged, i.e., $G() = 1$.

- **Case 2**: $f(X^t_{r1}) ≤ f(Y^t_i) ≤ f(X^t_{pbest})$, $f(Y^t_i) ≤ f(X^t_{r1}) ≤ f(X^t_{pbest})$, and $f(Y^t_i) ≤ f(X^t_{pbest}) ≤ f(Y^t_i)$. In this example, $X^t_{pbest}$ performs worse than $Y^t_i$, indicating that such search direction is more likely directing to an undesirable search area. As a result, the opposing movement direction is likely to make $X^t_i$ better, particularly in later search phases of evolution when $X^t_{r1}$ may have been locked in a local optimal position.

- **Case 3**: $f(X^t_{pbest}) ≤ f(Y^t_i) ≤ f(X^t_{r1})$. In this case, $X^t_{pbest}$ performs better than $Y^t_i$ in this situation, while $X^t_{r1}$ performs the worst. The fact that the differential vector $(X^t_{pbest} - Y^t_i)$ is favorable for improving $X^t_{r1}$, $X^t_{r1}$ is changed in the opposite direction as $(Y^t_i - X^t_{pbest})$ to give the population more diversity in the early search periods. The population diversity can be preserved accordingly, which improves the algorithm's global exploration capability. With the iteration number increases, $X^t_{r1}$ moves along with the movement direction of the differential vector, thus increasing the convergence speed of PAIDDE.

#### Algorithm 1: The main procedure of PAIDDE.

1. Input: $T_{max}$
2. Output: the optimal solution in the population when the algorithm is terminated.

3. begin
   4. Set $H = 5$, $p = 0.11$, $t = 1$, all entries in $M_F$ and $M_C$, to 0.5.
   5. Initialize the main evolution population $X^0_i$ and external archive $Y^0_i$ ($i = 1, 2, \ldots, NP$) randomly.
   6. while the termination criteria are not met do
      7. for $i=1$ to $NP\ t$ do
         8. Generate the scale parameter $Cr_i$ and crossover rate parameter $F_i$ for the individual $X^t_i$ according to Eqs. (18) and (17).
         9. Randomly select $X^t_{pbest}$ as one of the best 100p% individuals.
         10. Randomly select $r1 \neq i$.
         11. Generate mutant vector $V^t_i$ using Eq. (10).
         12. Generate trail vector $U^t_i$ using Eq. (13).
         13. if $f(U^t_i) < f(X^t_i)$ then
            14. $X^t_{i+1} = U^t_i$, $S_{Cr} = S_{Cr} \cup \{Cr_i\}$,
            15. $S_F = F_i$.
         else
            16. $X^t_{i+1} = X^t_i$.
         17. Update the $M_F$ success memory using Eq. (14).
         18. Update the $M_{Cr}$ success memory using Eq. (15).
         19. $T = T + 1$ at every time when the fitness evaluation is called.
         20. Calculate the population size $NP+t+1$ using Eq. (19).
         21. if $NP > NP+t+1$ then
            22. sort individuals in the population in an ascending order of fitness and delete the worst $NP - NP+t+1$ individuals.
            23. $t += 1$;
   6. end

Besides, such transformation from opposite direction to original direction is dynamically controlled by a threshold $Q$, which increases along with the function evaluation number. $rand(0, 1) ≥ Q$ indicates the early search phase, and $rand(0, 1) < Q$ means the later one. To summarize, the opposite direction of differential vector is more likely generated to enable PAIDDE to maintain a good population diversity in the early search phase or jump out of the trapped local optima in the later search phase, while the same direction is used to accelerate the convergence speed and find more promising search areas.
D. REMAINING ASPECTS OF PAIDDE

In PAIDDE, \( V^t_i = (v^t_{i1}, v^t_{i2}, ..., v^t_{ij}, ..., v^t_{iD}) \) is generated by Eq. (10). Then \( U^t_i = (u^t_{i1}, u^t_{i2}, ..., u^t_{ij}, ..., u^t_{iD}) \) is generated using the crossover operation:

\[
\begin{align*}
    u^t_{ij} = \begin{cases} 
    v^t_{ij}, & \text{if } rand(0,1) \leq Cr_i \text{ or } j = j_{rand} \\
    x^t_{ij}, & \text{otherwise}
    \end{cases}
\end{align*}
\]  

(13)

where \( X^t_i \) is connected with the crossover rate parameter \( Cr_i \in [0, 1] \). Then, using the selection operation indicated in Eq. (6), an evolved individual \( X^t_{i+1} \) is created.

**Remark 3:** In PAIDDE, each individual \( X^t_i \) has its own scale parameter \( F_i \) and crossover rate parameter \( Cr_i \), which can be adjusted in a methodical manner.

First, two matrices \( M_F = \{ M_F(1), M_F(2), ..., M_F(H) \} \) and \( M_{Cr} = \{ M_{Cr}(1), M_{Cr}(2), ..., M_{Cr}(H) \} \) are generated, and \( H \) is the number of historical memory entries (previous research has shown that \( H = 5 \) is the most effective [49], [50]). All entries in \( M_F \) and \( M_{Cr} \) are initialized to 0.5. In the \( k \)-iteration, when an individual \( X^t_{i+1} \) is replaced according to the fitness by its trial vector in Eq. (6), the \( F_i \) and \( Cr_i \) values used by this individual are recorded in two discrete sets, \( S_F \) and \( S_{Cr} \), respectively. Then these entries are updated according to:

\[
M_F^{t+1}(k) = \begin{cases} 
\frac{\sum_{k=1}^{\left| S_F \right|} w_k S_F^2(k)}{\sum_{k=1}^{\left| S_F \right|} w_k S_F(k)}, & \text{if } S_F \neq \emptyset \\
M_F^{t}(k), & \text{otherwise}
\end{cases}
\]

(14)

\[
M_{Cr}^{t+1}(k) = \begin{cases} 
\frac{\sum_{k=1}^{\left| S_{Cr} \right|} w_k S_{Cr}(k)}{\sum_{k=1}^{\left| S_{Cr} \right|} w_k}, & \text{if } S_{Cr} \neq \emptyset \\
M_{Cr}^{t}(k), & \text{otherwise}
\end{cases}
\]

(15)

\[
w_k = \frac{|f(U^t_{j_{rand}}) - f(X^t_j)|}{\sum_k |f(U^t_k) - f(X^t_k)|}
\]

(16)

The index value \( k \) records the position of the entries to be updated in the memory \( M_F \) and \( M_{Cr} \), and \( 1 \leq k \leq H \). Initially, \( k \) is firstly set to be 1 and thereafter incremented whenever a new memory entry is inserted.

