Multiple Elite Individual Guided Piecewise Search-Based Differential Evolution

Shubham Gupta, Shitu Singh, Rong Su, *Senior Member, IEEE,* Shangce Gao, *Senior Member, IEEE,* Jagdish Chand Bansal

***Abstract*—The differential evolution (DE) algorithm relies mainly on mutation strategy and control parameters' selection. To take full advantage of top elite individuals in terms of fitness and success rates, a new mutation operator is proposed. The control parameters such as scale factor and crossover rate are tuned based on their success rates recorded over past evolutionary stages. The proposed DE variant, MIDE, performs the evolution in a piecewise manner, i.e., after every predefined evolutionary stages, MIDE adjusts its settings to enrich its diversity skills. The performance of the MIDE is validated on two different sets of benchmarks: CEC 2014 and CEC 2017 (special sessions & competitions on real-parameter single objective optimization) using different performance measures. In the end, MIDE is also applied to solve constrained engineering problems. The efficiency and effectiveness of the MIDE are further confirmed by a set of experiments.**

***Index Terms*—Metaheuristic algorithms, Differential Evolution, Mutation operator, Control parameters.**

I. INTRODUCTION

METAHEURISTIC algorithms (MAs) are stochastic search algorithms inspired by several sources such as nature, evolution, physics, biology, and so forth. These algorithms are quite popular among researchers due to their flexibility, derivative-free search scheme, and adaptability to solve a wide range of optimization problems [1, 2, 3, 4, 5]. Within their search procedures, two fundamental features called exploration and exploitation of solution space play an important role. Exploration operates on the whole solution space to discover new promising areas, while exploitation examines a local portion of the solution space and extracts useful information from discovered search areas. An appropriate balance between these two features is a desirable goal in the MAs. Some of the well-known and efficient MAs are particle swarm optimization (PSO) [6], differential evolution (DE) [7], ant colony optimization (ACO) [8], artificial bee colony [9] algorithm, grey wolf optimizer (GWO) [10], harmony search (HS) [11], etc. Compared with deterministic gradient-based optimization methods, MAs have more benefits in terms of easy implementation, gradient-free search procedure, and favorable global search ability. Among these algorithms, the DE algorithm has gained increasing attention among researchers because of its easy implementation and excellent performance.

The DE, proposed by Storn and Price [7], mimics the concept of Darwin's theory of evolution in nature [12] and therefore, this algorithm belongs to the field of evolutionary algorithms (EAs). Within these algorithms, two search concepts are included: 1) survival of the fittest (the good solution will pass to the next generation, while inferior solutions will die), and 2) inheritance (the features of solutions will be passed to the next generation of evolution). In the DE, three fundamental evolutionary operators called mutation, crossover, and selection are used to evolve the population.

Finding a suitable mutation operator in the DE, which can provide a promising direction of search and can balance the exploration and exploitation features in an optimization process is a challenging task. An imbalance between these two features causes stagnation at local optima and premature convergence issues in the DE [13, 14]. Moreover, the control parameters of the DE also affect the performance of the DE. Using appropriate guidance of search and historical evolution information, the performance of the DE can be enhanced. By aiming these facts, in this paper, we propose a new variant of the DE named MIDE. In the MIDE, a new mutation operator and parameter selection approaches are proposed. In summary, the contribution of this study is organized as follows:

1) A new mutation operator in the MIDE is proposed, which includes computationally efficient guidance based on multiple elite individuals of the population and one of the most successful individuals in some recent evolutionary generations. This guiding behavior provides a balancing ability to the MIDE between exploration and exploitation. The inclusion of an archive that stores the discarded individuals in past generations helps to determine the parent individual in the mutation operator. This archive helps to maintain the diversity of the MIDE population, which is desired to avoid local optimal solutions during the search.
2) Diversified and historical information-based parameter selection scheme for control parameters called scale factor and crossover rate are determined using Gaussian and Normal distribution, and opposition-based learning (OBL) scheme. This approach provides enhanced exploration strength to the MIDE and avoids the situation of getting trapped at local optimal solutions.
3) A dynamic reduction in population size is adopted in the

This work is supported by the A*STAR under its RIE2020 Advanced Manufacturing and Engineering (AME) Industry Alignment Fund - Pre-Positioning (IAF-PP) (Award A19D6a0053). (Corresponding authors: Shangce Gao, and Shubham Gupta.)

S. Gupta, and R. Su are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (email: shubham.gupta@ntu.edu.sg;rsu@ntu.edu.sg)

S. Singh, and J. C. Bansal are with the Department of Mathematics, South Asian University, New Delhi 110021, India (email: singh.shitu93@gmail.com;jcbansal@sau.ac.in)

S. Gao is with Faculty of Engineering, University of Toyama, Toyama-shi 930-8555, Japan (email: gaosc@eng.u-toyama.ac.jp)

MIDE, which discards the unfavorable individuals from the optimization process. This strategy helps to save computational resources and keeps the search procedure in the most promising parts of the solution space.

4) Different components of the MIDE are reset and controlled in a piecewise manner, i.e., the optimization process of the MIDE is performed in different pieces. This strategy aims to perform the search in such a way that it can extract the useful information collected at different evolutionary stages.
5) The experimental validation of the MIDE is carried out on standard benchmark sets of the IEEE CEC 2014 [15] and CEC 2017 [16] using various performance metrics. These problem sets include several difficulty levels of functions such as unimodal, multimodal, hybrid, and composite. These problems also include non-separable, asymmetrical, and non-differentiable functions.

The remaining of the paper is structured as follows: Section II explains the framework of the classical DE with a summary of literature work conducted to improve its search performance. The description of the proposed MIDE algorithm is discussed in Section III with its computational steps. In Section IV, extensive experiments for validation of the MIDE are carried out on two different benchmark sets and four constrained engineering problems. In this section, the performance of the MIDE is also compared with other state-of-the-art algorithms. Finally, conclusions are drawn in Section V.

II. Related works

In this section, the classical DE algorithm is introduced first. Then, the related work to the DE will be reviewed.

A. Classical DE Algorithm

The classical DE algorithm is proposed in [7] for solving single-objective bound-constrained optimization problems. DE evolves the population of N_P individuals using three elementary operators namely, mutation, crossover, and selection. In the search procedure of the DE, first, a population of N_P individuals is initialized randomly within the solution space. Then a mutation operator is applied to produce a mutant vector. Some of the commonly used mutation operators are stated as follows:

DE/rand/1:

$$V_i^g = X_{r_1}^g + F \times (X_{r_2}^g - X_{r_3}^g) \tag{1}$$

DE/best/1:

$$V_i^g = X_b^g + F \times (X_{r_2}^g - X_{r_3}^g) \tag{2}$$

DE/rand/2:

$$V_i^g = X_{r_1}^g + F \times (X_{r_2}^g - X_{r_3}^g) + F \times (X_{r_4}^g - X_{r_5}^g) \tag{3}$$

DE/best/2:

$$V_i^g = X_b^g + F \times (X_{r_1}^g - X_{r_2}^g) + F \times (X_{r_3}^g - X_{r_4}^g) \tag{4}$$

where r_1, r_2, r_3, and r_4 are randomly selected distinct integers from the range $[1, N_P]$. X_b^g represents the fittest individual obtained till g^{th} generation. The parameter F, known as the scale factor, is a control parameter used to control the magnitude of difference vector(s) involved in the corresponding mutation operators.

After generating a mutant vector $V_i^g = (v_{i,1}^g, v_{i,2}^g, \ldots, v_{i,d}^g)$ corresponding to a target vector $X_i^g = (x_{i,1}^g, x_{i,2}^g, \ldots, x_{i,d}^g)$, crossover operator is employed to generate a trial vector $U_i^g = (u_{i,1}^g, u_{i,2}^g, \ldots, u_{i,d}^g)$. The binomial crossover is commonly used in the DE, which is stated in (5).

$$u_{i,j}^g = \begin{cases} v_{i,j}^g & \text{if } rand_{i,j}^g < C_R \text{ or } j = j_{rand} \\ x_{i,j}^g & \text{otherwise} \end{cases} \tag{5}$$

where C_R is a crossover rate and j_{rand} is a randomly selected integer in the range $[1, d]$. If U_i^g falls outside of the solution space, it is passed to a rebounding mechanism, which resets the components of the trial vector that falls out of the solution space boundary.

When the feasible trial vector is generated successfully, a selection operator, given by (6), is used to select the best-fitted individual for the next generation by comparing the fitnesses of the original target vector and trial vector.

$$X_i^{g+1} = \begin{cases} U_i^g & \text{if } f(U_i^g) \le f(X_i^g) \\ X_i^g & \text{otherwise} \end{cases} \tag{6}$$

where $f(X)$ indicates the fitness of the individual vector X.

B. Literature Review

Although the classical DE has performed excellent on a wide range of optimization problems, it suffers from some serious issues such as slow convergence, stagnation at local optima, and premature convergence [13, 14, 17, 18]. Therefore, researchers have put their efforts into developing new improved DE variants. We have categorized these developments into two sections as follows:

1) Trail Vector Generation Strategies: In the literature, to improve the search performance of the DE, several mutation operators are designed. For example, Fan and Lampinen [19] have introduced a trigonometric mutation operator intending to achieve a better trade-off between the convergence rate and the robustness. In their work, the authors have used this mutation along with the DE/rand/1 mutation operator. A selection probability is used to manage the generation of mutation vector either by the trigonometric mutation operator or by DE/rand/1 mutation operator. In [20], a new mutation rule DE/current-to-pbest/1 is designed, where $pbest$ indicates the participation of one of the top-fitted individuals from $100{\cdot}p\%$ individuals of the population as a parent vector of the first difference vector in the mutation operator. In this algorithm, an optional archive is used to store the inferior solutions discarded through the selection operator. This archive is utilized to select the parent vector in the second difference vector of the mutation operator. In [21], a ranking-based mutation operator is designed in which parents in the mutation operators are proportionally selected according to their fitness rankings in the current population. In [22], fitness and diversity-based selection have been taken into account at the same time to select the parent individuals of the mutation operator. The reason for diversity-based selection is to improve the exploration ability of the DE. Later, Cheng et al. [23] introduced another metric to define diversity in terms of fitness to overcome the computational burden of evaluating the diversity measure. Zhang and Yuen [24] have designed a directional mutation operator, in which a pool of difference vectors is calculated when fitness is improved at a generation. This pool is utilized to guide the individual's search procedure in the next generation. To enhance the performance of DE, in [25], a Gaussian mutation operator and a new modified mutation operator based on the worst individual are defined to collaboratively produce new mutant vectors. In addition to these mutation operators, other mutation operators are also developed in the literature to enhance the DE performance such as the Laplace mutation operator [26], wavelet theory-based mutation [27], neighborhood-dependent mutation operator [28], parent-centric mutation [29], intersect mutation operator [30], etc.

Since different mutation operators have their advantages and limitations, researchers have combined different mutation operators to take their advantages in improving the DE performance. For example, Ameca-Alducin et al. [31] have combined two base mutation operators namely, DE/rand/1 and DE/best/1 to propose a new mutation strategy. In [32], authors have created a pool of mutation operators (DE/rand/1, DE/best/2 and DE/current to rand/1) to produce mutant vectors. The mutation operator producing better trial vector survive while those fail to produce better trial vector are reinitialized. Wang et al. [33] have utilized DE/rand/1, DE/rand2 and DE/current-to-rand/1 mutation operators to produce three different trial vectors and then among these three trial vectors, the fittest trial vector is passed to the selection operator.

Although most of the DE variants adopt the binomial crossover to generate a trail vector, some researchers have also developed various crossover strategies like parent centric crossover [34], orthogonal crossover [35], hybrid linkage crossover [36], and so forth.

2) Tuning of Control Parameters: In the DE, mainly three parameters, F, C_R, and N_P, need to be set for its execution. Storn and Price [37] have suggested the size of the population to be $5d$ to $10d$ and the value of F as 0.5. Later, Gämperle et al. [38] suggested the value of N_P as a range of $[3d, 8d]$, F as 0.6, and C_R as $[0.3, 0.9]$. According to Storn and Price [7], the suitable value of C_R lies in the range $[0.5, 1]$. Ronkkonen et al. [39] have suggested a value of F in the range $[0.4, 0.95]$. For separable objective function, they suggested the C_R value in the range $[0, 0.2]$, while for non-separable function, this value should lie in the range $[0.9, 1]$

Zhao et al. [40] have used the Cauchy and Normal distributions to update the values of F and C_R, respectively. In [41], authors have developed a self-adaptive DE variant, in which F and C_R are updated by learning from their previous generation experiences in improving individuals. Ghosh et al. [42] have defined a simple and effective fitness-based adaptation scheme for the F and C_R. [43] have used auto-adaptive control parameters to overcome the local optima stagnation issue. [44, 45, 46] have utilized a self-adaptive mechanism to tune the control parameters F and C_R. [47] have used the time-varying control parameter strategy to update F and C_R. In some variants of the DE [32, 33], a pool of control parameter settings is created, and later, during the evolutionary process, based on the success rate the parameters are adopted from the pool.

III. Proposed MIDE

Like other metaheuristic algorithms, DE also needs to establish a proper trade-off between exploitation and exploration. To accomplish this, the most effective tools are mutation operator and selection of control parameters. The mutation also provides directions of search within the evolutionary search procedure. In the literature, most of the DE variants take a set of elite or random individuals to set as parent individuals of the mutation operator. However, with this adjustment, either the DE shows biasedness towards the elite individual or creates high randomness in evolving the population over generations. The biasedness towards the elite individuals degrades the diversity skill of the algorithm, while the high randomness slows down the convergence speed of the algorithm. DE/best/1 and DE/rand/1 mutation schemes are examples of such scenarios, where the DE shows low exploration and exploitation, respectively. This inappropriate synergy between exploration and exploitation may cause serious concerns of local optima stagnation and/or premature convergence.

Although some existing DE variants discussed in the previous section have provided efficient mutation strategies to provide significant advantages over complex optimization problems, finding a suitable direction of search is still a challenging task within the mutation operator. Along with the mutation operator, the control parameter setting is also responsible for establishing a proper balance between the exploration and exploitation features in the DE. An imbalance between these features creates a problem of stagnation at local optima and premature convergence in the DE [13, 14]. Relying on all the previous generations of evolution during the search procedure may not provide a promising evolutionary search procedure because the evolutionary state of the population varies with the progress of generations. Therefore, to include the solution to these issues, we propose a new DE variant named MIDE, which performs the search procedure based on the status of some past evolutionary generations and multiple top-fitted individuals of the population. The proposed MIDE algorithm also creates more diversity and convergence towards promising and elite individuals through a new mutation scheme and control parameter settings.

In this section, we mainly describe the MIDE algorithm, and the framework of the MIDE is divided into three subsections: the first two subsections present the proposed mutation operator and selection schemes for involved parent individuals, and the third subsection describes the control parameters used in the MIDE.

A. Multiple Elite Individuals and Achieve-Based Mutation Operator

For each individual of the DE, promising directions of search are needed to update their states. In this paper, we have defined a new mutation operator, DE/current-to-leader/1, based on the guidance of multiple elite individuals enrich with fitness quality and success rates. The proposed mutation operator is expressed as follows:

$$V_i^g = X_i^g + F_i^g \times (X_u^g - X_i^g) + F_i^g \times (X_v^g - \hat{X}_r^g) \quad (7)$$

where X_u^g is a weighted center of m high-fitted leading individuals of the population in terms of fitness, and X_v^g is randomly chosen as one of the top m individuals of the population in terms of success rates. F_i^g is a mutation factor or scale factor associated with individual X_i^g and re-generated at each generation using (11), which will be discussed later. Weighted center of m individuals and selection of any of m individuals in terms of success rate reduces the chance of biasing the search in a single elite direction, which can prevent the situation of getting trapped at local optimal solutions and premature convergence.