Based on \( M_F \) and \( M_{Cr} \), the parameters \( Cr_i \) and \( F_i \) are generated by:

\[
Cr_i = randn_i(M_{Cr}(r_i), 0.1)
\]

(17)

\[
F_i = randc_i(M_F(0.1))
\]

(18)

where \( randn_i(\mu, \sigma^2) \) and \( randc_i(\mu, \sigma^2) \) are two values randomly selected from Cauchy and normal distributions with mean \( \mu \) and variance \( \sigma^2 \), respectively.

In addition, the linear decline scheme proposed in [48], [64] is also used in PAIDDE. The size of individuals decreases linearly, which is formulated as a function of the number of fitness evaluations, expressed as:

\[
NP^{t+1} = \text{round} \left[ NP^{\text{init}} - (NP^{\text{init}} - NP^{\text{min}}) \frac{T}{MaxT} \right]
\]

(19)

where the initial size of the population \( NP^{\text{init}} \) is set to \( 18D \) according to [64], and the minimal number of the population \( NP^{\text{min}} = 4 \).

Algorithm 1 depicts the PAIDDE’s implementation structure.

E. ANALYSIS OF COMPUTATIONAL COMPLEXITY

In comparison with the original DE, PAIDDE’s computational complexity is increased by the initialization (Line 5 in Algorithm 1), new mutant vector generation (Line 12), and update operations (Lines 18 and 19) of the success memory and external archive. The computational complexity of these additional updating operators requires up to \( O(NP^{\text{init}} \times D) \). As a result, overall computing complexity of PAIDDE is \( O(NP^{\text{init}} \times D \times MaxG) \), where \( MaxG \) indicates the maximal number of iterations. It is worth pointing out that this time complexity of PAIDDE is the same as those of DE and many other DE variants if the population size is kept constant. From the above analysis, we can conclude that PAIDDE does not significantly increase the computational complexity of DE.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. BENCHMARK FUNCTIONS

The widely used IEEE CEC2017 benchmark test set [65] consists of 29 numerical functions. These 29 functions are divided into different types, where F1-F2 are unimodal functions, F3-F9 are multimodal functions, F10-F19 are hybrid functions, and F20-F29 are composition functions. Each function with dimensions of 30, 50 and 100 is tested.

B. EXPERIMENTAL SETUP

The maximum value of function evaluation \( T_{\text{max}} \) is set to \( 10^4 \cdot D \), where \( D \) denotes a function’s dimension and is set to 30, 50, and 100. Each function is run 30 times independently for each dimension to acquire statistical data. Matlab software is used to implement all experimental data on a PC with a 3.10GHz Intel(R) Core(TM) i5-4400 processor and 8GB RAM.

C. PERFORMANCE EVALUATION CRITERIA

1. **Non-parametric statistical test**: To determine whether the difference between each set of algorithms is significant, we utilize the Wilcoxon rank-sum test [66]. The sign “+” represents the PAIDDE is significantly better, tied or significantly worse than its competitor.

2. **Convergence curve graph**: Convergence curves are used to measure the algorithms’ convergence rates, and they show the history of the current best solution in each iteration. The number of function evaluations is represented on the X-axis, while the average best fitness is represented on the Y-axis.

3. **Box plot diagram**: The quality of the solutions is represented by the box-and-whisker diagrams. The red “+”, upper block line, upper blue line, red line, lower blue line, and lower block line denote extreme, maximum, first quartile, median, third quartile, and minimum value, respectively. The solution’s distribution is indicated by the distance between the maximum and...
minimum values, where a shorter distance value suggests that the standard deviation of the solution is smaller and the search performance is more stable, and vice-versa. Furthermore, the lowest height indicates the algorithm’s solution quality. Generally, a lower altitude indicates that the algorithm can produce a better result.

### D. COMPARISON BETWEEN PAIDDE AND OTHER VARIANTS OF DE

In this section, we compare the following algorithms with PAIDDE: DEGoS [67], CJADE [68], SCJADE [45], IMODE [14], and SHADE [49], and their parameters are set according to the corresponding literature, as summarized in Table 1.

The statistical results between PAIDDE and other variants of DE on 29 benchmark functions with 30, 50 and 100 dimensions are shown in Table 2. According to this table, the number of numerical functions on which the PAIDDE has considerably outperformed DEGoS, CJADE, SCJADE, IMODE, and SHADE is 21, 23, 24, 26, and 22 for $D = 30$, 21, 25, 26, and 25 for $D = 50$, and 24, 21, 21, 25, and 23 for $D = 100$, respectively. In addition, according to Table 3, the Friedman statistical results [66] show that PAIDDE has a significant advantage over its competitors in the 30, 50 and 100 dimensions.

The box-and-whisker diagrams for typical benchmark functions (a multimodal function $F_{4}$, a hybrid function $F_{17}$, and a composition function $F_{25}$) are shown in Fig. 1 to show the difference between PAIDDE and other DE variations in 30, 50, and 100 dimensions. Fig. 1 demonstrates that PAIDDE has better robustness and higher accuracy than its competitors for low, medium and high dimensional optimization problems.

The convergence graphs for these typical benchmark functions are illustrated in Fig. 2 to show the difference between the PAIDDE and other DE variations in 30, 50, and 100 dimensions. From it, two characteristics of PAIDDE in the convergence process can be found: 1) The slower convergence of PAIDDE in the early part of the iteration indicates that

| Algorithm | Year | Parameters |
|-----------|------|------------|
| PAIDDE    | 2022 | $N^{\text{init}} = 18 \times D$, $N^{\min} = 4$, $p = 0.11$, $H = 5$ |
| DEGoS     | 2019 | $N = 100$, $F = 0.5$, $C_{r} = 0.9$ |
| CJADE     | 2019 | $N = 100$, $p = 0.05$ |
| SCJADE    | 2021 | $N = 100$, $p = 0.05$ |
| IMODE     | 2020 | $N^{\text{init}} = 6 \times D^2$, $N^{\min} = 4$, $p = 0.1$, $H = 20 \times D$ |
| SHADE     | 2013 | $N = 100$, $p = 0.11$, $H = 5$ |
| DGSA      | 2021 | $N = 100$, $D_N = 5$ |
| MLGSA     | 2020 | $N = 100$ |
| ALGSA     | 2020 | $N = 100$, $G_{;}(0) = 100$, $\alpha = 20$, $\text{limit} = 2$, $p = 0.5$ |
| CMAES     | 2001 | $N = \text{round}(4 \times 3 \times \log(D))/2$, $\sigma = 0.25$ |
| GLPSO     | 2016 | $N = 100$, $\omega = 0.7298$, $pm = 0.01$, $c = 1.49618$, $sg = 7$ |
| SE        | 2019 | $N = 20$, $D_{SF} = [5 \sim 10]$ |
| SASS      | 2019 | $N^{\text{init}} = 18 \times D$, $N^{\min} = 4$, $p = 0.11$, $H = 5$ |

| Algorithm | PAIDDE | DEGoS | CJADE | SCJADE | IMODE | SHADE |
|-----------|--------|-------|-------|--------|-------|-------|
| $D = 30$  | 1      | 5     | 5     | 4      | 6     | 2     |
| $D = 50$  | 1      | 4     | 5     | 3      | 6     | 2     |
| $D = 100$ | 1      | 5     | 4     | 2      | 5     | 3     |

| Algorithm | PAIDDE | DEGoS | CJADE | SCJADE | IMODE | SHADE |
|-----------|--------|-------|-------|--------|-------|-------|
| $D = 30$  | 25/1/3 | 24/4/1| 23/1/5| 26/2/1 | 29/0/0| 29/0/0|
| $D = 50$  | 26/1/2 | 27/0/2| 21/3/5| 24/3/2 | 29/0/0| 28/1/0|
| $D = 100$ | 28/1/0 | 27/0/2| 15/1/3| 21/4/4 | 29/0/0| 27/1/1|

| Algorithm | PAIDDE | DEGoS | CJADE | SCJADE | IMODE | SHADE |
|-----------|--------|-------|-------|--------|-------|-------|
| $D = 30$  | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ |
| $D = 50$  | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ |
| $D = 100$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ | $+/ \approx /$ |

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
TABLE 5: Friedman statistical rankings of PAIDDE and other comparison algorithms on 29 benchmark functions with 30, 50 and 100 dimensions.

| Algorithm | PAIDDE | DGSA | MLGSA | ALGSA | CMAES | GLPSO | SE | SASS |
|-----------|--------|------|-------|-------|-------|-------|----|------|
| $D = 30$  | 1      | 4    | 5     | 3     | 7     | 8     | 6  | 2    |
| $D = 50$  | 1      | 7    | 6     | 3     | 5     | 8     | 4  | 2    |
| $D = 100$ | 1      | 7    | 6     | 3     | 4     | 8     | 5  | 2    |
| **Total Ranking** | 1      | 7    | 6     | 3     | 5     | 8     | 4  | 2    |

Fig. 1: Box plot diagrams of optimization errors obtained by PAIDDE and other variants of DE on F4, F17 and F25 with 30, 50, and 100 dimensions, respectively.

the proposed algorithm has a strong exploration capability in the early period of the iteration and is less likely to fall into local optima. During the late stages, PAIDDE converges much faster than the other algorithms, indicating that it has a better ability to exploit in the late stages and performs better at finding promising solutions. 2) At the end of the iteration, the solution obtained by PAIDDE has the smallest fitness value, which demonstrates that the performance of PAIDDE is stronger than the other algorithms.

The aforementioned results indicate that, in comparison with other state-of-the-art DE variants, PAIDDE has superior search performance with the aid of full utilization of the population’s holistic information and the direction of differential vectors.

E. COMPARISON BETWEEN PAIDDE AND OTHER STATE-OF-THE-ART ALGORITHMS

In this section, we compare the following related metaheuristic algorithms with PAIDDE: DGSA [69], MLGSA [70], ALGSA [71], CMAES [72], GLPSO [73], SE [74],
and SASS [75], and their parameters are set according to the corresponding literature, as summarized in Table 1.

The statistical results between PAIDDE and other algorithms in 30, 50, and 100 dimensions, respectively. From Fig. 3, we can find that the PAIDDE possesses the lowest altitude and shortest distant on F7, F16, and F20 in 30, 50, and 100 dimensions. Figs. 3 and 4 provide box-and-whisker diagrams and convergence graphs for the benchmark function to show the difference between the PAIDDE and other algorithms in 30, 50, and 100 dimensions, respectively. From Fig. 4, it can be noticed that the solution obtained by PAIDDE has the smallest fitness value at the end of the iteration on F7, F16, and F20 in 30, 50, and 100 dimensions, respectively.

TABLE 6: Running times of all compared algorithms for all test functions in IEEE CEC2017.

| Algorithm | D = 30  | D = 50  | D = 100 | Total   |
|-----------|---------|---------|---------|---------|
| PAIDDE    | 3.46E+02| 1.11E+03| 5.93E+03| 7.38E+03|
| DEGoS     | 1.27E+02| 3.25E+02| 1.39E+03| 1.84E+03|
| CJADE     | 1.04E+02| 2.74E+02| 1.30E+03| 1.67E+03|
| SCJADE    | 1.01E+02| 2.70E+02| 1.30E+03| 1.67E+03|
| IMODE     | 9.86E+01| 8.23E+02| 2.36E+04| 2.45E+04|
| SHADE     | 1.55E+02| 3.06E+02| 1.29E+03| 1.75E+03|
| DGSA      | 3.05E+02| 7.04E+02| 3.07E+03| 4.08E+03|
| MLGSA     | 1.15E+03| 2.56E+03| 3.60E+03| 7.31E+03|
| ALGSA     | 1.12E+03| 1.37E+03| 3.70E+03| 6.19E+03|
| CMAES     | 1.07E+02| 3.05E+02| 1.89E+03| 2.50E+03|
| GLPSO     | 2.15E+03| 7.69E+03| 1.95E+04| 2.93E+04|
| SE        | 7.53E+02| 1.30E+03| 3.28E+03| 5.33E+03|
| SASS      | 2.01E+02| 3.99E+02| 1.56E+03| 2.16E+03|

and SASS [75], and their parameters are set according to the corresponding literature, as summarized in Table 1.

The statistical results between PAIDDE and other algorithms on 29 benchmark functions with 30, 50 and 100 dimensions are shown in Table 4. According to this table, the number of numerical functions on which the PAIDDE has significantly outperformed DGSA, MLGSA, ALGSA, CMAES, GLPSO, SE, and SASS is 25, 24, 23, 26, 29, 29 and 12, for D = 30, 26, 27, 21, 24, 29, 28 and 17 for D = 50, and 28, 27, 15, 21, 29, 27 and 17 for D = 100, respectively. In addition, according to Table 5, we can see that PAIDDE has a significant advantage over its competitors in the 30, 50 and 100 dimensions, respectively. From Fig. 3, we can find that the PAIDDE possesses the lowest altitude and shortest distant on F7, F16, and F20 in 30, 50, and 100 dimensions, respectively. From Fig. 4, it can be noticed that the solution obtained by PAIDDE has the smallest fitness value at the end of the iteration on F7, F16, and F20 in 30,
50, and 100 dimensions, respectively.