The second difference vector in mutation operator (7) is added to enhance the chance of exploration of solution space. Recently discarded mutant vectors may have a greater exploration ability, therefore an archive **A** is constructed to store discarded mutant vectors. From this archive **A**, an individual $\hat{X}_r^g$ is selected randomly at generation g for individual X_i^g. Archive **A** is initiated to be empty, and later with the evolutionary process, discarded mutant vectors are added to this archive.

B. Adaptive Strategy to Manage Archive and Leading Individuals

The selection of parent individuals in the mutation operator plays an important role to provide efficient search directions. To determine a parent individual X_u^g involved in mutation operator (7), we have used a weighted center-based approach. In this approach, we calculated an individual X_u^g as follows:

$$X_u^g = \begin{cases} \sum_{k=1}^{m} \frac{f(X_w^g) - f(X_k^g)}{\sum_{p=1}^{m} f(X_w^g) - f(X_p^g))} X_k & \text{if } f(X_b^g) < f(X_w^g) \\ X_b^g & \text{if } f(X_b^g) = f(X_w^g) \end{cases} \quad (8)$$

where X_b and X_w refer to the best and worst-fitted individuals, respectively. It can be seen that an individual X_u^g is a convex combination of m top-fitted individuals of the population and will have a lesser distance to the higher-ranked individual as compared to the lower rank individual in terms of fitness. Furthermore, to establish a good trade-off between exploitation and exploration, the number m is chosen as a time-varying parameter given by

$$m = round\Big[\frac{N_P}{2} - \Big(\frac{N_P}{2} - 1\Big) \times \frac{F_E}{F_{E,max}}\Big] \quad (9)$$

where $round[k]$ denotes the greatest value of integer less than or equal to k.

Our weighted center-based approach is similar to the approach introduced by Zhang et al. [48]. They have used this approach to define one of the parent vectors involved in the particle swarm optimization (PSO) search scheme. This approach differs from our approach in two senses. First, its weight coefficients are defined using an exponential function which is computationally more expensive than our approach where the weight coefficients are defined just using a ratio of fitness differences. Second, it reduced the number of leaders linearly from a fixed value of 20 to 10 over the generations. However, in our approach, we decrease the number of leaders based on the size of the population. We have gradually decreased the number of leaders from the size $N_P/2$ to 1 during the evolution procedure. Hence, our approach focuses comparatively more on exploration as the number of leaders is high at the beginning of the evolutionary process. Similarly, the MIDE contributes more to exploitation at the end of the evolutionary process as only a few top-fitted individuals are used as leaders.

The number m defined in (9) is also used to select the individual X_v^g, but it is selected based on the success history of individuals in the population. The success rate counter for each individual is calculated within a certain period of generations. When this period ends, the success counter for each individual is re-initialized to zero value. Mathematically, the success rate counter for individual X_i is updated as follows:

$$S_i = \begin{cases} S_i + 1 & \text{if } f(V_i^g) < f(X_i^g) \\ 0 & \text{if } f(V_i^g) \geq f(X_i^g) \text{ or } rem(g, g_L) = 0 \end{cases} \quad (10)$$

where S_i is a success counter associated with individual X_i, $rem(a, b)$ indicates the remainder value, when a is divided by b. g_L is a predefined number of generations to execute the piecewise search-based evolution in DE.

The parent individual X_v^g is selected as any of the individuals from m top leading individuals in terms of success rate calculated using (10).

At each generation g, where $rem(g, g_L) = 0$, all the individuals are deleted from archive **A** except the best one from them. In this way, the computational burden of storing individuals in an archive is reduced in the proposed MIDE.

C. Settings of Control Parameters

The parameter settings for scale factor F and crossover rate C_R are tuned in our MIDE through the extension of settings adopted by the JADE algorithm [20]. Since our optimization process in the MIDE is followed in a piecewise manner, we have generated comparatively more diversified parameter values. During the generation of parameter values, we also integrated the concept of OBL [49] when we found that the current chosen settings of parameters are unable to provide good solutions. Mathematically, our parameter control strategy is explained as follows:

The scale factor plays a crucial role in controlling the diversity of the DE population. A larger perturbation in scale factor values provides a chance of escaping from local optimal solutions. Hence, in our MIDE algorithm, we have generated scale factor F_i associated with individual X_i using Cauchy distribution with location parameter μ_F and scale parameter 0.1 or 0.5 as follows:

$$F_i = \begin{cases} Cauchy(\mu_F, 0.1) & \text{if } S_F \neq \emptyset \\ Cauchy(\mu_F, 0.5) & \text{if } S_F = \emptyset \end{cases} \quad (11)$$

where S_F is a set that stores all the successful values of F_i, when it produces a better trial vector as compared to the target vector. The Cauchy distribution is used to diversify the values of F more as compared to the normal distribution because the Cauchy distribution has a wider tail than the normal distribution. This behavior is beneficial when the global optima is far away from the current search region because the F values generated from the tail region provide sufficient perturbation to avoid stagnation issues at the local optima. We have used a scale parameter as 0.5 to generate more diversified F values when the pool of recent successful scale factors is empty. When the value of F_i generated from (11) exceeds the value 1, it is truncated to value 1, and if $F_i \leq 0$, it is regenerated until $F_i \leq 1$. The location parameter μ_F is initially set to 0.5, and later it is updated as follows:

$$\mu_F = \begin{cases} (1-c)\cdot\mu_F + c\cdot mean_L(S_F) & \text{if } S_F \neq \emptyset \text{ and } rem(g, g_L) \neq 0 \\ (1-\mu_F) & \text{if } S_F = \emptyset \text{ and } rem(g, g_L) \neq 0 \\ \mu_{F,old} & \text{if } rem(g, g_L) = 0 \end{cases} \tag{12}$$

where $\mu_{F,old}$ is the value of μ_F calculated in the last generation. The second expression of the μ_F is calculated using the OBL mechanism [49] using lower and upper bounds as 0 and 1, respectively. The purpose of introducing OBL is to generate an opposite approximate of the μ_F so that the more suitable setting can be explored quickly. $mean_L(\cdot)$ is the Lehmer mean calculated as

$$mean_L(S_F) = \frac{\sum_{F\in S_F} F^2}{\sum_{F\in S_F} F} \tag{13}$$

The Lehmer mean places more weight on larger successful F values as compared to the usual arithmetic mean [20]. Hence this mean is helpful to propagate larger mutation factors, which improves the progress rate.

To generate the crossover rate C_{R_i} associated with individual X_i, we have used a normal distribution with mean μ_{C_R} and standard deviation 0.1 or 0.5 as follows:

$$C_{R_i} = \begin{cases} Normal(\mu_{C_R}, 0.1) & \text{if } S_{C_R} \neq \emptyset \\ Normal(\mu_{C_R}, 0.5) & \text{if } S_{C_R} = \emptyset \end{cases} \tag{14}$$

where S_{C_R} is a set that stores all the successful values of C_{R_i}, when it produces a better trial vector as compared to the target vector. If the generated C_{R_i} exceeds the limit 0 or 1, then it is truncated to its nearest value 0 or 1. We have used normal distribution instead of Cauchy distribution because of its short tail property. If we use the Cauchy distribution, then it will generate higher values of C_R which will be truncated at 1. In this case, the generated C_R values will be independent of the Cauchy distribution. When we use Normal distribution to generate C_R values, then due to the short tail property, most of these values will be generated within the interval (0,1) and will follow the desired distribution.

Equation (14) indicates that the C_R values are more likely to be selected closer to the μ_{C_R}, when the set S_{C_R} is non-empty. When the set S_{C_R} is empty, then the C_R values are generated using the normal distribution with the same mean μ_{C_R}, but with standard deviation 0.5. In this case, comparatively more diversified C_R values are generated to explore its more promising setting. μ_{C_R} is initially assigned as the value 0.5, later, it updates its value as follows:

$$\mu_{C_R} = \begin{cases} (1-c)\cdot\mu_{C_R} + c\cdot mean_A(S_{C_R}) & \text{if } S_{C_R} \neq \emptyset \text{ and } rem(g, g_L) \neq 0 \\ (1-\mu_{C_R}) & \text{if } S_{C_R} = \emptyset \text{ and } rem(g, g_L) \neq 0 \\ \mu_{C_R,old} & \text{if } rem(g, g_L) = 0 \end{cases} \tag{15}$$

where $mean_A(\cdot)$ is the usual arithmetic mean, $\mu_{C_R,old}$ is the value of μ_{C_R} calculated in the last generation. The second expression of the μ_{C_R} is calculated using the OBL mechanism [49], using lower and upper bounds as 0 and 1, respectively. The OBL scheme provides an opposite approximate of the current settings of the C_R, which can have better chances of setting new promising C_R values. The third expression of the μ_{C_R} fixes the old setting of μ_{C_R} for the next piecewise search of the evolution procedure.

The population size N_P also plays a crucial role in the performance of the DE. In our MIDE, we have chosen a time-varying setting to update the N_P, which is expressed as follows:

$$N_P^g = round\Big[N_{P_{max}} - (N_{P_{max}} - N_{P_{min}}) \times \frac{F_E}{F_{E,max}}\Big] \tag{16}$$

where $N_{P_{max}}$ and $N_{P_{min}}$ are upper and lower bounds for the population size. Since a minimum of four individuals are needed to define our mutation operator (7) effectively, we have set the $N_{P_{min}}$ to 4 in our MIDE. The $N_{P_{max}}$ value we have fixed to 100 in this paper. This reduction in population size discards the unfavorable individuals over the generations and maintains the balance between diversity and convergence of the population. At the initial phases of the evolutionary procedure of the MIDE, a large population size helps to avoid the chances of stagnation at local optimal solutions and provides sufficient exploration. Later, it gradually decreases to focus on exploiting the discovered search areas to avoid the chances of skipping true solutions.

A summary of the diverse impacts of embedded strategies in the MIDE is explained as follows:

- A new mutation scheme is proposed to utilize the information of multiple elite individuals and evolution feedback from previous generations. This operator provides efficient search direction towards promising explored states of individuals.
- Adaptive settings for selection of leading parent individuals of the mutation operator (7) and management of an achieve **A**, establish the balanced evolutionary search between diversity and convergence rate to avoid local optima stagnation and premature convergence issues.
- Parameters settings for F and C_R are tuned based on a piecewise search and population size is chosen as a time-varying setting to learn them based on diverse evolutionary stages and to avoid the unfavorable computational burden of evolving inferior individuals, respectively.

As the pseudo-code shown in Algorithm 1, in the proposed MIDE, a set of individuals called the population is first generated randomly within the solution space, and the settings of algorithm parameters are set to initiate the optimization process. Then, the iterative procedure of the MIDE is performed based on the new mutation scheme, and parameter selection approach. Within the iterative procedure of the MIDE, parameters μ_F, μ_{C_R}, F and C_R are determined as discussed in Section III-C and the proposed mutation rule is implemented to generate mutant vectors corresponding to each individual of the population as explained in Section III-A. Finally, the usual crossover and selection operators are applied to update the population of the MIDE. During the implementation of these operators, archive **A** is updated based on historically discarded individuals, success counter vector S is updated based on the success of trial vectors, and the pools S_{C_R} and S_F are updated based on the successful values of control parameters as explained in Sections III-A, B, and C.

IV. Experimental Results

A. Benchmark Functions and Experimental Setup

In this section, 30 problems of CEC 2014 [15], and 30 problems of CEC 2017 [16] test suites are used to validate the performance of the MIDE. These problems contain four categories of difficulties, including unimodality, multimodality, hybrid, and composite. A detailed explanation of these problems is provided in their original papers. In CEC 2014, problems F1 to F3 are unimodal, F4 to F16 are multimodal, F17 to F22 are hybrid, and F23 to F30 are composite. On the other hand, in CEC 2017, F1 to F3 are unimodal, F4 to F10 are multimodal, F11 to F20 are hybrid, and F21 to F30 are composite problems.

We will compare the MIDE with 12 other state-of-the-art algorithms including classical DE or ODE [7], DE variants namely, EPSDE [32], SaDE [41], SinDE [50], jDE [44], JADE [20], NBOLDE [51], TVDE [47], MPEDE [52], RSDE [53], and other algorithms namely MLGSA [54], DLABC [55]. To perform fair and reliable comparisons, settings of parameters are fixed as per suggestions given in their original papers. The maximum function evaluation $F_{E,max}$ is set to $10^4 \times d$ in all the algorithms [15, 16]. All the algorithms are coded and implemented by MATLAB R2020b on a personal computer with the Intel 2.30GHz CPU, 16 GB memory, and Windows 10 operating system.

Due to the stochastic nature of metaheuristic algorithms, 30 independent trials of each compared algorithm will be conducted. To statistically analyze the results, Wilcoxon rank-sum test [56] will be used at a 5% significance level. The outcomes of the statistical test will be indicated by the symbols "+/-/≈" to demonstrate the better, worse or identical performance of the MIDE and compared algorithms, respectively.

B. Effectiveness of Strategies Applied in the MIDE

This section analyzes the impact of each strategy which is used to design the MIDE. Instead of showing the objective function value for each problem, we have calculated the Friedman ranking corresponding to different strategies of the MIDE. These rankings are shown in Table I. "Strategy 1" refers to the mutation scheme with the random individual in the place of the first individual of the first difference vector and random individuals in the second difference vector. "Strategy 2" refers to the mutation scheme with a weighted center as a parent individual in the first difference vector, while the parent individuals in the second difference vector are randomly chosen distinct individuals from the population. We have chosen the same parameter settings for F, C_R, and N_P as used in the JADE for strategies 1 and 2. "Strategy 3" refers to the mutation scheme as given in (7), but with the parameters setting used by the JADE. "Strategy 4" corresponds

Algorithm 1 Pseudo-code of MIDE

Set up the maximum function evaluations $F_{E,max}$, maximum generations g_{max}, function evaluation counter $F_E = 0$, generation counter $g = 0$, and $S_{C_R} = \emptyset$
Randomly initialize the population P^g of N_P individuals within the solution space $[lb, ub]$ as $P^g = \{X_1^g, X_2^g, \ldots, X_{N_P}^g\}$ with $X_i^g = (x_{i,1}^g, x_{i,2}^g, \ldots, x_{i,d}^g)$, where $i = 1, 2, \ldots, N_P$
Initialize the parameters $\mu_{C_R} = \mu_F = 0.5$, $c = 0.1$, and archive **A** = $\emptyset$
Evolutionary procedure of the MIDE:
while $F_E < F_{E,max}$ and $g < g_{max}$ **do**
  if $(rem(g, g_L) = 0)$ **then**
    $S_F = S_{C_R} = \emptyset$
    Delete all the individuals from archive **A** except the best one from them
    $\mu_F = \mu_{F,old}$; $\mu_{C_R} = \mu_{C_R,old}$
  end if
  if $(S_F = S_{C_R} = \emptyset$ and $rem(g, g_L) \neq 0)$ **then**
    $\mu_F = (1 - \mu_F)$; $\mu_{C_R} = (1 - \mu_{C_R})$
  end if
  if $(S_F \neq \emptyset$ and $S_{C_R} \neq \emptyset$ and $rem(g, g_L) \neq 0)$ **then**
    $\mu_F = (1 - c) \cdot \mu_F + c \cdot mean_L(S_F)$
    $\mu_{C_R} = (1 - c) \cdot \mu_{C_R} + c \cdot mean_A(S_{C_R})$
  end if
  Sort the population in decreasing order of success rate of individuals
  for $(i = 1$ to $N_P)$ **do**
    Generate scale factor F_i as

$$F_i = \begin{cases} Cauchy(\mu_F, 0.1) & \text{if } S_F \neq \emptyset \\ Cauchy(\mu_F, 0.5) & \text{if } S_F = \emptyset \end{cases}$$

    Determine the individual X_u^g as a weighted center of m high-fitted individuals using eq. (8)
    Randomly chose individual X_v^g from m top individuals in terms of success rate
    Randomly chose individual $\hat{X}_r^g$ from archive **A**
    Generate a mutant vector V_i^g as

$$V_i^g = X_i^g + F_i^g \times (X_u^g - X_i^g) + F_i^g \times (X_v^g - \hat{X}_r^g)$$

    Generate crossover rate C_{R_i} as

$$C_{R_i} = \begin{cases} Normal(\mu_{C_R}, 0.1) & \text{if } S_{C_R} \neq \emptyset \\ Normal(\mu_{C_R}, 0.5) & \text{if } S_{C_R} = \emptyset \end{cases}$$

    Generate index j_{rand} from a set $\{1, 2, \ldots, d\}$
    for $(j = 1$ to $d)$ **do**
      if $(rand_{i,j}^g < C_{R_i}$ or $j = j_{rand})$ **then**
        $u_{i,j}^g = v_{i,j}^g$
      else
        $u_{i,j}^g = x_{i,j}^g$
      end if
    end for
    if $(f(U_i^g < f(X_i^g))$ **then**
      $X_i^{g+1} = U_i^g$
      $S_i = S_i + 1$; $F_i \to S_F, C_{R_i} \to S_{C_R}$
    else
      $X_i^{g+1} = X_i^g$; $S_i = 0$; $U_i^g \to$ **A**
    end if
  end for
  Update the size of the population using eq. (16) by discarding the low-fitted individuals
  Increase the generation and function evaluation count as $F_E = F_E + N_P$, $g = g + 1$
end while
return Best individual X_b

to the mutation scheme as given in (7) and uses a proposed parameter setting for F and C_R. Finally, "Strategy 5" refers to the combination of Strategy 4 and linearly decreasing parameter setting for N_P. Hence, strategy 5 is nothing but the proposed MIDE algorithm. We can see from the ranking of different strategies how they hierarchically improve the search performance of the MIDE. Hence, it can be concluded that all the strategies have their impact in improving the levels of exploration and exploitation.

TABLE I: Friedman ranking of each applied strategies of the MIDE

Instances	Strategy 1	Strategy 2	Strategy 3	Strategy 4	Strategy 5/MIDE
30d (CEC 2014)	3.63	3.60	3.00	2.90	1.87
30d (CEC 2017)	4.28	3.78	2.75	2.70	1.48
50d (CEC 2014)	4.00	3.37	3.03	2.83	1.77
50d (CEC 2017)	4.33	3.83	3.20	2.33	1.30
Overall	4.06	3.65	3.00	2.69	1.60

C. Comparison on 30d Problems

We first compare the MIDE with 12 state-of-the-art algorithms on $30d$ CEC 2014 and CEC 2017 problems. Optimization results in terms of average error (Avg error) and standard deviation (Std) of errors are provided in Tables S1 and S2 of Supplementary file. This error is defined as $(f(X) - f(X^*))$, where $f(X)$ is the fitness value at the best solution obtained so far and $f(X^*)$ is the fitness value at optimal solution X^*. From these tables, we can observe the following outcomes.

On most of the unimodal problems (F1-F3) of CEC 2014 and CEC 2017, the MIDE provides near-optimal solutions and shows competitive performance with other algorithms.

On 5 out of 13 multimodal problems F4-F16 of CEC 2014, the MIDE keeps its superiority and outperforms all other algorithms. On 2 problems, it obtains the second rank, while on the remaining problems, the results are competitive with most of the algorithms. For the second set of benchmark CEC 2017, MIDE outperforms other algorithms on 4 out of 7 problems. On problems F6 and F9, MIDE obtains near-optimal solutions. Only in one problem F4, the solution quality of the MIDE is worse than the other five algorithms and better than seven algorithms.

On hybrid and composite problems, our MIDE performs very competitively with other compared algorithms. It does not obtain rank one on all the problems, but on average, its performance is very promising on these problems.

Overall, from ranking and statistical test reported in the last three rows of tables, MIDE outperforms all other algorithms. Hence, it can be concluded that for solving diverse categories of optimization problems, our MIDE can be picked over other compared algorithms.

D. Comparison on 50d Problems

In this part, the performance of the MIDE is evaluated on $50d$ of CEC 2014 and CEC 2017. The results are shown in Tables S3 and S4 of Supplementary file, which indicate the competitive performance of the MIDE with other DE variants and other algorithms on unimodal problems.

For multimodal problems of CEC 2014, MIDE provides the best fitness value in 4 out of 13 problems. In the remaining problems, the MIDE competes with most of the DE variants and other algorithms. On the other hand, for CEC 2017 problems, it shows the best fitness in 3 out of 7 problems.

The performance comparison on hybrid and composite problems indicates the competitive search ability of the MIDE with other compared algorithms.

Overall evaluation of performance verifies the superior solution accuracy of the MIDE. Statistical comparison, reported in the last 3 rows of tables, verified this fact.

To analyze the winning performance among all the compared algorithms, their ranking is calculated using the Friedman test [57], which is shown in Table II. It is verified from this ranking that in all the comparisons, MIDE outperforms all other algorithms, and overall, it obtains the first rank.

To further analyze the search behavior of the MIDE and compare it with other algorithms, we present their convergence curves in Fig. 1 for the $50d$ problems of CEC 2014 and CEC 2017. The convergence curves are shown for problems F7, F11, F15, and F16 of CEC 2014, and F8, F10, F11, and F16 of CEC 2017. It can be seen from these figures that in most of the curves, the MIDE shows a faster convergence rate as compared to other algorithms. It is also evident from these curves that our proposed strategies of the MIDE do not lead the population to converge too quickly and maintain the evolution procedure of finding new best solutions throughout the whole optimization process. We have also added the box plots in Fig. 2 for some of the test problems to analyze the results obtained in each trial of algorithms. These box plots show the better performance of the MIDE than other compared algorithms on almost every trial of its execution, which verifies its reliability to produce promising optimization results.

TABLE II: Average rank of algorithms in Friedman test

Algorithm	CEC 2014		CEC 2017		Avg	Final
	30D	50D	30D	50D	rank	rank
ODE	8.28	9.07	9.47	9.52	9.08	11
EPSDE	7.03	6.75	6.67	7.15	6.90	6
SaDE	7.45	7.68	6.70	6.87	7.18	7
SinDE	4.78	5.03	4.83	4.75	4.85	5
jDE	4.90	4.97	4.48	4.73	4.77	4
JADE	4.65	5.00	4.58	4.18	4.60	3
NBOLDE	12.50	12.13	12.43	12.43	12.38	13
TVDE	7.35	7.48	7.02	7.13	7.25	8
MPEDE	4.62	5.05	3.78	4.27	4.43	2
MLGSA	6.63	6.38	8.47	8.33	7.45	9
RSDE	8.43	7.37	8.67	8.17	8.16	10
DLABC	9.88	9.22	10.27	9.77	9.78	12
MIDE	4.48	4.87	3.63	3.70	4.17	1

E. Analysis of Population Diversity

The main feature of the MIDE is to provide sufficient diversity to reduce the shortcomings of stagnation at local optima and premature convergence. To analyze this feature, we have evaluated the population diversity (P_D) [58], which is calculated as follows:

$$P_D = \frac{1}{N_P \times \mathcal{D}} \sum_{i=1}^{N_P} \sqrt{\sum_{j=1}^{d} (x_{i,j} - \bar{x}_j)^2} \tag{17}$$

where P_D indicates the diversity of the algorithm population. $\mathcal{D}$ is the maximum of the distances calculated between each pair of individuals and can be computed as follows:

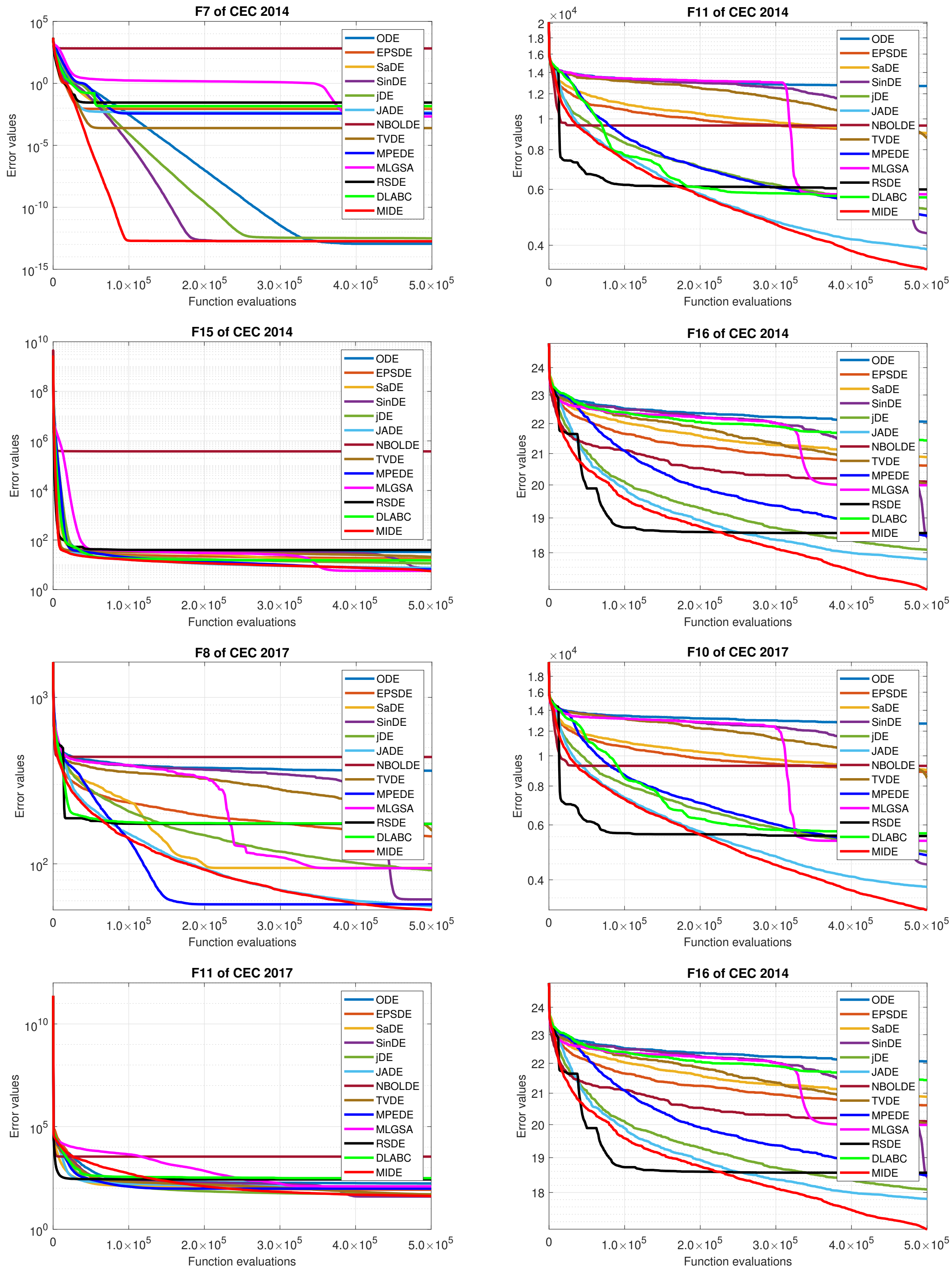

Fig. 1: Comparisons of convergence performance for 50 dimensional CEC 2014 and CEC 2017 benchmarks

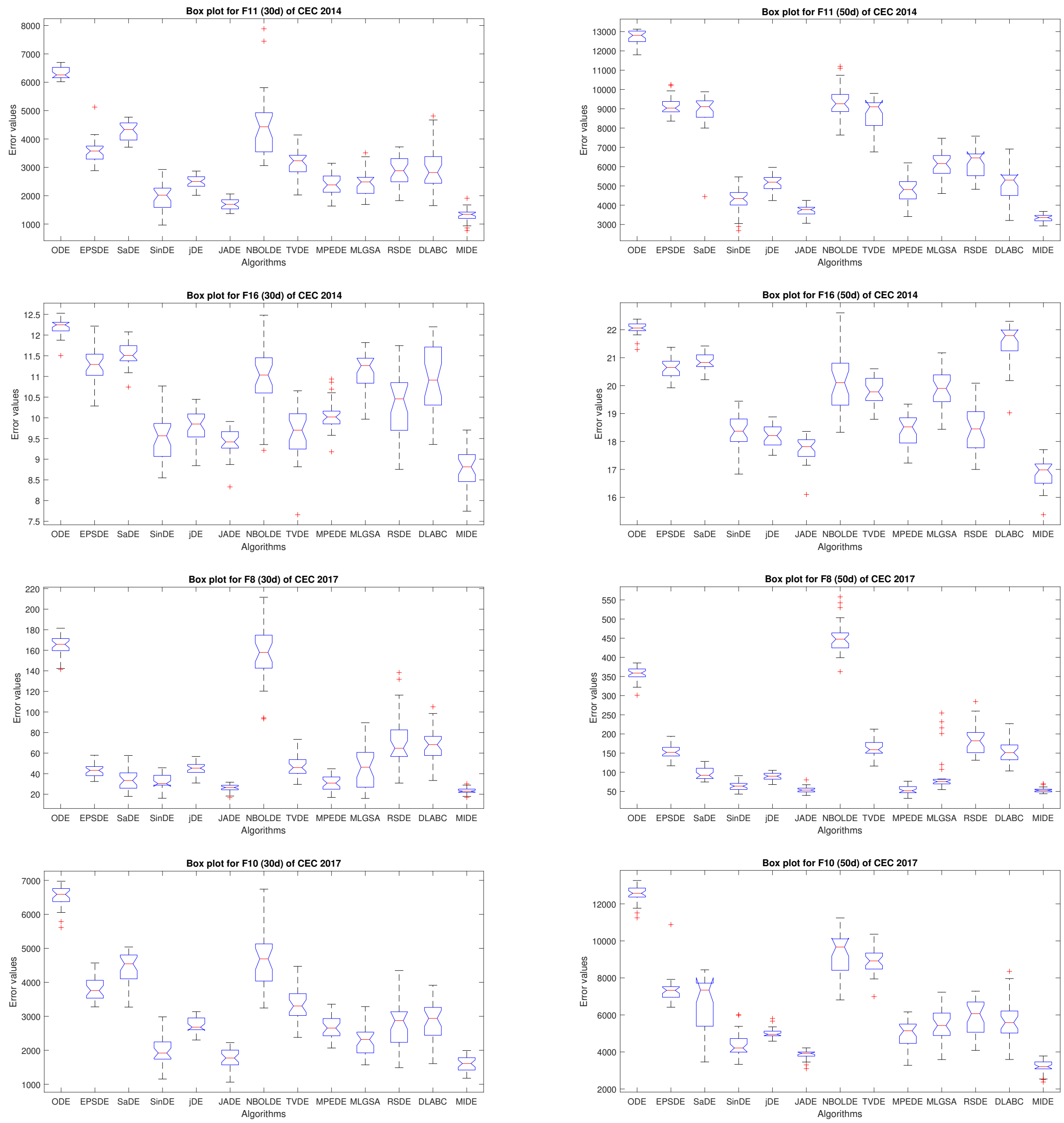

Fig. 2: Visualization of results using box plots

$$\mathcal{D} = \max_{(i \neq k) \in [1, N_P]} \left(\sqrt{\sum_{j=1}^{d} (X_{i,j} - X_{k,j})^2} \right) \tag{18}$$

$\bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_d)$ is the average position of the population given by

$$\bar{X} = \frac{1}{N_P} \sum_{i=1}^{N_P} X_i \tag{19}$$

In metaheuristic algorithms, proper diversity can improve their search performance. High-level diversity promotes the exploration of solution space, while low-level diversity focuses on exploitation. We have taken one problem from each category of benchmark sets to analyze the population diversity. These problems are F1 (unimodal), F10 (multimodal), F18 (hybrid), and F27 (composite) from both benchmark sets CEC 2014 and CEC 2017. The diversity curves for the MIDE on these problems with dimension 30 and 50 are plotted in Fig. 3. It can be observed that the MIDE maintains the diversity of the population and prevents the population to be trapped at local optima during the optimization process. Hence, the proposed mutation operator and parameter selection schemes are effective to enhance the diversity skills of the algorithm and improving the ability to jump out from local optima when individuals are stuck at local optimal solutions and show stagnation behavior.