In addition, the running times in seconds of PAIDDE and its peers on IEEE CEC2017 are summarized in Table 6. From it, we can find that the most computationally expensive is GLPSO, and the fastest is SCJADE. Together with analysis results in Section III-E, we can conclude that PAIDDE is computationally effective. From the above analysis, we can see that PAIDDE has an absolute advantage in the face of other state-of-the-art meta-heuristic algorithms in terms of solution qualities.

F. REAL-WORLD OPTIMIZATION PROBLEMS

To further test the performance of PAIDDE, 22 real-world problems from IEEE CEC2011 [76] are used. In this section, we compared all 12 algorithms with PAIDDE mentioned before, including DEGoS, CJADE, SCJADE, IMODE, SHADE, DGSA, MLGSA, ALGSA, CMAES, GLPSO, SE, and SASS, and their parameters are set according to the corresponding literature, as summarized in Table 1.

The statistical results between PAIDDE and other algorithms on 22 real-world problems are shown in Tables 7 and 8. From these tables, it can be observed that the number of problems on which the PAIDDE has significantly outperformed DEGoS, CJADE, SCJADE, IMODE, SHADE, DGSA, MLGSA, ALGSA, CMAES, GLPSO, SE, and SASS are 16, 17, 16, 18, 22, 19, 20, 21, 20, 20, and 5, respectively. According to Tables 7 and 8, we can see that PAIDDE has a significant advantage over its competitors in the IEEE CEC2011 real-world application, which verifies its practicality.

G. ANALYSIS OF THE BALANCE BETWEEN EXPLORATION AND EXPLOITATION

To clearly illustrate the search dynamics and trajectory of PAIDDE, Figs. 5 and 6 show the distribution of populations in the solution space of three typical functions with 2 dimensions. These functions are the unimodal function F2, the multimodal function F4 and the composition function F27. The population size is 200 and the maximum number of iterations is 100. The search space range for each dimension is [-100, 100]. In these figures, the blue color indicates the individuals of the population, and the red pentagram represent the current best individual. As can be seen from them, PAIDDE shows good exploration ability as the distribution of the population in the solution space is more uniform and dispersed when the number of iterations is small. When the number of iterations increases, the population of PAIDDE gradually converges to the global optimum and converges to the optimal point at the end of the iteration, which indicates that PAIDDE has strong exploitation ability in the late iteration.

H. ANALYSIS OF POPULATION DIVERSITY

The main feature of PAIDDE is to balance the exploration and exploitation with the aid of a sophisticated search direction. To better visualize the characteristics of PAIDDE, the population diversity is considered and calculated as follows:

\[
Div(x) = \frac{1}{N} \sum_{i=1}^{N} \|x_i - \bar{x}\| / \max_{1 \leq i,j \leq N} \|x_i - x_j\| 
\]

where \(N\) and \(\bar{x}\) are the population size and the average point, respectively.

\[
\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i 
\]

From Fig. 7, the following conclusions can be drawn: 1) PAIDDE has different diversity curves for different problems, which indicates that PAIDDE has a strong adaptive mechanism for different problems. 2) The overall trend in population diversity for PAIDDE is increasing over the course of the iterations, suggesting that PAIDDE has a strong ability to jump out of local optima. It’s important to point out that, in the later iterations, PAIDDE speeds up the rate of convergence even although it has more population diversity. This shows that PAIDDE has a good balance of exploitation and exploration.

V. CONCLUSION

In this study, a permutation-archive information directed differential evolution method (PAIDDE) is proposed. In PAIDDE, the modified mutation operator can utilize the population’s holistic information by using a random permutation of individuals in the external archive. As a result, the mutant vector can inherit all potentially significant information contained in these individuals, enhancing the algorithm’s search efficiency. The improved use of the differential vector’s search direction not only gives the algorithm a more powerful capacity to jump out the local optima, but it also accelerates its convergence speed.

The above findings open up new avenues for future research: 1) We will apply the new technique adopted in this paper to other algorithms, aiming to apply this improvement pervasively to all types of meta-heuristic algorithms. 2) Performance comparison with other state-of-the-art meta-heuristic algorithms, e.g., Jaya algorithm [77] and brain storm optimization [78], could be also implemented to further show PAIDDE’s superiority. 3) The performance of PAIDDE should be further verified on other real-world prediction [79], recognition [80] and classification problems, e.g., sequence planning [81], Internet of vehicles [82], and power converters problems [83]. 4) Theoretical analyses for PAIDDE, e.g., the first hitting time [33], [84], and the ablation study of the individual impact of reusing holistic
Fig. 3: Box plot diagrams of optimization errors obtained by PAIDDE and other comparison algorithms on F7, F16 and F20 with 30, 50, and 100 dimensions, respectively.

TABLE 8: Wilcoxon statistical results obtained by PAIDDE and comparison algorithms on 22 real-world problems.

| PAIDDE vs. | DGSA | MLGSA | ALGSA | CMAES | GLPSO | SE | SASS |
|------------|------|-------|-------|-------|-------|----|------|
| +/− | 22/00 | 19/12 | 20/11 | 21/01 | 20/11 | 20/20 | 5/14/3 |

information and differential vector’s direction, are also worth being investigated.

REFERENCES
[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] A. Song, G. Wu, W. Pedrycz, and L. Wang, “Integrating variable reduction strategy with evolutionary algorithms for solving nonlinear equations systems,” IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 1, pp. 75–89, 2021.
[3] 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.
[4] Y. Hua, Q. Liu, K. Hao, and Y. Jin, “A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts,” IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 2, pp. 303–318, 2021.
[5] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, “Dendritic neural model with effective learning algorithms for classification, approximation, and prediction,” IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 2, pp. 601–614, 2019.
[6] J. Wang, Y. Sun, Z. Zhang, and S. Gao, “Solving multirip pickup and delivery problem with time windows and manpower planning using multiobjective algorithms,” IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 4, pp. 1134–1153, 2020.
[7] G.-Q. Zeng, X.-Q. Xie, M.-R. Chen, and J. Weng, “Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems,” Swarm and Evolutionary Computation, vol. 44, pp. 320–334, 2019.
[8] P. G. Espejo, S. Ventura, and F. Herrera, “A survey on the application of genetic programming to classification,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 2, pp. 121–144, 2009.
[9] H.-G. Beyer and H.-P. Schwefel, “Evolution strategies—a comprehensive introduction,” Natural Computing, vol. 1, no. 1, pp. 3–52, 2002.
Fig. 4: Convergence graphs of average errors obtained by PAIDDE and other comparison algorithms on F7, F16 and F20 with 30, 50, and 100 dimensions, respectively.
Fig. 5: Search history of individuals in PAIDDE with respect to iteration (1).
Fig. 6: Search history of individuals in PAIDDE with respect to iteration (2).