Furthermore, to systematically assess the balancing ability between exploitation and exploration in the MIDE, we have adopted a dimension-wise diversity-based method [59]. In

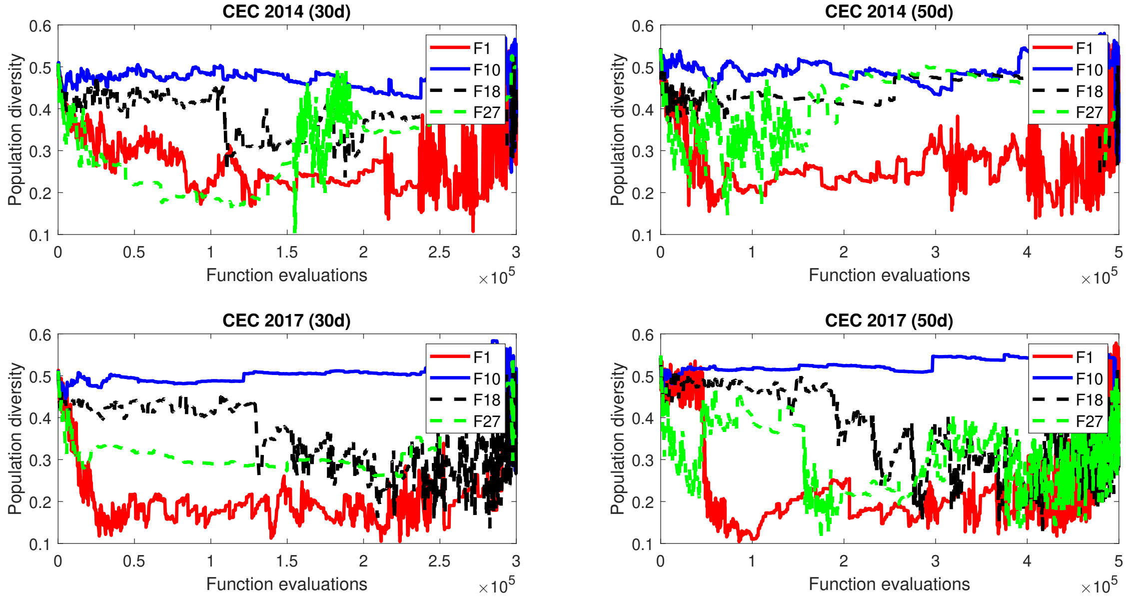

Fig. 3: The Diversity curves of the MIDE population for problems F1, F10, F18 and F27 with dimensions 30 and 50.

metaheuristic algorithms, the balance can be characterized by the percentage of exploitation and exploration, which is given by:

$$EXPT\% = \left(\frac{|Div - Div_{max}|}{Div_{max}}\right) \times 100 \quad (20)$$

$$EXPL\% = \left(\frac{Div}{Div_{max}}\right) \times 100 \quad (21)$$

where $EXPT\%$ represents the percentage of exploitation-level and $EXPL\%$ refers to the percentage of exploration-level in the algorithm. Div_{max} represents the maximum diversity found in the entire optimization process. In the dimension-wise diversity-based method [59], the population diversity Div is calculated as follows:

$$Div = \frac{1}{d}\sum_{j=1}^{d} Div_j \quad (22)$$

$$Div_j = \frac{1}{N_P}\sum_{i=1}^{N_P} |median_j(X) - x_{i,j}| \quad (23)$$

where $median_j(X)$ represents the median of dimension j in the whole population.

Using the above expressions, we have shown the exploitation and exploration effects of the MIDE in Fig. 4. We can observe from the figure that the MIDE focuses more on exploration at the early stages of evolution, and sustains the exploration levels until the end of the optimization process. However, an appropriate transition is performed by the MIDE from exploration to exploitation. This characteristic is an impact of the proposed mutation operator guided by multiple elite individuals and piecewise search-based evolutionary process, which promotes diversity in the algorithm. We can also observe from this figure that the powerful exploitation ability is present in the MIDE, which helps in boosting its convergence speed. This realization of balancing ability between exploitation and exploration proves the promising search performance of the MIDE in terms of fast convergence and avoiding local optima during the optimization process.

F. Computational Complexity

Compared with the classical DE, additional computations in the MIDE are due to selection schemes for parent individuals involved in the proposed mutation scheme and the newly introduced parameter setting. The overall complexity of the MIDE can be calculated using the pseudo-code provided in Algorithm 1. First, in the initialization (see steps 1-3), the time complexity of MIDE is $O(N_P)$. Then, steps 5-16 are executed, whose total complexity is $O(n_L \cdot N_A \cdot log(N_A))$. Here n_L indicates the total number when the condition given in line 5 is true, and N_A is the size of the archive. When this condition (line 5) is true the archive is needed to be sorted based on the fitness values of individuals and it needs the complexity $O(n_L \cdot N_A \cdot log(N_A))$. Step 17 sorts the population in terms of success rate and it has the complexity of $O(N_P \cdot log(N_P))$. In steps 21 and 22, random selection takes $O(1)$ and in total, steps 18 to 39 have the complexity $O(N_P + N_P \cdot d)$ including $O(N_P)$ complexity for mutation and $O(N_P \cdot d)$ for crossover. The assignment corresponding to different parameters takes $O(1)$. The population size reduction (at step 40) needs the sorting process and deletion of low-fitted individuals, and it has the complexity $O(N_P \cdot log(N_P))$. By summing up all, an order of the complexity for the MIDE in the worst case is $O(g_{max} \cdot (N_P \cdot d + N_P \cdot log(N_P)))$, where g_{max} indicates the maximum generations set for the MIDE. Besides MIDE, the computational complexities of the JADE, NBOLDE, and MPEDE is $O(g_{max} \cdot (N_P \cdot d + N_P \cdot log(N_P)))$, which is the same

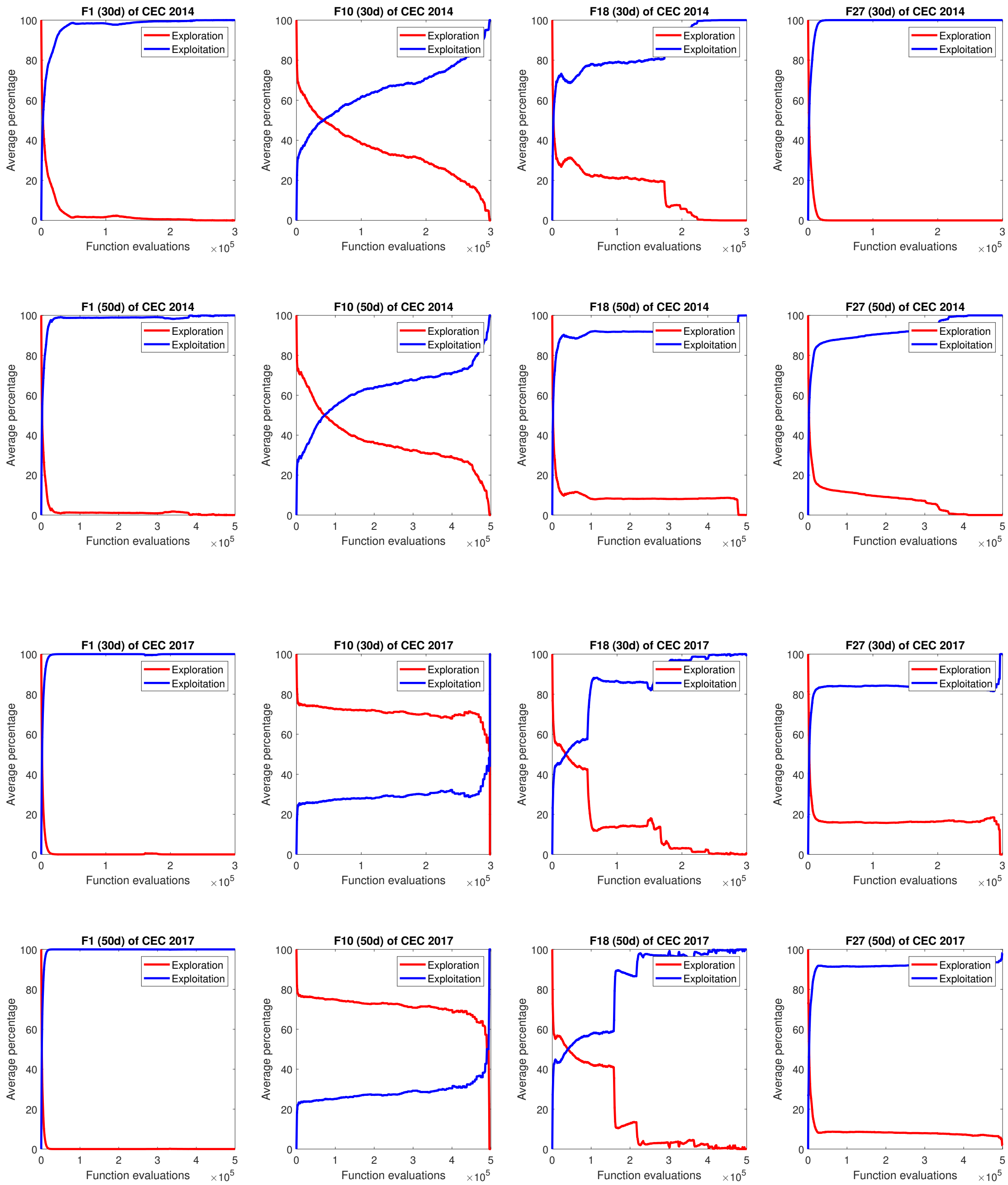

Fig. 4: Average balance employed by the MIDE on F1, F10, F18 and F27 of CEC 2014 and CEC 2017 with dimensions 30 and 50.

TABLE III: Computational complexity of algorithms

	CEC 2014							CEC 2017					
	30d				50d			30d			50d		
Algorithm	$T0$	$T1$	$\hat{T}2$	$(\hat{T}2-T1)/T0$	$T1$	$\hat{T}2$	$(\hat{T}2-T1)/T0$	$T1$	$\hat{T}2$	$(\hat{T}2-T1)/T0$	$T1$	$\hat{T}2$	$(\hat{T}2-T1)/T0$
ODE	0.072	0.006	2.074	28.805	0.009	2.514	34.896	0.006	2.357	32.748	0.015	3.184	44.138
EPSDE	0.072	0.006	5.100	70.963	0.009	5.539	77.029	0.006	5.190	72.215	0.015	5.840	81.130
SaDE	0.072	0.006	11.297	157.283	0.009	12.630	175.802	0.006	12.165	169.369	0.015	12.054	167.701
SinDE	0.072	0.006	1.843	25.595	0.009	2.570	35.666	0.006	2.455	34.117	0.015	3.070	42.553
jDE	0.072	0.006	1.241	17.203	0.009	1.912	26.511	0.006	1.836	25.498	0.015	2.508	34.718
JADE	0.072	0.006	1.434	19.888	0.009	2.020	28.005	0.006	1.923	26.699	0.015	2.703	37.434
NBOLDE	0.072	0.006	1.301	18.042	0.009	1.771	24.540	0.006	1.779	24.697	0.015	2.528	35.001
TVDE	0.072	0.006	1.943	26.981	0.009	2.660	36.927	0.006	2.482	34.492	0.015	3.328	46.138
MPEDE	0.072	0.006	1.282	17.770	0.009	1.876	26.009	0.006	1.804	25.039	0.015	2.816	39.004
MLGSA	0.072	0.006	14.865	206.986	0.009	18.643	259.567	0.006	16.375	228.017	0.015	19.930	277.413
RSDE	0.072	0.006	2.607	36.233	0.009	3.082	42.806	0.006	3.150	43.795	0.015	3.969	55.068
DLABC	0.072	0.006	2.231	30.999	0.009	3.222	44.759	0.006	2.753	38.268	0.015	4.006	55.582
MIDE	0.072	0.006	3.609	50.198	0.009	5.112	71.079	0.006	4.191	58.295	0.015	5.435	75.499

as the complexity of the MIDE. Moreover, the complexity of the ODE, EPSDE, SaDE, SinDE, jDE, TVDE, RSDE is $O(g_{max} \cdot N_P \cdot d)$, of the MLGSA is $O(g_{max} \times N_P^2)$, and of the DLABC is $O(g_{max} \times N_P)$.

Furthermore, we have also analyzed the computational complexity of all the compared algorithms using the criteria provided by the CEC 2014 and CEC 2017. Table III shows the computational complexity of compared algorithms. In this table, $T0$ is the computational time of a test program given in [15, 16]. $T1$ is the computational time of function F18 with 200000 function evaluations. $\hat{T}2$ is the average computational time recorded by the algorithm over 5 independent runs to solve the problem F18 with 200000 function evaluations. The final results are shown in Table III by $T0$, $T1$, $\hat{T}2$, and $(\hat{T}2 - T1)/T0$.

According to Table III, the MIDE has greater $\hat{T}2$ and $(\hat{T}2 - T1)/T0$ than ODE, SinDE, jDE, JADE, NBOLDE, TVDE, MPEDE, RSDE, and DLABC. The algorithms EPSDE, SaDE, and MLGSA have greater $\hat{T}2$ and $(\hat{T}2 - T1)/T0$ than the MIDE for both the benchmark sets with 30d and 50d. Hence, computational efficiency is one aspect on which we can work more in the future to make the MIDE comparatively more computationally efficient. The results of Table III also illustrate that jDE is the fastest and MLGSA is the slowest in producing the optimization results.

G. Comparison on Constrained Engineering Problems

In this section, the proposed MIDE and other algorithms such as ODE, EPSDE, SaDE, SinDE, jDE, JADE, NBOLDE, TVDE, MPEDE, MLGSA, RSDE, and DLABC are used to solve 4 engineering problems, namely speed reducer design (EP1), welded beam design (EP2), tension/compression spring design (EP3), and pressure vessel design (EP4). The description of these problems can be found in [10, 60]. The optimization results are shown in Table IV with the best results in boldface. The MIDE algorithm obtains the best solution in EP1 and EP2, while in EP3 EPSDE and in EP4, MPEDE algorithm has provided the best solutions. The overall ranking shows that the MIDE is superior compared to other algorithms to solve these engineering problems. This performance behavior indicates that the MIDE establishes a reasonable balance between exploitation and exploration during its evolution process.