Xiaosi Li et al.: A Permutation-Archive Information Directed Differential Evolution Algorithm

estimation of solar photovoltaic models via a self-adaptive ensemble-based differential evolution,” Solar Energy, vol. 207, pp. 336–346, 2020.

[42] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 398–417, 2009.

[43] Y. Wang, Z. Cai, and Q. Zhang, “Enhancing the search ability of differential evolution through orthogonal crossover,” Information Sciences, vol. 185, no. 1, pp. 153–177, 2012.

[44] 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.

[45] Z. Xu, S. Gao, H. Yang, and Z. Lei, “SCJADE: Yet another state-of-the-art differential evolution algorithm,” IEEE Transactions on Electrical and Electronic Engineering, vol. 16, no. 4, pp. 644–646, 2021.

[46] Y. Yu, K. Wang, T. Zhang, Y. Wang, C. Peng, and S. Gao, “A population diversity-controlled differential evolution for parameter estimation of solar photovoltaic models,” Sustainable Energy Technologies and Assessments, vol. 51, pp. 101938, 2022.

[47] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, “Novel mutation strategy for enhancing shade and Ishade algorithms for global numerical optimization,” Swarm and Evolutionary Computation, vol. 50, pp. 100455, 2019.

[48] P. P. Biswas, P. N. Suganthan, G. Wu, and G. A. Amaratunga, “Parameter estimation of solar cells using datasheet information with the application of an adaptive differential evolution algorithm,” Renewable Energy, vol. 132, pp. 425–438, 2019.

[49] R. Tanabe and A. Fukunaga, “Success-history based parameter adaptation for differential evolution,” in 2013 IEEE Congress on Evolutionary Computation. IEEE, 2013, pp. 71–78.

[50] J. Zhang and A. C. Sanderson, “Jade: Adaptive differential evolution with optional external archive,” IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 945–958, 2009.

[51] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, “Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems,” IEEE Transactions on Evolutionary Computation, vol. 16, no. 6, pp. 646–657, 2006.

[52] B. Dorronsoro and P. Bouvry, “Improving classical and decentralized differential evolution with new mutation operator and population topologies,” IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 67–98, 2011.

[53] 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, 2021.

[54] Z.-H. Zhan, Z.-J. Wang, H. Jin, and J. Zhang, “Adaptive distributed differential evolution,” IEEE Transactions on Cybernetics, vol. 50, no. 11, pp. 4633–4647, 2020.

[55] Z. Meng and C. Yang, “Two-stage differential evolution with novel parameter control,” Information Sciences, vol. 596, pp. 321–342, 2022.
TABLE 9: Experimental results of PAIDDE on F2, F9, F15 and F27 with 30, 50, and 100 dimensions.

| Algorithm | F1 | F2 | F3 | F4 | F5 |
|-----------|----|----|----|----|----|
| DEGoS     | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 5.856E+01 ± 1.056E+14 | 6.853E+00 ± 1.487E+00 | 4.562E-09 ± 2.994E-08 |
| SHADE     | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 7.196E+00 ± 1.122E+00 | 0.000E+00 ± 0.000E+00 | 4.184E-02 ± 2.532E-02 |
| DEGoS     | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 2.131E+00 ± 3.217E+01 | 1.165E+02 ± 3.157E+02 | 1.713E+02 ± 2.015E+02 |
| SHADE     | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 3.571E+00 ± 4.694E+00 | 3.969E+00 ± 5.096E+00 | 1.946E+00 ± 2.384E+00 |
| DEGoS     | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 2.131E+00 ± 3.217E+01 | 1.165E+02 ± 3.157E+02 | 1.713E+02 ± 2.015E+02 |
| SHADE     | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 3.571E+00 ± 4.694E+00 | 3.969E+00 ± 5.096E+00 | 1.946E+00 ± 2.384E+00 |
| DEGoS     | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 2.131E+00 ± 3.217E+01 | 1.165E+02 ± 3.157E+02 | 1.713E+02 ± 2.015E+02 |
| SHADE     | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 3.571E+00 ± 4.694E+00 | 3.969E+00 ± 5.096E+00 | 1.946E+00 ± 2.384E+00 |

[56] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, “Opposition-based differential evolution,” IEEE Transactions on Evolutionary computation, vol. 12, no. 1, pp. 64–79, 2008.

[57] U. K. Chakraborty, S. K. Das, and T. E. Abbott, “Energy-efficient routing in hierarchical wireless sensor networks using differential-evolution-based memetic algorithm,” in 2012 IEEE Congress on Evolutionary Computation. IEEE, 2012, pp. 1–8.

[58] E. Mininni and F. Neri, “Estimation distribution differential evolution,” in European Conference on the Applications of Evolutionary Computation. Springer, 2010, pp. 522–531.

[59] G.-G. Wang and Y. Tan, “Improving metaheuristic algorithms with information feedback models,” IEEE Transactions on Cybernetics, vol. 49, no. 2, pp. 542–555, 2019.

[60] J. Peng, Y. Li, H. Kang, Y. Shen, X. Sun, and Q. Chen, “Impact of population topology on particle swarm optimization and its variants: An information propagation perspective,” Swarm and Evolutionary Computation, vol. 69, p. 100990, 2022.

[61] Y. Lou, S. Y. Yuen, and G. Chen, “Non-revisiting stochastic search revisited: Results, perspectives, and future directions,” Swarm and Evolutionary Computation, vol. 61, p. 100828, 2021.

[62] H. Yang, S. Tao, Z. Zhang, Z. Cai, and S. Gao, “Spatial information feedback models,” IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 2202–2215, 2013.
| Algorithm | F1 | F2 | F3 | F4 | F5 |
|-----------|----|----|----|----|----|
| PAIDDE    | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 5.380E+01 ± 1.108E+14 | 6.853E+00 ± 1.487E+00 | 4.562E-09 ± 2.399E-08 |
| DGA      | 4.141E+02 ± 2.304E+03 | 1.902E+02 ± 9.360E+01 | 1.373E+04 ± 2.746E+02 | 1.628E+00 ± 2.058E+01 |
| MGA      | 3.925E+02 ± 4.261E+03 | 5.000E+01 ± 8.000E+00 | 1.190E+04 ± 2.067E+01 | 7.157E+00 ± 2.574E+01 | 1.628E+00 ± 2.058E+01 |
| ALGSA    | 4.383E+02 ± 4.216E+03 | 3.955E+04 ± 1.141E+04 | 2.064E+01 ± 6.918E-01 | 1.627E+00 ± 3.799E+00 | 1.298E+00 ± 6.166E+07 |
| AIDDE    | 4.141E+02 ± 2.304E+03 | 1.902E+02 ± 9.360E+01 | 1.373E+04 ± 2.746E+02 | 1.628E+00 ± 2.058E+01 | 9.636E+00 ± 1.576E+01 |
| GLPSO    | 1.645E+03 ± 1.614E+03 | 2.180E+02 ± 8.526E+01 | 2.784E+01 ± 8.337E+02 | 1.754E+00 ± 2.125E+02 | 5.108E+00 ± 1.075E+03 |
| SE       | 1.300E+01 ± 1.875E+00 | 9.341E+04 ± 3.117E+04 | 1.797E+01 ± 2.172E+01 | 4.244E+00 ± 7.847E+01 | 1.274E+01 ± 1.597E+03 |
| SASS     | 0.000E+00 ± 0.000E+00 | 0.000E+00 ± 0.000E+00 | 2.600E+00 ± 2.964E+01 | 9.300E+00 ± 1.801E+02 | 4.562E-09 ± 2.399E-08 |