V. Conclusion

In this article, we have proposed a multiple elite individual guided piecewise search-based differential evolution algorithm

TABLE IV: Comparison of results on constrained engineering problems

Algorithm	Result	EP1	EP2	EP3	EP4	Avg rank	Overall rank
ODE	**Avg**	2.99647E+03	1.76673E+00	1.28164E-02	6.06775E+03	8.25	8
	Std	3.32931E-02	1.55792E-02	8.85808E-05	8.92388E+00		
EPSDE	**Avg**	2.99641E+03	1.72565E+00	**1.26724E-02**	6.05971E+03	3.75	2
	Std	1.85310E-02	1.08135E-03	**9.40827E-06**	7.49349E-04		
SaDE	**Avg**	2.99640E+03	1.72497E+00	1.26889E-02	6.05971E+03	3.75	2
	Std	1.73896E-02	**9.19808E-05**	1.95799E-05	**2.66489E-04**		
SinDE	**Avg**	**2.99635E+03**	1.74610E+00	1.27580E-02	6.06228E+03	6	5
	Std	2.83512E-07	6.96052E-02	1.58034E-04	8.89462E+00		
jDE	**Avg**	2.99666E+03	1.73175E+00	1.27058E-02	6.06005E+03	6.75	6
	Std	5.23516E-02	3.33565E-03	3.09752E-05	2.31999E-01		
JADE	**Avg**	2.99637E+03	1.72817E+00	1.26845E-02	6.05993E+03	4.5	3
	Std	5.60281E-03	1.46457E-03	1.56561E-05	3.21180E-01		
NBOLDE	**Avg**	3.01550E+03	1.86752E+00	1.38312E-02	6.72678E+03	12	10
	Std	2.28908E+01	2.12650E-01	1.05674E-03	4.44989E+02		
TVDE	**Avg**	**2.99635E+03**	1.81117E+00	1.31416E-02	6.16331E+03	8.75	9
	Std	7.16567E-05	1.19137E-01	7.53703E-04	1.97830E+02		
MPEDE	**Avg**	3.00254E+03	1.79955E+00	1.29989E-02	**6.03525E+03**	7.5	7
	Std	1.26573E+00	2.40903E-02	2.16144E-04	4.37082E+01		
MLGSA	**Avg**	3.16942E+03	2.08128E+00	1.75232E-02	6.34955E+03	12.75	11
	Std	4.25442E+01	1.32591E-01	2.48796E-03	2.54874E+02		
RSDE	**Avg**	**2.99635E+03**	1.72876E+00	1.26833E-02	6.10837E+03	5	4
	Std	6.34468E-05	2.19337E-03	1.19232E-05	3.46051E+02		
DLABC	**Avg**	**2.99635E+03**	1.87659E+00	1.36663E-02	6.18536E+03	8.75	9
	Std	**1.38756E-12**	2.17045E-01	6.88520E-04	4.68159E+02		
MIDE	**Avg**	**2.99635E+03**	**1.72491E+00**	1.27465E-02	6.05972E+03	**3.25**	**1**
	Std	1.65313E-07	1.17442E-04	1.27426E-04	3.97034E-03		

(MIDE), in which a new mutation operator and control parameter settings are adopted. The mutation operator is designed to embed promising directions of search and to fully utilize the stages of evolution. The control parameters are tuned based on the past stages of evolution. However, these parameters have been tuned in a piecewise manner, i.e., historical information is utilized within a predefined evolution period, and thereafter, they are generated in a comparatively more diversified manner to maintain the diversity. Furthermore, the population size of the MIDE is linearly decreased by discarding low-fitted individuals from the population. To evaluate the performance of the MIDE, extensive experiments have been carried out on two different benchmark testbeds: CEC 2014 and CEC 2017, with two-dimensionality cases, as well as four constrained engineering problems. The comparison between the MIDE with other algorithms manifests that the MIDE offers comparatively more favorable performance on benchmark testbeds and engineering design problems.

Although MIDE demonstrates a competitive performance verified by experiments, some problems like the effectiveness of the mutation operator and control parameters need to be further analyzed in our future work. We will also work towards the computational efficiency of the MIDE using parallel processing. Moreover, we will test the ability of the MIDE for solving large-scale problems [61] and problems having noisy environments [62]. Based on this performance, in the future, we will extend the MIDE for these types of problems.

References

[1] X. Yu and M. Gen, *Introduction to evolutionary algorithms*. Springer Science & Business Media, 2010.

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

[3] J. Bi, H. Yuan, J. Zhai, M. Zhou, and H. V. Poor, "Self-adaptive bat algorithm with genetic operations," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 7, pp. 1284–1294, 2022.

[4] W. Gu, Y. Yu, and W. Hu, "Artificial bee colony algorithmbased parameter estimation of fractional-order chaotic system with time delay," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 107–113, 2017.

[5] J. Zhao, S. Liu, M. Zhou, X. Guo, and L. Qi, "Modified cuckoo search algorithm to solve economic power dispatch optimization problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 4, pp. 794–806, 2018.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[7] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[8] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[9] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

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

[11] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[12] M. Mahalakshmi, P. Kalaivani, and E. K. Nesamalar, "A review on genetic algorithm and its applications," *International Journal of Computing Algorithm*, vol. 2, no. 2, pp. 415–423, 2013.

[13] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE transactions on evolutionary computation*, vol. 13, no. 3, pp. 526–553, 2009.

[14] E. V. Altay and B. Alatas, "Differential evolution and sine cosine algorithm based novel hybrid multi-objective approaches for numerical association rule mining," *Information Sciences*, vol. 554, pp. 198–221, 2021.

[15] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization," *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, vol. 635, p. 490, 2013.

[16] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization," *Nanyang Technological University, Singapore and Jordan University of Science and Technology, Jordan and Zhengzhou University, Zhengzhou China, Tech. Rep., Oct 2016*, 2016.

[17] Y. Yu, S. Gao, Y. Wang, and Y. Todo, "Global optimum-based search differential evolution," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 2, pp. 379–394, 2019.

[18] 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.

[19] H.-Y. Fan and J. Lampinen, "A trigonometric mutation operation to differential evolution," *Journal of global optimization*, vol. 27, no. 1, pp. 105–129, 2003.

[20] 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.

[21] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 2066–2081, 2013.

[22] J. Wang, J. Liao, Y. Zhou, and Y. Cai, "Differential evolution enhanced with multiobjective sorting-based mutation operators," *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2792–2805, 2014.

[23] J. Cheng, Z. Pan, H. Liang, Z. Gao, and J. Gao, "Differential evolution algorithm with fitness and diversity ranking-based mutation operator," *Swarm and Evolutionary Computation*, vol. 61, p. 100816, 2021.

[24] X. Zhang and S. Y. Yuen, "A directional mutation operator for differential evolution algorithms," *Applied soft computing*, vol. 30, pp. 529–548, 2015.

[25] G. Sun, Y. Lan, and R. Zhao, "Differential evolution with gaussian mutation and dynamic parameter adjustment," *Soft Computing*, vol. 23, no. 5, pp. 1615–1642, 2019.

[26] M. Pant, R. Thangaraj, A. Abraham, and C. Grosan, "Differential evolution with laplace mutation operator," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 2841–2849.

[27] J. C. Lai, F. H. Leung, and S.-H. Ling, "A new differential evolution with wavelet theory based mutation operation," in *2009 IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 1116–1122.

[28] G. Sun and Y. Cai, "A novel neighborhood-dependent mutation operator for differential evolution," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, vol. 1. IEEE, 2017, pp. 837–841.

[29] S. Biswas, S. Kundu, and S. Das, "An improved parent-centric mutation with normalized neighborhoods for inducing niching behavior in differential evolution," *IEEE transactions on cybernetics*, vol. 44, no. 10, pp. 1726–1737, 2014.

[30] Y. Zhou, X. Li, and L. Gao, "A differential evolution

algorithm with intersect mutation operator," *Applied Soft Computing*, vol. 13, no. 1, pp. 390–401, 2013.

[31] M.-Y. Ameca-Alducin, E. Mezura-Montes, and N. Cruz-Ramírez, "A repair method for differential evolution with combined variants to solve dynamic constrained optimization problems," in *Proceedings of the 2015 annual conference on genetic and evolutionary computation*, 2015, pp. 241–248.

[32] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied soft computing*, vol. 11, no. 2, pp. 1679–1696, 2011.

[33] 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.

[34] M. Pant, M. Ali, and V. P. Singh, "Differential evolution with parent centric crossover," in *2008 Second UKSIM European Symposium on Computer Modeling and Simulation*. IEEE, 2008, pp. 141–146.

[35] 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.

[36] Y. Cai and J. Wang, "Differential evolution with hybrid linkage crossover," *Information Sciences*, vol. 320, pp. 244–287, 2015.

[37] R. Storn and K. Price, "Minimizing the real functions of the icec'96 contest by differential evolution," in *Proceedings of IEEE international conference on evolutionary computation*. IEEE, 1996, pp. 842–844.

[38] R. Gämperle, S. D. Müller, and P. Koumoutsakos, "A parameter study for differential evolution," *Advances in intelligent systems, fuzzy systems, evolutionary computation*, vol. 10, no. 10, pp. 293–298, 2002.

[39] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *2005 IEEE congress on evolutionary computation*, vol. 1. IEEE, 2005, pp. 506–513.

[40] Z. Zhao, J. Yang, Z. Hu, and H. Che, "A differential evolution algorithm with self-adaptive strategy and control parameters based on symmetric latin hypercube design for unconstrained optimization problems," *European Journal of Operational Research*, vol. 250, no. 1, pp. 30–45, 2016.

[41] 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, 2008.

[42] A. Ghosh, S. Das, A. Chowdhury, and R. Giri, "An improved differential evolution algorithm with fitness-based adaptation of the control parameters," *Information Sciences*, vol. 181, no. 18, pp. 3749–3765, 2011.

[43] A. Ortiz, J. Cabrera, F. Nadal, and A. Bonilla, "Dimensional synthesis of mechanisms using differential evolution with auto-adaptive control parameters," *Mechanism and Machine Theory*, vol. 64, pp. 210–229, 2013.

[44] 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. 10, no. 6, pp. 646–657, 2006.

[45] H. Guo, Y. Li, J. Li, H. Sun, D. Wang, and X. Chen, "Differential evolution improved with self-adaptive control parameters based on simulated annealing," *Swarm and Evolutionary Computation*, vol. 19, pp. 52–67, 2014.

[46] A. Zamuda and J. Brest, "Self-adaptive control parameters^2 randomization frequency and propagations in differential evolution," *Swarm and Evolutionary Computation*, vol. 25, pp. 72–99, 2015.

[47] G. Sun, G. Xu, and N. Jiang, "A simple differential evolution with time-varying strategy for continuous optimization," *Soft Computing*, vol. 24, no. 4, pp. 2727–2747, 2020.

[48] J. Zhang, X. Zhu, Y. Wang, and M. Zhou, "Dual-environmental particle swarm optimizer in noisy and noise-free environments," *IEEE transactions on cybernetics*, vol. 49, no. 6, pp. 2011–2021, 2018.

[49] 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.

[50] A. Draa, S. Bouzoubia, and I. Boukhalfa, "A sinusoidal differential evolution algorithm for numerical optimisation," *Applied Soft Computing*, vol. 27, pp. 99–126, 2015.

[51] W. Deng, S. Shang, X. Cai, H. Zhao, Y. Song, and J. Xu, "An improved differential evolution algorithm and its application in optimization problem," *Soft Computing*, vol. 25, no. 7, pp. 5277–5298, 2021.

[52] G. Wu, R. Mallipeddi, P. N. Suganthan, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Information Sciences*, vol. 329, pp. 329–345, 2016.

[53] C. Xu, H. Huang, and S. Ye, "A differential evolution with replacement strategy for real-parameter numerical optimization," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 1617–1624.

[54] Y. Wang, S. Gao, M. Zhou, and Y. Yu, "A multi-layered gravitational search algorithm for function optimization and real-world problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 94–109, 2020.

[55] J. Lu, X. Zhou, Y. Ma, M. Wang, J. Wan, and W. Wang, "A novel artificial bee colony algorithm with division of labor for solving cec 2019 100-digit challenge benchmark problems," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 387–394.

[56] F. Wilcoxon, "Individual comparisons by ranking methods. biom. bull., 1, 80–83," 1945.

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

[58] O. Olorunda and A. P. Engelbrecht, "Measuring exploration/exploitation in particle swarms using swarm

diversity," in *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1128–1134.

[59] B. Morales-Castañeda, D. Zaldivar, E. Cuevas, F. Fausto, and A. Rodríguez, "A better balance in metaheuristic algorithms: Does it exist?" *Swarm and Evolutionary Computation*, vol. 54, p. 100671, 2020.

[60] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with computers*, vol. 29, no. 1, pp. 17–35, 2013.

[61] M. Cui, L. Li, M. Zhou, and A. Abusorrah, "Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, 2021.

[62] H. Pan, L. Wang, and B. Liu, "Particle swarm optimization for function optimization in noisy environment," *Applied mathematics and Computation*, vol. 181, no. 2, pp. 908–919, 2006.

SUPPLEMENTARY FILE

This part gives supplementary materials to support the research in the main body. Tables S1 to S4 give the numerical results produced by the proposed MIDE algorithm and other algorithms, which are used to compare the performance of the proposed algorithm. In these tables, the results are provided for $30d$ and $50d$ optimization problems of IEEE CEC 2014 and CEC 2017 in terms of average error (Avg error), standard deviation (Std), ranking (Rank), and statistical outcome (Stat) collected by conducting the Wilcoxon rank-sum test.