**TABLE 10:** Experimental results of PAIDDE and other comparison algorithms on 29 CEC2017 benchmark functions ($D = 30$).
### TABLE 11: Experimental results of PAIDDE and other variants of DE on 29 CEC2017 benchmark functions ($D = 50$).

| Algorithm       | F1 | F2 | F3 | F4 | F5 |
|-----------------|----|----|----|----|----|
| **PAIDDE**      | $0.000E+00$ | $0.000E+00$ | $0.000E+00$ | $5.199E+00$ | $1.344E+01$ |
| **DE/ES**       | $1.135E+01$ | $5.317E+01$ | $1.293E+01$ | $2.088E+04$ | $5.231E+04$ |
| **CAJDE**       | $0.000E+00$ | $0.000E+00$ | $3.129E+01$ | $4.355E+01$ | $5.372E+01$ |
| **SCADE**       | $0.000E+00$ | $0.000E+00$ | $8.228E+00$ | $1.565E+00$ | $5.085E+00$ |
| **IMODE**       | $1.570E+02$ | $1.631E+03$ | $9.556E+01$ | $3.407E+02$ | $3.874E+02$ |
| **SHADE**       | $0.000E+00$ | $0.000E+00$ | $8.000E+00$ | $4.058E+01$ | $4.422E+01$ |
| **PAIDDE**      | $6.458E+01$ | $1.382E+02$ | | | |
| **DE/ES**       | $1.397E+02$ | $1.913E+02$ | $6.022E+01$ | $5.638E+01$ | $5.119E+00$ |
| **CAJDE**       | $1.001E+02$ | $7.469E+00$ | $5.377E+00$ | $1.261E+00$ | $3.683E+00$ |
| **SCADE**       | $9.742E+01$ | $8.015E+00$ | $5.166E+00$ | $1.098E+00$ | $3.955E+00$ |
| **IMODE**       | $2.868E+02$ | $2.631E+03$ | $9.179E+00$ | $6.707E+00$ | $2.626E+00$ |
| **SHADE**       | $8.884E+01$ | $7.250E+00$ | $4.407E+01$ | $5.856E+01$ | $5.017E+01$ |
| **PAIDDE**      | $2.110E+02$ | $5.108E+02$ | | | |
| **DE/ES**       | $7.790E+01$ | $3.397E+00$ | $2.459E+01$ | $5.803E+01$ | $3.720E+01$ |
| **CAJDE**       | $5.522E+01$ | $2.824E+00$ | $2.322E+01$ | $6.762E+00$ | $4.291E+00$ |
| **SCADE**       | $5.950E+01$ | $4.984E+00$ | $3.494E+01$ | $2.364E+01$ | $3.071E+01$ |
| **IMODE**       | $2.214E+01$ | $5.060E+00$ | $4.891E+01$ | $2.928E+01$ | $3.169E+00$ |
| **SHADE**       | $4.917E+03$ | $8.332E+03$ | $2.575E+02$ | $1.745E+02$ | $2.170E+02$ |
| **PAIDDE**      | $2.708E+04$ | $7.176E+01$ | | | |
| **DE/ES**       | $4.284E-01$ | $3.646E+01$ | $2.565E+01$ | $2.747E+01$ | $2.137E+00$ |
| **CAJDE**       | $6.279E+00$ | $1.159E+02$ | $1.304E+02$ | $1.496E+02$ | $4.406E+02$ |
| **SCADE**       | $5.274E+00$ | $1.497E+02$ | | | |
| **IMODE**       | $2.111E+00$ | $6.056E+01$ | $2.573E+01$ | $6.214E+01$ | $5.398E+00$ |
| **SHADE**       | $1.519E+01$ | $2.452E+01$ | | | |
| **PAIDDE**      | $1.485E+03$ | $3.014E+00$ | | | |
| **DE/ES**       | $5.959E+00$ | $5.087E+00$ | $4.898E+02$ | $5.425E+02$ | $5.182E+02$ |
| **CAJDE**       | $4.853E+00$ | $1.241E+01$ | $4.668E+01$ | $5.814E+01$ | $5.199E+01$ |
| **SCADE**       | $2.985E+00$ | $1.689E+01$ | $4.742E+01$ | $3.814E+01$ | $3.228E+01$ |
| **IMODE**       | $6.906E+00$ | $1.972E+01$ | $1.306E+01$ | $9.283E+01$ | $7.510E+01$ |
| **SHADE**       | $1.081E+03$ | $3.994E+01$ | $4.070E+02$ | $5.696E+02$ | $5.298E+02$ |
| **PAIDDE**      | $9.584E+03$ | $6.182E+03$ | | | |
| **DE/ES**       | $9.350E+02$ | $1.735E+02$ | $4.947E+01$ | $3.526E+01$ | $6.039E+01$ |
| **CAJDE**       | $9.614E+02$ | $2.671E+01$ | $4.857E+01$ | $3.248E+01$ | $4.142E+01$ |
| **SCADE**       | | | | | |
| **IMODE**       | $1.884E+02$ | $2.084E+02$ | | | |
| **SHADE**       | $5.329E+02$ | $2.971E+02$ | | | |

---

[80] C. Wang, W. Pedrycz, Z. Li, and M. Zhou, “Residual-driven fuzzy c-means clustering for image segmentation,” IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 4, pp. 876–889, 2020.

[81] X. Guo, M. Zhou, A. Abusorrah, F. Alshokhri, and K. Sdraouli, “Disassemble sequence planning: a survey,” IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 7, pp. 1308–1324, 2020.