Table S1: Comparison of solution accuracy on 30-dimensional CEC 2014 problems

Problem	Result	ODE	EPSDE	SaDE	SinDE	jDE	JADE	NBOLDE	TVDE	MPEDE	MLGSA	RSDE	DLABC	MIDE
F1	Avg error	3.82E+07	1.23E+04	7.88E+05	1.08E+06	5.84E+04	2.79E+03	2.12E+08	1.51E+06	2.19E-10	2.00E+05	6.77E+04	7.66E+05	6.09E+04
	Std	6.72E+06	2.65E+04	7.19E+05	5.97E+05	3.21E+04	3.25E+03	8.28E+07	1.10E+06	1.17E-09	2.73E+05	3.17E+04	4.25E+05	3.60E+04
	Rank/Stat	12 +	3 -	9 +	10 +	4 -	2 -	13 +	11 +	1 -	7 +	6 ≈	8 +	5
F2	Avg error	1.42E-14	7.48E-14	1.04E-14	2.84E-15	1.89E-15	2.65E-14	1.92E+10	9.47E-16	1.89E-15	7.91E+03	6.75E-12	1.54E+01	9.47E-16
	Std	1.45E-14	1.47E-13	2.42E-14	8.67E-15	7.21E-15	7.21E-15	6.20E+09	5.19E-15	7.21E-15	6.65E+03	2.03E-11	8.43E+01	5.19E-15
	Rank/Stat	7 +	9 +	6 +	5 +	3.5 +	8 +	13 +	1.5 +	3.5 ≈	12 +	10 +	11 +	1.5
F3	Avg error	5.68E-14	6.37E-12	9.56E+00	5.07E-12	2.27E-14	3.28E-10	2.30E+04	1.71E-14	4.74E-14	4.90E+01	6.46E-10	4.98E+02	7.20E-14
	Std	0.00E+00	1.75E-11	3.79E+01	2.14E-11	2.83E-14	1.72E-09	1.38E+04	2.65E-14	2.15E-14	5.94E+01	2.29E-09	1.96E+03	3.64E-14
	Rank/Stat	4 -	7 +	10 +	6 +	2 -	8 +	13 +	1 -	3 -	11 +	9 +	12 +	5
F4	Avg error	1.30E+00	3.84E+00	4.34E+01	8.33E+00	6.24E+00	1.55E-13	1.94E+03	2.89E+00	1.98E-01	1.20E+02	7.52E+00	3.97E+01	1.34E-01
	Std	1.14E+00	2.24E+00	2.54E+01	2.67E+01	1.71E+01	9.19E-14	6.65E+02	1.25E+01	4.36E-01	4.22E+01	2.17E+01	4.43E+01	1.49E-01
	Rank/Stat	4 +	6 +	11 +	9 +	7 +	1 -	13 +	5 +	3 +	12 +	8 +	10 +	2
F5	Avg error	2.09E+01	2.04E+01	2.06E+01	2.05E+01	2.04E+01	2.03E+01	2.06E+01	2.05E+01	2.04E+01	2.00E+01	2.00E+01	2.09E+01	2.01E+01
	Std	4.22E-02	2.50E-02	4.76E-02	6.39E-02	3.61E-02	2.93E-02	1.80E-01	4.33E-02	4.44E-02	7.32E-04	4.04E-02	5.90E-02	5.03E-02
	Rank/Stat	12 +	5 +	10 +	8 +	6 +	4 +	11 +	9 +	7 +	1 -	2 -	13 +	3
F6	Avg error	3.99E+00	1.89E+01	7.86E+00	1.38E-01	2.13E+00	1.02E+01	2.64E+01	4.86E-01	1.37E+01	5.53E+00	1.05E+01	1.42E+01	7.36E+00
	Std	7.80E+00	1.41E+00	4.55E+00	3.20E-01	3.28E+00	1.53E+00	1.99E+00	7.52E-01	5.76E-01	2.39E+00	2.88E+00	3.39E+00	1.78E+00
	Rank/Stat	4 -	12 +	7 +	1 -	3 -	8 +	13 +	2 -	10 +	5 -	9 +	11 +	6
F7	Avg error	4.93E-14	8.21E-04	6.80E-03	3.79E-14	7.58E-14	8.72E-14	2.11E+02	8.72E-14	3.25E-11	5.41E-03	2.88E-02	4.18E-03	5.31E-14
	Std	5.73E-14	3.19E-03	1.05E-02	5.45E-14	5.45E-14	4.89E-14	7.04E+01	4.89E-14	8.33E-10	8.12E-03	2.85E-02	7.34E-03	5.77E-14
	Rank/Stat	2 -	8 +	11 +	1 -	4 +	5.5 +	13 +	5.5 +	7 +	10 +	12 +	9 +	3
F8	Avg error	8.41E+01	0.00E+00	1.99E-01	7.40E-02	0.00E+00	0.00E+00	1.46E+02	7.53E+00	7.87E+00	7.15E+01	5.94E-08	1.47E-07	0.00E+00
	Std	7.32E+00	0.00E+00	4.82E-01	2.51E-01	0.00E+00	0.00E+00	2.62E+01	2.69E+00	9.34E-01	1.61E+01	2.39E-07	8.06E-07	0.00E+00
	Rank/Stat	12 +	2.5 +	8 +	7 +	2.5 +	2.5 +	13 +	9 +	10 ≈	11 +	5 +	6 +	2.5
F9	Avg error	1.69E+02	4.51E+01	3.89E+01	2.94E+01	4.16E+01	2.58E+01	1.75E+02	4.64E+01	2.69E+01	3.53E+01	7.21E+01	6.05E+01	2.17E+01
	Std	6.56E+00	6.38E+00	9.17E+00	6.84E+00	6.44E+00	4.78E+00	2.77E+01	1.15E+01	8.21E+00	2.10E+01	1.51E+01	1.51E+01	3.42E+00
	Rank/Stat	12 +	8 +	6 +	4 +	7 +	2 +	13 +	9 +	3 +	5 +	11 +	10 +	1
F10	Avg error	2.77E+03	2.23E-01	1.82E+01	6.24E+00	6.94E-04	3.47E-03	3.14E+03	1.18E+02	1.22E+00	1.42E+03	6.22E-01	1.33E+01	2.31E-01
	Std	2.47E+02	2.72E-01	2.22E+01	2.64E+00	3.80E-03	9.60E-03	5.24E+02	1.03E+02	6.16E-01	3.24E+02	6.90E-01	4.82E+00	6.37E-02
	Rank/Stat	12 +	3 -	9 +	7 +	1 -	2 -	13 +	10 +	6 +	11 +	5 ≈	8 +	4
F11	Avg error	6.32E+03	3.56E+03	4.28E+03	1.92E+03	2.48E+03	1.69E+03	4.53E+03	3.12E+03	2.40E+03	2.44E+03	2.83E+03	2.96E+03	1.31E+03
	Std	2.18E+02	4.34E+02	3.11E+02	4.85E+02	2.30E+02	2.09E+02	1.16E+03	4.70E+02	4.12E+02	4.14E+02	5.36E+02	7.55E+02	2.39E+02
	Rank/Stat	13 +	10 +	11 +	3 +	6 +	2 +	12 +	9 +	4 +	5 +	7 +	8 +	1
F12	Avg error	1.98E+00	5.09E-01	8.89E-01	4.90E-01	4.80E-01	2.76E-01	5.97E-01	7.61E-01	5.07E-01	3.71E-02	2.18E-01	2.33E+00	1.43E-01
	Std	2.39E-01	7.62E-02	9.68E-02	1.52E-01	6.48E-02	3.67E-02	3.15E-01	9.78E-02	7.50E-02	2.61E-02	6.24E-02	5.27E-01	3.46E-02
	Rank/Stat	12 +	8 +	11 +	6 +	5 +	4 +	9 +	10 +	7 +	1 -	3 +	13 +	2
F13	Avg error	3.38E-01	2.49E-01	2.68E-01	1.62E-01	3.00E-01	2.05E-01	3.89E+00	2.21E-01	2.15E-01	9.36E-02	4.48E-01	2.82E-01	1.91E-01
	Std	4.30E-02	3.15E-02	3.51E-02	3.76E-02	3.67E-02	3.24E-02	6.09E-01	4.07E-02	3.17E-02	2.27E-02	1.09E-01	5.58E-02	4.92E-02
	Rank/Stat	11 +	7 +	8 +	2 -	10 +	4 +	13 +	6 +	5 +	1 -	12 +	9 +	3
F14	Avg error	2.57E-01	2.81E-01	2.65E-01	2.20E-01	2.73E-01	2.28E-01	7.58E+01	2.41E-01	2.73E-01	2.02E-01	3.34E-01	4.71E-01	2.62E-01
	Std	2.50E-02	4.08E-02	3.82E-02	2.79E-02	2.77E-02	2.66E-02	2.30E+01	3.36E-02	2.64E-02	3.13E-02	1.42E-01	1.55E-01	3.51E-02
	Rank/Stat	5 -	10 +	7 +	2 -	8.5 +	3 -	13 +	4 -	8.5 +	1 -	11 +	12 +	6
F15	Avg error	1.58E+01	5.75E+00	4.47E+00	3.72E+00	5.73E+00	3.13E+00	8.39E+03	7.09E+00	4.16E+00	3.22E+00	1.15E+01	5.51E+00	2.40E+00
	Std	1.14E+00	6.32E-01	1.57E+00	6.61E-01	6.51E-01	4.16E-01	8.46E+03	9.49E-01	8.59E-01	7.01E-01	5.96E+00	1.81E+00	3.56E-01
	Rank/Stat	12 +	9 +	6 +	4 +	8 +	2 +	13 +	10 +	5 +	3 +	11 +	7 +	1
F16	Avg error	1.22E+01	1.13E+01	1.15E+01	9.50E+00	9.80E+00	9.42E+00	1.10E+01	9.65E+00	1.01E+01	1.11E+01	1.03E+01	1.09E+01	8.79E+00
	Std	2.05E-01	4.21E-01	2.81E-01	5.58E-01	3.87E-01	3.37E-01	7.60E-01	6.25E-01	3.74E-01	4.59E-01	8.14E-01	8.57E-01	4.88E-01
	Rank/Stat	13 +	11 +	12 +	3 +	5 +	2 +	9 +	4 +	6 +	10 +	7 +	8 +	1
F17	Avg error	7.85E+05	4.79E+04	3.69E+04	8.10E+04	1.43E+03	1.13E+03	3.65E+06	3.22E+05	2.31E+02	2.96E+05	2.10E+03	1.15E+05	7.60E+03
	Std	3.36E+05	5.68E+04	3.98E+04	7.53E+04	1.12E+03	3.71E+02	4.68E+06	2.08E+05	1.66E+02	3.28E+05	1.01E+03	9.43E+04	9.60E+03
	Rank/Stat	12 +	7 +	6 +	8 +	3 -	2 -	13 +	11 +	1 -	10 +	4 -	9 +	5
F18	Avg error	5.80E+02	6.57E+02	6.77E+02	3.78E+02	1.76E+01	2.54E+02	7.38E+06	1.68E+03	1.48E+01	2.86E+02	1.55E+02	5.45E+03	6.68E+01
	Std	3.41E+02	1.53E+03	1.02E+03	5.25E+02	6.96E+00	8.61E+02	3.03E+07	2.80E+03	5.88E+00	2.81E+02	5.75E+01	7.14E+03	6.59E+01
	Rank/Stat	8 +	9 +	10 +	7 +	2 -	5 +	13 +	11 +	1 -	6 +	4 +	12 +	3
F19	Avg error	5.78E+00	1.33E+01	9.15E+00	3.43E+00	4.41E+00	4.34E+00	1.24E+02	4.63E+00	3.68E+00	1.62E+01	1.11E+01	9.82E+00	4.41E+00
	Std	4.77E-01	1.15E+00	2.02E+00	7.46E-01	7.28E-01	7.00E-01	4.61E+01	8.25E-01	6.75E-01	1.30E+01	1.22E+01	1.09E+01	5.56E-01
	Rank/Stat	7 +	11 +	8 +	1 -	5 +	3 -	13 +	6 +	2 -	12 +	10 +	9 +	4
F20	Avg error	7.51E+01	7.22E+01	2.22E+02	1.13E+01	1.09E+01	2.20E+03	1.16E+04	1.81E+01	9.15E+00	2.08E+03	8.41E+01	3.42E+03	1.21E+01
	Std	6.33E+00	1.11E+02	3.94E+02	5.44E+00	3.83E+00	2.67E+03	8.33E+03	6.58E+00	3.88E+00	5.91E+02	4.12E+01	2.12E+03	6.81E+00
	Rank/Stat	7 +	6 +	9 +	3 -	2 -	11 +	13 +	5 +	1 -	10 +	8 +	12 +	4
F21	Avg error	1.72E+04	1.21E+04	2.73E+03	6.45E+03	1.81E+02	9.39E+03	4.72E+05	2.21E+04	1.10E+02	8.26E+03	6.70E+02	7.32E+04	1.61E+03
	Std	6.92E+03	1.47E+04	4.27E+03	7.90E+03	1.62E+02	4.99E+04	8.07E+05	1.92E+04	1.05E+02	2.03E+03	2.24E+02	8.01E+04	1.22E+03
	Rank/Stat	10 +	9 +	5 +	6 +	2 -	8 +	13 +	11 +	1 -	7 +	3 -	12 +	4
F22	Avg error	1.29E+02	2.12E+02	1.27E+02	7.28E+01	1.52E+02	1.30E+02	6.23E+02	2.30E+02	1.18E+02	6.20E+02	3.08E+02	3.47E+02	1.43E+02
	Std	6.04E+01	8.63E+01	7.83E+01	6.18E+01	7.41E+01	6.74E+01	1.71E+02	1.02E+02	6.38E+01	2.35E+02	1.40E+02	1.75E+02	5.08E+01
	Rank/Stat	4 -	8 +	3 -	1 -	7 +	5 -	13 +	9 +	2 ≈	12 +	10 +	11 +	6
F23	Avg error	3.15E+02	3.15E+02	3.15E+02	3.15E+02	3.15E+02	3.15E+02	4.15E+02	3.15E+02	3.15E+02	2.58E+02	3.15E+02	3.15E+02	3.15E+02
	Std	5.78E-14	5.78E-14	5.78E-14	5.78E-14	5.78E-14	5.78E-14	3.88E+01	5.78E-14	5.78E-14	5.88E+01	3.70E-13	5.78E-14	5.78E-14
	Rank/Stat	6.5 ≈	6.5 ≈	6.5 ≈	6.5 ≈	6.5 ≈	6.5 ≈	13 +	6.5 ≈	6.5 ≈	1 -	12 +	6.5 ≈	6.5
F24	Avg error	2.23E+02	2.30E+02	2.28E+02	2.23E+02	2.24E+02	2.25E+02	2.84E+02	2.26E+02	2.26E+02	2.00E+02	2.32E+02	2.34E+02	2.25E+02
	Std	1.20E+00	5.93E+00	4.42E+00	1.61E+00	8.15E-01	2.46E+00	1.00E+01	3.33E+00	4.30E+00	7.49E-02	7.40E+00	6.31E+00	1.34E+00
	Rank/Stat	2 -	10 +	9 +	3 -	4 -	6 +	13 +	7 +	8 ≈	1 -	11 +	12 +	5
F25	Avg error	2.14E+02	2.01E+02	2.03E+02	2.04E+02	2.03E+02	2.06E+02	2.07E+02	2.05E+02	2.03E+02	2.00E+02	2.05E+02	2.06E+02	2.07E+02
	Std	1.75E+00	8.99E-01	4.91E+00	4.17E-01	4.45E-01	2.40E+00	1.14E+01	1.20E+00	2.06E-01	1.73E-06	2.62E+00	2.37E+00	3.34E+00
	Rank/Stat	13 +	2 -	4 -	6 -	5 -	10 -	11 -	7 -	3 -	1 -	8 -	9 ≈	12
F26	Avg error	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.35E+02	1.17E+02	1.00E+02	1.80E+02	1.15E+02	1.00E+02	1.04E+02
	Std	3.36E-02	3.92E-02	4.75E-02	4.20E-02	4.00E-02	2.98E-02	6.09E+01	3.79E+01	3.02E-02	4.05E+01	4.87E+01	6.47E-02	1.82E+01
	Rank/Stat	8 -	5 -	7 -	1 -	6 -	3 -	12 +	11 +	2 ≈	13 +	10 +	4 -	9
F27	Avg error	3.00E+02	8.52E+02	4.86E+02	3.01E+02	3.56E+02	3.39E+02	8.48E+02	3.41E+02	3.69E+02	3.09E+02	6.41E+02	7.28E+02	3.12E+02
	Std	2.73E-13	8.09E+01	1.63E+02	6.80E+00	5.01E+01	4.57E+01	3.06E+02	4.63E+01	4.55E+01	2.71E+01	1.58E+02	1.13E+02	2.79E+01
	Rank/Stat	1 -	13 +	9 +	2 -	7 +	5 +	12 +	6 +	8 +	3 ≈	10 +	11 +	4
F28	Avg error	7.86E+02	4.01E+02	4.20E+02	8.07E+02	7.88E+02	7.86E+02	2.83E+03	8.28E+02	8.25E+02	3.00E+02	1.23E+03	9.88E+02	8.30E+02
	Std	2.00E+01	2.07E+01	1.02E+01	3.89E+01	2.92E+01	4.37E+01	6.15E+02	5.06E+01	3.98E+01	1.24E-04	2.89E+02	1.64E+02	2.30E+01
	Rank/Stat	4 -	2 -	3 -	7 -	6 -	5 -	13 +	9 -	8 ≈	1 -	12 +	11 +	10
F29	Avg error	3.77E+03	2.14E+02	2.07E+02	1.44E+03	7.75E+02	7.04E+02	3.50E+07	1.32E+03	6.64E+02	7.76E+02	2.00E+06	1.40E+06	9.62E+02
	Std	1.29E+03	1.25E+00	1.88E+00	1.94E+02	1.29E+02	7.37E+01	2.88E+07	1.46E+02	1.66E+02	2.05E+02	3.68E+06	3.18E+06	1.04E+02
	Rank/Stat	10 +	2 -	1 -	9 +	5 -	4 -	13 +	8 +	3 -	6 -	12 ≈	11 +	7
F30	Avg error	2.25E+03	5.59E+02	3.37E+02	1.26E+03	1.05E+03	1.32E+03	2.57E+05	1.73E+03	6.49E+02	2.26E+03	2.51E+03	2.78E+03	2.22E+03
	Std	4.73E+02	1.49E+02	8.72E+01	4.47E+02	3.87E+02	3.90E+02	1.70E+05	5.74E+02	2.38E+02	5.47E+02	1.33E+03	8.58E+02	7.75E+02
	Rank/Stat	9 +	2 -	1 -	5 -	4 -	6 -	13 +	7 -	3 -	10 ≈	11 ≈	12 +	8
	#(+)	20	22	23	16	15	17	29	23	13	18	22	27	NA
	#(-)	9	7	6	13	14	12	1	6	10	10	4	1	NA
	#(≈)	1	1	1	1	1	1	0	1	7	2	4	2	NA

Table S2: Comparison of solution accuracy on 30-dimensional CEC 2017 problems

Problem	Result	ODE	EPSDE	SaDE	SinDE	jDE	JADE	NBOLDE	TVDE	MPEDE	MLGSA	RSDE	DLABC	MIDE
F1	Avg error	2.04E-04	7.35E-13	1.94E+03	1.28E-02	1.09E-14	1.61E-14	1.40E+10	1.09E-14	1.89E-15	3.73E+03	1.96E-10	7.74E+03	4.78E-14
	Std	1.83E-04	1.48E-12	3.63E+03	6.88E-02	6.11E-15	8.12E-15	4.84E+09	6.11E-15	4.91E-15	3.84E+03	3.34E-10	6.93E+03	1.92E-14
	Rank/Stat	8 +	6 +	10 +	9 +	2.5 -	4 -	13 +	2.5 -	1 -	11 +	7 +	12 +	5
F2	Avg error	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	0.00E+00	0.00E+00	0.00E+00
	Std	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	0.00E+00	0.00E+00	0.00E+00
	Rank/Stat	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7
F3	Avg error	1.93E+04	5.21E+03	8.43E-02	7.04E+02	6.96E-04	5.89E+03	2.31E+03	2.30E+01	7.50E-10	7.31E-09	1.65E+02	6.71E+04	3.32E-06
	Std	3.73E+03	1.99E+04	3.66E-01	4.63E+02	1.18E-03	1.38E+04	1.48E+03	2.09E+01	4.06E-09	2.36E-09	5.32E+01	9.06E+03	1.64E-05
	Rank/Stat	12 +	10 ≈	5 +	8 +	4 +	11 ≈	9 +	6 +	1 -	2 ≈	7 +	13 +	3
F4	Avg error	8.30E+01	1.41E+00	5.18E+01	8.42E+01	5.69E+01	2.97E+01	1.99E+03	5.14E+01	9.64E+01	1.13E+02	3.71E+01	6.28E+01	5.28E+01
	Std	2.05E+00	1.76E+00	2.99E+01	5.35E+00	1.58E+01	3.12E+01	8.95E+02	2.65E+01	2.02E+01	2.24E+01	3.05E+01	6.98E+00	2.09E+01
	Rank/Stat	9 +	1 -	5 ≈	10 +	7 ≈	2 -	13 +	4 ≈	11 +	12 +	3 ≈	8 +	6
F5	Avg error	1.62E+02	4.09E+01	3.69E+01	2.92E+01	4.05E+01	2.75E+01	1.89E+02	4.67E+01	2.65E+01	7.90E+01	8.01E+01	6.28E+01	2.17E+01
	Std	8.87E+00	5.55E+00	9.90E+00	7.44E+00	6.31E+00	4.78E+00	3.56E+01	1.12E+01	8.81E+00	2.86E+01	2.33E+01	2.09E+01	3.23E+00
	Rank/Stat	12 +	7 +	5 +	4 +	6 +	3 +	13 +	8 +	2 +	10 +	11 +	9 +	1
F6	Avg error	1.14E-13	1.14E-13	9.13E-06	1.14E-13	1.14E-13	1.36E-13	4.31E+01	7.07E-07	1.44E-13	4.07E-01	4.13E-03	4.11E-08	1.14E-13
	Std	0.00E+00	0.00E+00	5.00E-05	0.00E+00	0.00E+00	4.63E-14	5.90E+00	2.87E-06	5.11E-14	6.99E-01	4.96E-03	2.00E-07	0.00E+00
	Rank/Stat	3 ≈	3 ≈	10 +	3 ≈	3 ≈	6 +	13 +	9 +	7 +	12 +	11 +	8 ≈	3
F7	Avg error	2.06E+02	7.39E+01	6.57E+01	6.16E+01	7.66E+01	5.42E+01	4.31E+02	8.59E+01	5.59E+01	4.20E+01	1.29E+02	9.27E+01	5.22E+01
	Std	8.06E+00	7.05E+00	9.09E+00	6.84E+00	5.68E+00	3.99E+00	7.42E+01	1.10E+01	6.17E+00	4.38E+00	2.33E+01	2.10E+01	4.08E+00
	Rank/Stat	12 +	7 +	6 +	5 +	8 +	3 ≈	13 +	9 +	4 +	1 -	11 +	10 +	2
F8	Avg error	1.65E+02	4.30E+01	3.40E+01	3.18E+01	4.44E+01	2.58E+01	1.60E+02	4.76E+01	1.75E+02	4.83E+01	7.19E+01	6.78E+01	2.32E+01
	Std	9.87E+00	6.54E+00	9.60E+00	6.11E+00	6.23E+00	3.76E+00	3.07E+01	1.09E+01	7.01E+00	2.13E+01	2.56E+01	1.71E+01	3.02E+00
	Rank/Stat	12 +	5 +	4 +	3 +	6 +	2 +	11 +	7 +	13 +	8 +	10 +	9 +	1
F9	Avg error	1.89E-14	1.69E-01	2.20E+00	1.14E-14	3.79E-15	7.58E-15	2.63E+03	2.11E-02	2.98E-03	2.41E-02	5.24E+02	2.76E+01	3.79E-15
	Std	4.31E-14	6.26E-01	3.23E+00	3.47E-14	2.08E-14	2.88E-14	7.92E+02	8.49E-02	1.63E-02	1.16E-01	4.69E+02	3.06E+01	2.08E-14
	Rank/Stat	5 ≈	9 +	10 +	4 ≈	1.5 ≈	3 ≈	13 +	7 ≈	6 ≈	8 +	12 +	11 +	1.5
F10	Avg error	6.53E+03	3.83E+03	4.46E+03	1.95E+03	2.72E+03	1.76E+03	4.67E+03	3.34E+03	2.69E+03	2.26E+03	2.83E+03	2.86E+03	1.60E+03
	Std	3.19E+02	3.73E+02	4.32E+02	4.43E+02	2.32E+02	3.19E+02	8.46E+02	5.15E+02	3.34E+02	4.45E+02	6.71E+02	5.89E+02	2.22E+02
	Rank/Stat	13 +	10 +	11 +	3 +	6 +	2 +	12 +	9 +	5 +	4 +	7 +	8 +	1
F11	Avg error	8.93E+01	2.90E+01	6.83E+01	2.82E+01	2.17E+01	2.30E+01	7.16E+02	4.74E+01	2.07E+01	9.00E+01	1.29E+02	1.10E+02	1.23E+01
	Std	2.81E+01	1.77E+01	2.47E+01	2.67E+01	2.18E+01	1.78E+01	4.78E+02	3.02E+01	6.40E+00	2.93E+01	3.83E+01	4.74E+01	1.08E+01
	Rank/Stat	9 +	6 +	8 +	5 +	3 ≈	4 +	13 +	7 +	2 +	10 +	12 +	11 +	1
F12	Avg error	2.52E+06	3.25E+04	3.59E+04	8.65E+04	1.01E+04	2.55E+03	1.04E+09	2.51E+04	1.07E+03	1.64E+04	1.53E+04	5.84E+04	1.01E+04
	Std	1.37E+06	3.76E+04	2.59E+04	8.24E+04	8.97E+03	2.80E+03	6.22E+08	1.24E+04	4.04E+02	6.02E+03	1.45E+04	3.88E+04	5.02E+03
	Rank/Stat	12 +	8 +	9 +	11 +	4 ≈	2 -	13 +	7 +	1 -	6 +	5 ≈	10 +	3
F13	Avg error	7.98E+03	3.58E+03	6.48E+03	5.10E+03	3.61E+01	4.01E+01	6.65E+07	8.07E+03	2.38E+01	1.42E+04	1.25E+02	1.45E+04	6.80E+02
	Std	3.50E+03	1.55E+04	8.65E+03	4.56E+03	1.59E+01	2.19E+01	1.31E+08	6.66E+03	5.46E+00	4.90E+03	7.62E+01	1.74E+04	7.10E+02
	Rank/Stat	9 +	6 ≈	8 +	7 +	2 -	3 -	13 +	10 +	1 -	11 +	4 -	12 +	5
F14	Avg error	9.14E+01	6.53E+01	7.01E+01	2.86E+01	2.10E+01	1.84E+03	2.64E+04	4.13E+01	1.58E+01	1.90E+03	1.77E+02	9.30E+03	2.72E+01
	Std	5.76E+00	3.18E+01	2.55E+01	6.80E+00	1.20E+01	4.46E+03	3.84E+04	9.01E+00	1.13E+01	7.48E+02	6.06E+01	6.29E+03	7.12E+00
	Rank/Stat	8 +	6 +	7 +	4 ≈	2 -	10 +	13 +	5 +	1 -	11 +	9 +	12 +	3
F15	Avg error	1.09E+02	1.07E+02	4.08E+02	1.40E+01	1.11E+01	2.86E+02	4.77E+04	2.61E+02	8.73E+00	7.23E+02	8.23E+01	1.38E+04	1.46E+01
	Std	1.98E+01	2.65E+02	8.08E+02	8.04E+00	4.00E+00	1.35E+03	6.75E+04	6.19E+02	3.19E+00	9.10E+02	4.94E+01	1.44E+04	1.47E+01
	Rank/Stat	7 +	6 +	10 +	3 ≈	2 ≈	9 +	13 +	8 +	1 ≈	11 +	5 +	12 +	4
F16	Avg error	8.16E+02	6.03E+02	3.76E+02	1.46E+02	4.36E+02	3.97E+02	1.35E+03	2.95E+02	3.61E+02	1.13E+03	4.36E+02	7.29E+02	3.70E+02
	Std	1.51E+02	1.43E+02	1.74E+02	1.57E+02	1.62E+02	1.40E+02	2.93E+02	2.08E+02	1.67E+02	2.69E+02	2.43E+02	2.71E+02	1.30E+02
	Rank/Stat	11 +	9 +	5 ≈	1 -	7 ≈	6 ≈	13 +	2 ≈	3 ≈	12 +	8 ≈	10 +	4
F17	Avg error	1.59E+02	2.02E+02	6.48E+01	2.81E+01	8.34E+01	8.20E+01	5.04E+02	8.49E+01	5.51E+01	5.76E+02	2.75E+02	3.23E+02	3.98E+01
	Std	2.42E+01	8.48E+01	4.02E+01	1.14E+01	1.87E+01	2.88E+01	2.04E+02	7.11E+01	2.43E+01	3.02E+02	1.35E+02	1.94E+02	2.69E+01
	Rank/Stat	8 +	9 +	4 +	1 -	6 +	5 +	12 +	7 +	3 +	13 +	10 +	11 +	2
F18	Avg error	2.86E+05	1.77E+03	7.45E+03	3.39E+04	4.79E+01	9.90E+03	1.68E+05	8.17E+04	2.51E+01	4.09E+04	1.72E+02	1.71E+05	5.40E+03
	Std	7.91E+04	1.85E+03	8.09E+03	2.07E+04	2.83E+01	3.88E+04	3.10E+05	5.50E+04	4.36E+00	6.99E+03	6.28E+02	1.14E+05	3.75E+03
	Rank/Stat	13 +	4 -	6 ≈	8 +	2 -	7 +	11 +	10 +	1 -	9 +	3 -	12 +	5
F19	Avg error	4.59E+01	2.81E+01	4.28E+01	1.52E+01	1.04E+01	1.30E+01	1.75E+06	2.05E+01	1.72E+01	2.78E+03	4.66E+01	2.06E+04	1.27E+01
	Std	3.70E+00	1.94E+01	2.57E+01	3.71E+01	3.09E+00	1.02E+01	2.15E+06	2.44E+01	2.12E+00	1.05E+03	2.14E+01	2.08E+04	7.14E+00
	Rank/Stat	9 +	7 +	8 +	4 +	1 ≈	3 ≈	13 +	6 +	5 +	11 +	10 +	12 +	2
F20	Avg error	1.66E+02	1.43E+02	7.85E+01	3.05E+01	7.89E+01	8.95E+01	4.52E+02	1.24E+02	6.82E+01	1.93E+02	2.05E+02	3.04E+02	7.78E+01
	Std	4.08E+01	5.72E+01	6.91E+01	3.82E+01	4.22E+01	4.46E+01	2.15E+02	7.22E+01	5.11E+01	1.84E+01	1.19E+02	1.48E+02	6.13E+01
	Rank/Stat	9 +	8 +	4 ≈	1 -	5 ≈	6 ≈	13 +	7 +	2 ≈	10 +	11 +	12 +	3
F21	Avg error	3.60E+02	2.47E+02	2.35E+02	2.29E+02	2.45E+02	2.24E+02	3.78E+02	2.43E+02	2.30E+02	2.81E+02	2.91E+02	2.69E+02	2.24E+02
	Std	1.04E+01	7.81E+00	8.81E+00	6.45E+00	5.34E+00	5.58E+00	3.39E+01	6.99E+00	7.40E+00	1.84E+01	2.82E+01	1.93E+01	4.71E+00
	Rank/Stat	12 +	8 +	5 +	3 +	7 +	2 ≈	13 +	6 +	4 +	10 +	11 +	9 +	1
F22	Avg error	1.00E+02	3.64E+03	8.96E+02	1.00E+02	1.00E+02	1.00E+02	2.63E+03	3.11E+02	1.00E+02	1.00E+02	1.81E+03	3.06E+02	1.00E+02
	Std	1.96E-13	1.54E+03	1.81E+03	1.39E-13	0.00E+00	0.00E+00	1.32E+03	8.35E+02	0.00E+00	1.24E-05	1.81E+03	8.05E+02	0.00E+00
	Rank/Stat	6 +	13 +	10 +	5 ≈	2.5 ≈	2.5 ≈	12 +	9 ≈	2.5 ≈	7 +	11 +	8 +	2.5
F23	Avg error	5.07E+02	3.94E+02	3.91E+02	3.77E+02	3.87E+02	3.72E+02	7.33E+02	3.86E+02	3.96E+02	4.22E+02	4.48E+02	4.26E+02	3.75E+02
	Std	1.03E+01	9.19E+00	1.30E+01	7.26E+00	5.63E+00	5.64E+00	6.57E+01	9.48E+00	9.46E+00	2.30E+01	2.37E+01	1.55E+01	3.75E+00
	Rank/Stat	12 +	7 +	6 +	3 ≈	5 +	1 -	13 +	4 +	8 +	9 +	11 +	10 +	2
F24	Avg error	5.86E+02	4.64E+02	4.68E+02	4.44E+02	4.57E+02	4.39E+02	8.24E+02	4.64E+02	4.44E+02	4.56E+02	5.23E+02	4.97E+02	4.48E+02
	Std	8.97E+00	7.02E+00	1.06E+01	6.86E+00	6.12E+00	5.13E+00	8.78E+01	1.43E+01	6.64E+00	5.05E+01	3.31E+01	1.76E+01	5.32E+00
	Rank/Stat	12 +	7 +	9 +	3 -	6 +	1 -	13 +	8 +	2 -	5 ≈	11 +	10 +	4
F25	Avg error	3.87E+02	3.79E+02	3.78E+02	3.87E+02	3.87E+02	3.87E+02	1.07E+03	3.87E+02	3.87E+02	3.89E+02	3.90E+02	3.90E+02	3.87E+02
	Std	1.06E-01	2.89E-01	1.70E+00	1.33E-01	1.52E-01	2.68E-01	1.74E+02	3.37E-01	6.30E-02	5.76E+00	1.22E+01	1.23E+01	2.85E-01
	Rank/Stat	4 -	2 -	1 -	5 -	6 -	8 ≈	13 +	9 ≈	3 -	10 ≈	12 ≈	11 ≈	7
F26	Avg error	2.49E+03	1.26E+03	1.33E+03	1.19E+03	1.30E+03	1.18E+03	4.56E+03	1.12E+03	1.10E+03	2.57E+02	2.10E+03	1.89E+03	1.22E+03
	Std	1.25E+02	1.59E+02	2.60E+02	6.80E+01	6.41E+01	7.09E+01	7.48E+02	3.37E+02	2.99E+02	5.04E+01	5.82E+02	2.71E+02	8.97E+01
	Rank/Stat	12 +	7 ≈	9 +	5 ≈	8 +	4 -	13 +	3 ≈	2 ≈	1 -	11 +	10 +	6
F27	Avg error	5.00E+02	5.00E+02	5.00E+02	5.00E+02	4.99E+02	5.05E+02	8.58E+02	5.08E+02	4.99E+02	5.63E+02	5.31E+02	5.18E+02	5.05E+02
	Std	7.39E+00	1.00E-04	1.01E-04	5.31E+00	5.07E+00	6.65E+00	1.39E+02	5.66E+00	6.82E+00	1.99E+01	2.56E+01	8.98E+00	5.21E+00
	Rank/Stat	6 -	4 -	5 -	3 -	1 -	8 ≈	13 +	9 +	2 -	12 +	11 +	10 +	7
F28	Avg error	3.58E+02	4.99E+02	4.99E+02	3.38E+02	3.23E+02	3.22E+02	1.54E+03	3.47E+02	3.41E+02	3.03E+02	3.31E+02	3.78E+02	3.38E+02
	Std	5.62E+01	2.42E+00	3.88E+00	5.43E+01	4.78E+01	4.43E+01	4.49E+02	6.00E+01	5.50E+01	1.71E+01	5.91E+01	7.19E+01	5.11E+01
	Rank/Stat	9 +	12 +	11 +	5 ≈	3 ≈	2 ≈	13 +	8 ≈	7 ≈	1 ≈	4 ≈	10 +	6
F29	Avg error	8.03E+02	4.77E+02	3.68E+02	4.35E+02	5.04E+02	4.81E+02	1.63E+03	5.05E+02	4.48E+02	8.28E+02	6.04E+02	7.86E+02	4.65E+02
	Std	7.56E+01	9.74E+01	7.56E+01	3.05E+01	2.71E+01	4.00E+01	4.37E+02	5.04E+01	3.79E+01	2.91E+02	1.61E+02	2.17E+02	2.58E+01
	Rank/Stat	11 +	5 ≈	1 -	2 -	7 +	6 +	13 +	8 +	3 ≈	12 +	9 +	10 +	4
F30	Avg error	2.56E+04	2.22E+02	6.04E+02	4.40E+03	2.12E+03	2.33E+03	1.55E+07	4.15E+03	2.04E+03	3.94E+03	2.54E+03	7.89E+03	3.07E+03
	Std	1.30E+04	9.36E+00	1.81E+03	1.34E+03	1.09E+02	1.05E+03	2.25E+07	1.30E+03	1.51E+02	4.25E+02	4.79E+02	4.72E+03	4.07E+02
	Rank/Stat	12 +	1 -	2 -	10 +	4 -	5 -	13 +	9 +	3 -	8 +	6 -	11 +	7
	#(+)	25	19	21	14	11	10	29	21	11	23	21	27	NA
	#(-)	2	5	4	7	7	8	0	1	10	2	3	0	NA
	#(≈)	3	6	5	9	12	12	1	8	9	5	6	3	NA