[82] J. J. Cheng, C. R. Cao, M. C. Zhou, C. Liu, S. C. Gao, and C. J. Jiang, “A dynamic evolution mechanism for ioc evolution in an urban scene,” IEEE Internet of Things Journal, vol. 8, no. 9, pp. 7521–7530, 2021.

[83] L. Wu, J. Liu, S. Vazquez, and S. K. Mazumder, “Sliding mode control in power converters and drives: A review,” IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 3, pp. 392–406, 2022.

[84] J. He and X. Yao, “Towards an analytic framework for analysing the computation time of evolutionary algorithms,” Artificial Intelligence, vol. 145, no. 1-2, pp. 59–97, 2003.
| Algorithm | F1 | F2 | F3 | F4 | F5 | F6 |
|-----------|----|----|----|----|----|----|
| PAIDDE    | 0.000±0 | 0.000±0 | 0.000±0 | 6.719E1±0 | 4.991E1±0 | 1.344E1±0 | 2.279E0±0 | 2.808E-04±0.000E+00 |
| DGA      | 4.194E-03 | 2.878E+03 | 1.914E5±0 | 9.688E0±0 | 8.624E0±0 | 5.052E0±0 | 1.356E0±0 | 2.645E0±0 | 3.054E0±0 | 3.020E0±0 | 2.351E0±0 | 2.351E0±0 |
| MLGSA    | 6.725E+04 | 6.389E+04 | 7.084E0±0 | 2.258E4±0 | 1.257E0±0 | 6.653E0±0 | 8.505E0±0 | 3.883E0±0 | 2.052E0±0 | 2.512E0±0 | 3.104E0±0 | 2.294E0±0 |
| ALGSA    | 7.009E+04 | 7.126E+04 | 7.008E0±0 | 2.952E4±0 | 1.239E0±0 | 7.491E0±0 | 9.825E0±0 | 3.910E0±0 | 2.052E0±0 | 2.512E0±0 | 3.104E0±0 | 2.294E0±0 |
| CMASE    | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 |
| GLPSO    | 1.028±00 | 5.278±00 | 7.788±00 | 1.176±00 | 8.040±00 | 2.210±00 | 3.492±00 | 2.521±00 | 2.319±00 | 3.172±00 | 2.521±00 | 2.319±00 |
| SR       | 1.707±00 | 2.158±00 | 4.195±00 | 8.513±00 | 1.145±00 | 2.821±00 | 1.582±00 | 1.602±00 | 2.532±00 | 3.312±00 | 2.532±00 | 3.312±00 |
| SASS     | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 | 0.000±0 |

TABLE 12: Experimental results of PAIDDE and other comparison algorithms on 29 CEC2017 benchmark functions ($D = 50$).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
| Algorithm | F1 | F2 | F3 | F4 | F5 |
|-----------|----|----|----|----|----|
| PAIDDE | 0.0000E+00 | 1.117E-08 | 1.160E-08 | 1.832E+01 | 2.399E+01 |
| DEGeS | 6.127E+03 | 6.74E+03 | 1.497E+03 | 2.588E+02 | 4.30E+01 |
| CAJDE | 0.0000E+00 | 1.084E+00 | 1.365E+00 | 6.048E+01 | 4.702E+01 |
| SCJADE | 0.0000E+00 | 3.475E+00 | 5.155E+00 | 8.721E+01 | 7.449E+01 |
| IMODE | 1.90E+03 | 1.121E+03 | 7.058E+03 | 9.650E+01 | 9.87E+01 |
| SHADE | 0.0000E+00 | 1.430E+06 | 3.410E+01 | 9.299E+01 | 6.571E+01 |
| Algorithm | F6 | F7 | F8 | F9 | F10 |
| PAIDDE | 1.494E+02 | 6.274E+00 | 5.073E+01 | 1.013E+01 | 6.108E+01 |
| DEGeS | 5.131E+02 | 2.822E+02 | 2.174E+02 | 2.717E+03 | 8.549E+03 |
| CAJDE | 2.783E+02 | 3.097E+01 | 1.478E+01 | 1.168E+01 | 1.012E+02 |
| SCJADE | 2.759E+02 | 2.169E+01 | 1.512E+01 | 9.062E+01 | 6.821E+01 |
| IMODE | 2.216E+03 | 5.398E-02 | 1.705E+01 | 3.260E+01 | 5.700E+01 |
| SHADE | 2.452E+02 | 2.860E+01 | 1.380E+01 | 2.702E+01 | 1.105E+04 |
| Algorithm | F11 | F12 | F13 | F14 | F15 |
| PAIDDE | 1.740E+04 | 1.908E+03 | 2.111E+02 | 3.247E+03 | 2.147E+01 |
| DEGeS | 1.693E+03 | 7.008E+02 | 1.113E+01 | 2.176E+02 | 6.595E+02 |
| CAJDE | 1.951E+04 | 1.514E+04 | 1.518E+03 | 5.852E+04 | 3.997E+04 |
| SCJADE | 1.757E+04 | 7.242E+03 | 2.949E+03 | 3.289E+03 | 6.685E+02 |
| IMODE | 3.148E+05 | 5.788E+04 | 6.275E+02 | 1.608E+02 | 5.115E+02 |
| SHADE | 2.071E+04 | 7.781E+03 | 2.976E+03 | 3.221E+03 | 9.045E+03 |
| Algorithm | F16 | F17 | F18 | F19 | F20 |
| PAIDDE | 1.352E+05 | 1.429E+05 | 2.198E+02 | 9.057E+01 | 1.782E+01 |
| DEGeS | 1.674E+03 | 1.252E+03 | 1.297E+01 | 6.703E+04 | 3.903E+04 |
| CAJDE | 1.979E+03 | 2.279E+03 | 2.255E+03 | 1.894E+04 | 2.949E+04 |
| SCJADE | 1.610E+03 | 2.754E+03 | 2.058E+03 | 8.248E+03 | 1.131E+04 |
| IMODE | 3.927E+03 | 6.080E+02 | 2.727E+02 | 6.002E+02 | 3.757E+02 |
| SHADE | 1.844E+05 | 2.520E+02 | 7.358E+04 | 1.310E+04 | 1.218E+04 |
| Algorithm | F21 | F22 | F23 | F24 | F25 |
| PAIDDE | 1.139E+04 | 4.935E+02 | 5.629E+02 | 9.057E+03 | 1.035E+03 |
| DEGeS | 2.605E+03 | 4.751E+03 | 7.135E+03 | 1.058E+03 | 3.023E+01 |
| CAJDE | 1.144E+04 | 6.598E+02 | 6.508E+02 | 1.036E+03 | 2.725E+01 |
| SCJADE | 1.391E+04 | 6.501E+02 | 6.550E+02 | 1.738E+03 | 4.052E+03 |
| IMODE | 1.668E+03 | 1.300E+02 | 2.014E+02 | 6.022E+02 | 6.361E+02 |
| SHADE | 1.127E+04 | 5.796E+02 | 6.366E+02 | 1.394E+04 | 2.224E+03 |
| Algorithm | F26 | F27 | F28 | F29 | F30 |
| PAIDDE | 6.308E+02 | 1.582E+01 | 6.573E+02 | 1.251E+03 | 1.056E+03 |
| DEGeS | 6.543E+03 | 2.429E+01 | 5.527E+02 | 2.397E+01 | 1.956E+01 |
| CAJDE | 7.444E+03 | 4.381E+01 | 5.532E+02 | 1.026E+03 | 2.028E+01 |
| SCJADE | 7.419E+04 | 3.915E+01 | 5.311E+02 | 7.126E+03 | 2.202E+02 |
| IMODE | 1.610E+03 | 2.649E+02 | 4.944E+02 | 8.506E+01 | 4.756E+03 |

**TABLE 13:** Experimental results of PAIDDE and other variants of DE on 29 CEC2017 benchmark functions \((D = 100)\).
| Algorithm | F1 | F2 | F3 | F4 | F5 |
|-----------|-----|-----|-----|-----|-----|
| PAIDDE    | 0.000E+00 | 0.000E+00 | 1.174E+08 | 1.664E+08 | 1.832E+07 | 2.399E+01 | 6.464E+01 | 4.513E+00 | 7.439E-03 | 5.643E-03 |
| DGSA      | 4.091E+03 | 3.076E+03 | 3.246E+03 | 2.079E+04 | 2.275E+02 | 4.529E+01 | 4.798E+02 | 4.506E+01 | 2.151E+01 | 4.052E+00 |
| MLGSA     | 1.016E+03 | 8.754E+03 | 1.485E+03 | 1.529E+04 | 3.589E+02 | 1.864E+01 | 2.683E+02 | 2.284E+01 | 1.861E+01 | 1.853E+00 |
| ALGS      | 1.177E+05 | 7.909E+05 | 2.508E+04 | 2.095E+05 | 2.011E+00 | 7.555E+01 | 1.179E+04 | 2.213E+00 | 1.103E+04 | 5.643E-03 |
| CMAES     | 0.000E+00 | 0.000E+00 | 0.000E+00 | 0.000E+00 | 1.122E+02 | 1.351E+01 | 2.109E+03 | 2.318E+00 | 1.351E+00 | 5.643E-03 |
| GLPSO     | 9.859E+04 | 2.097E+05 | 1.208E+05 | 4.844E+04 | 3.107E+02 | 5.903E+01 | 3.946E+02 | 6.223E+01 | 1.406E+01 | 1.481E-01 |
| ML       | 4.270E+03 | 4.270E+03 | 2.579E+04 | 2.338E+04 | 2.346E+00 | 2.017E+00 | 2.711E+00 | 8.282E+00 | 2.720E-02 | 5.925E-04 |

**TABLE 14: Experimental results of PAIDDE and other comparison algorithms on 29 CEC2017 benchmark functions (D = 100).**
### TABLE 15: Experimental results of PAIDDE and other variants of DE on 22 real-world problems.

| Algorithm | F1 | F2 | F3 | F4 | F5 |
|-----------|----|----|----|----|----|
| PAIDDE    | 5.82E-01 ± 2.228E+00 | 2.75E-01 ± 5.170E-01 | 1.51E-05 ± 2.160E-19 | 1.85E+01 ± 3.179E+00 | -3.68E+01 ± 1.580E-02 |
| DEGO5     | 5.37E-01 ± 2.830E+00 | 2.96E-01 ± 6.764E+00 | 1.15E+01 ± 2.071E+00 | 2.16E+00 ± 3.245E+00 | -2.87E+00 ± 6.048E+00 |
| CMADE     | 2.94E+00 ± 2.842E+00 | 2.71E+00 ± 1.606E+00 | 1.15E+01 ± 2.047E+00 | 1.93E+01 ± 2.926E+00 | -3.86E+00 ± 3.290E+00 |
| MGS5      | 7.95E-01 ± 3.944E-01 | 9.63E-01 ± 3.186E-01 | 2.51E+01 ± 3.159E+00 | 6.20E+00 ± 4.534E+00 | -6.38E+00 ± 1.953E+00 |
| MODE      | 3.20E+00 ± 3.273E+00 | 2.65E+00 ± 1.409E+00 | 1.15E+01 ± 2.011E+00 | 2.22E+01 ± 4.945E+00 | -2.68E+01 ± 4.945E+00 |
| SHADE     | 2.93E+00 ± 3.316E+00 | 2.66E+00 ± 7.88E+01 | 1.15E+01 ± 2.848E+00 | 1.58E+01 ± 2.760E+00 | -3.61E+01 ± 1.092E+01 |

### TABLE 16: Experimental results of PAIDDE and other comparison algorithms on 22 real-world problems.

| Algorithm | F1 | F2 | F3 | F4 | F5 |
|-----------|----|----|----|----|----|
| PAIDDE    | 8.52E-01 ± 2.288E+00 | 2.75E-01 ± 5.170E-01 | 1.51E-05 ± 2.160E-19 | 1.85E+01 ± 3.179E+00 | -3.68E+01 ± 1.580E-02 |
| DEGO5     | 5.37E-01 ± 2.830E+00 | 2.96E-01 ± 6.764E+00 | 1.15E+01 ± 2.071E+00 | 2.16E+00 ± 3.245E+00 | -2.87E+00 ± 6.048E+00 |
| CMADE     | 2.94E+00 ± 2.842E+00 | 2.71E+00 ± 1.606E+00 | 1.15E+01 ± 2.047E+00 | 1.93E+01 ± 2.926E+00 | -3.86E+00 ± 3.290E+00 |
| MGS5      | 7.95E-01 ± 3.944E-01 | 9.63E-01 ± 3.186E-01 | 2.51E+01 ± 3.159E+00 | 6.20E+00 ± 4.534E+00 | -6.38E+00 ± 1.953E+00 |
| MODE      | 3.20E+00 ± 3.273E+00 | 2.65E+00 ± 1.409E+00 | 1.15E+01 ± 2.011E+00 | 2.22E+01 ± 4.945E+00 | -2.68E+01 ± 4.945E+00 |
| SHADE     | 2.93E+00 ± 3.316E+00 | 2.66E+00 ± 7.88E+01 | 1.15E+01 ± 2.848E+00 | 1.58E+01 ± 2.760E+00 | -3.61E+01 ± 1.092E+01 |