Table S3: Comparison of solution accuracy on 50-dimensional CEC 2014 problems

Problem	Result	ODE	EPSDE	SaDE	SinDE	jDE	JADE	NBOLDE	TVDE	MPEDE	MLGSA	RSDE	DLABC	MIDE
F1	Avg error	1.77E+08	1.05E+06	1.33E+06	2.81E+06	4.50E+05	2.35E+04	7.47E+08	4.04E+06	4.32E+04	6.69E+05	3.29E+05	1.38E+06	4.91E+05
	Std	2.49E+07	3.43E+06	7.44E+05	1.08E+06	2.34E+05	1.29E+04	2.59E+08	1.55E+06	2.38E+04	3.19E+05	1.81E+05	8.21E+05	1.97E+05
	Rank/Stat	12 +	7 +	8 +	10 +	4 ≈	1 -	13 +	11 +	2 -	6 +	3 -	9 +	5
F2	Avg error	4.37E+03	1.70E-08	4.26E+03	4.08E+03	7.73E-09	1.89E-13	7.12E+10	4.41E+03	4.64E-14	6.01E+03	2.88E+03	1.25E+04	3.18E+01
	Std	4.59E+03	3.83E-08	4.49E+03	4.66E+03	1.41E-08	3.54E-13	1.08E+10	3.62E+03	1.75E-14	6.09E+03	5.28E+03	1.22E+04	1.08E+02
	Rank/Stat	9 +	4 -	8 +	7 +	3 -	2 -	13 +	10 +	1 -	11 +	6 +	12 +	5
F3	Avg error	1.23E+04	8.26E-04	3.29E+03	4.40E+02	9.40E-09	2.77E+03	4.33E+04	6.56E+01	2.95E-04	1.93E+03	1.55E-01	1.08E+04	3.70E-04
	Std	2.36E+03	3.28E-03	2.57E+03	3.49E+02	2.46E-08	2.55E+03	2.44E+04	3.57E+01	6.59E-04	1.08E+03	2.66E-01	7.46E+03	1.31E-03
	Rank/Stat	12 +	4 ≈	10 +	7 +	1 -	9 +	13 +	6 +	2 ≈	8 +	5 +	11 +	3
F4	Avg error	9.25E+01	3.57E+01	8.55E+01	9.36E+01	8.37E+01	1.88E+01	1.20E+04	7.29E+01	1.09E+02	6.61E+01	6.82E+01	8.76E+01	6.45E+01
	Std	6.19E+00	2.53E+01	3.65E+01	5.65E+00	1.95E+01	3.37E+01	3.30E+03	2.96E+01	3.19E+01	2.85E+01	3.73E+01	3.27E+01	3.79E+01
	Rank/Stat	10 +	2 -	8 +	11 +	7 ≈	1 -	13 +	6 ≈	12 +	4 ≈	5 ≈	9 +	3
F5	Avg error	2.11E+01	2.06E+01	2.07E+01	2.07E+01	2.05E+01	2.04E+01	2.07E+01	2.09E+01	2.05E+01	2.00E+01	2.01E+01	2.11E+01	2.02E+01
	Std	3.53E-02	7.48E-02	4.44E-02	5.11E-02	3.22E-02	2.85E-02	1.41E-01	3.68E-02	5.14E-02	2.16E-04	5.21E-02	3.15E-02	6.54E-02
	Rank/Stat	12 +	7 +	10 +	8 +	5 +	4 +	9 +	11 +	6 +	1 -	2 -	13 +	3
F6	Avg error	3.10E+00	4.52E+01	1.98E+01	6.72E-01	4.18E+00	1.53E+01	5.49E+01	6.98E-01	2.85E+01	1.40E+01	2.81E+01	3.08E+01	1.15E+01
	Std	1.11E+01	2.73E+00	4.33E+00	1.14E+00	5.03E+00	7.86E+00	2.45E+00	9.14E-01	2.31E+00	3.52E+00	4.17E+00	3.70E+00	6.36E+00
	Rank/Stat	3 -	12 +	8 +	1 -	4 -	7 +	13 +	2 -	10 +	6 ≈	9 +	11 +	5
F7	Avg error	1.14E-13	8.52E-03	1.40E-02	1.67E-13	2.65E-13	4.51E-03	6.67E+02	2.47E-04	2.46E-03	3.53E-03	3.32E-02	1.46E-02	1.74E-13
	Std	0.00E+00	1.49E-02	1.45E-02	6.50E-14	1.01E-13	7.87E-03	1.14E+02	1.35E-03	4.29E-03	7.30E-03	4.32E-02	3.03E-02	7.15E-14
	Rank/Stat	1 -	9 +	10 +	2 ≈	4 +	8 +	13 +	5 +	6 ≈	7 +	12 +	11 +	3
F8	Avg error	2.26E+02	1.29E-13	1.13E+00	6.09E+00	9.85E-14	3.79E-15	3.87E+02	8.15E+00	1.92E+01	1.42E+02	2.31E-13	2.65E-01	2.20E-13
	Std	1.17E+01	3.93E-14	1.30E+00	3.60E+00	3.93E-14	2.08E-14	2.91E+01	3.61E+00	0.00E+00	5.61E+01	6.32E-14	4.48E-01	3.28E-13
	Rank/Stat	12 +	3 ≈	7 +	8 +	2 ≈	1 -	13 +	9 +	10 +	11 +	5 ≈	6 +	4
F9	Avg error	3.59E+02	1.50E+02	9.13E+01	6.56E+01	8.93E+01	5.40E+01	4.36E+02	1.65E+02	5.96E+01	1.08E+02	1.62E+02	1.69E+02	4.95E+01
	Std	1.35E+01	1.46E+01	1.73E+01	1.19E+01	1.02E+01	7.85E+00	5.07E+01	2.43E+01	1.64E+01	5.76E+01	2.80E+01	3.37E+01	6.94E+00
	Rank/Stat	12 +	8 +	6 +	4 +	5 +	2 +	13 +	10 +	3 +	7 +	9 +	11 +	1
F10	Avg error	7.95E+03	6.05E+02	3.37E+02	1.90E+01	3.08E+00	1.04E-02	7.52E+03	7.85E+01	2.66E+01	4.71E+03	2.77E+00	3.29E+01	6.03E-01
	Std	4.43E+02	6.32E+02	1.72E+02	7.59E+00	3.17E-03	1.09E-02	8.01E+02	9.68E+01	2.52E-01	7.86E+02	1.48E+00	4.17E+01	4.46E-01
	Rank/Stat	13 +	10 +	9 +	5 +	4 +	1 -	12 +	8 +	6 ≈	11 +	3 +	7 +	2
F11	Avg error	1.27E+04	9.15E+03	8.84E+03	4.29E+03	5.16E+03	3.72E+03	9.33E+03	8.80E+03	4.82E+03	6.08E+03	6.19E+03	5.19E+03	3.32E+03
	Std	3.38E+02	4.98E+02	9.62E+02	7.32E+02	4.03E+02	2.91E+02	8.68E+02	8.05E+02	6.61E+02	8.01E+02	7.77E+02	8.49E+02	1.97E+02
	Rank/Stat	13 +	11 +	10 +	3 +	5 +	2 +	12 +	9 +	4 +	7 +	8 +	6 +	1
F12	Avg error	3.13E+00	8.41E-01	1.19E+00	4.77E-01	5.08E-01	2.51E-01	9.24E-01	1.41E+00	5.42E-01	4.76E-02	2.24E-01	3.33E+00	1.48E-01
	Std	2.49E-01	9.37E-02	1.14E-01	1.96E-01	5.09E-02	3.30E-02	3.82E-01	1.89E-01	1.18E-01	3.85E-02	5.86E-02	3.06E-01	2.80E-02
	Rank/Stat	12 +	8 +	10 +	5 +	6 +	4 +	9 +	11 +	7 +	1 -	3 +	13 +	2
F13	Avg error	4.68E-01	3.64E-01	4.25E-01	2.37E-01	3.78E-01	3.25E-01	5.30E+00	3.13E-01	2.78E-01	2.03E-01	6.06E-01	3.59E-01	2.84E-01
	Std	4.79E-02	6.43E-02	7.45E-02	4.91E-02	2.79E-02	4.21E-02	5.18E-01	4.28E-02	2.41E-02	3.69E-02	8.39E-02	4.68E-02	3.52E-02
	Rank/Stat	11 +	8 +	10 +	2 -	9 +	6 +	13 +	5 +	3 ≈	1 -	12 +	7 +	4
F14	Avg error	3.34E-01	3.39E-01	3.15E-01	2.48E-01	3.17E-01	2.84E-01	1.67E+02	3.28E-01	3.33E-01	2.75E-01	3.56E-01	5.51E-01	3.39E-01
	Std	1.27E-01	6.19E-02	3.21E-02	3.28E-02	3.81E-02	2.68E-02	2.73E+01	1.37E-01	2.83E-02	3.30E-02	1.52E-01	2.62E-01	8.40E-02
	Rank/Stat	8 ≈	10 ≈	4 ≈	1 -	5 ≈	3 -	13 +	6 -	7 -	2 -	11 ≈	12 +	9
F15	Avg error	3.40E+01	1.81E+01	1.69E+01	7.59E+00	1.18E+01	8.00E+00	3.83E+05	1.92E+01	5.76E+00	5.61E+00	4.70E+01	1.55E+01	5.31E+00
	Std	1.07E+00	2.39E+00	3.94E+00	1.33E+00	1.23E+00	1.07E+00	3.08E+05	2.52E+00	1.30E+00	8.16E-01	1.30E+01	5.60E+00	7.69E-01
	Rank/Stat	11 +	9 +	8 +	4 +	6 +	5 +	13 +	10 +	3 ≈	2 ≈	12 +	7 +	1
F16	Avg error	2.21E+01	2.06E+01	2.09E+01	1.83E+01	2.42E+01	1.77E+01	2.02E+01	1.98E+01	1.84E+01	1.99E+01	1.84E+01	2.16E+01	1.69E+01
	Std	2.28E-01	3.41E-01	3.29E-01	6.92E-01	3.70E-01	4.59E-01	1.20E+00	5.05E-01	5.67E-01	7.35E-01	8.34E-01	7.09E-01	5.09E-01
	Rank/Stat	12 +	9 +	10 +	3 +	13 +	2 +	8 +	6 +	4 +	7 +	5 +	11 +	1
F17	Avg error	8.75E+06	1.74E+05	2.17E+05	3.19E+05	2.25E+04	2.63E+03	3.83E+07	1.32E+06	1.98E+03	7.91E+04	1.97E+04	3.08E+05	3.20E+04
	Std	2.30E+06	1.18E+05	1.18E+05	1.66E+05	1.23E+04	1.00E+03	2.16E+07	5.89E+05	4.70E+02	2.05E+05	1.33E+04	2.70E+05	2.59E+04
	Rank/Stat	12 +	7 +	8 +	10 +	4 ≈	2 -	13 +	11 +	1 -	6 ≈	3 -	9 +	5
F18	Avg error	1.08E+03	2.56E+03	3.31E+03	3.60E+02	5.04E+02	1.69E+02	1.05E+09	4.38E+02	1.20E+02	1.99E+03	3.44E+02	3.39E+03	2.97E+02
	Std	6.18E+02	2.56E+03	5.59E+03	3.68E+02	7.17E+02	4.07E+01	1.01E+09	4.47E+02	2.14E+01	4.72E+02	2.16E+02	2.34E+03	3.08E+02
	Rank/Stat	8 +	10 +	11 +	5 ≈	7 ≈	2 ≈	13 +	6 ≈	1 -	9 +	4 ≈	12 +	3
F19	Avg error	1.30E+01	2.46E+01	1.87E+01	9.68E+00	1.15E+01	1.62E+01	4.12E+02	3.00E+01	7.28E+00	2.77E+01	2.35E+01	1.59E+01	1.36E+01
	Std	4.09E-01	1.96E+00	2.64E+00	1.18E+00	9.73E-01	6.93E+00	1.29E+02	7.92E+00	1.33E+00	2.66E+01	1.59E+01	2.81E+00	1.73E+00
	Rank/Stat	4 ≈	10 +	8 +	2 -	3 -	7 ≈	13 +	12 +	1 -	11 ≈	9 +	6 +	5
F20	Avg error	3.14E+03	2.57E+02	9.40E+02	2.84E+02	5.71E+01	8.08E+03	9.67E+03	5.96E+02	5.83E+01	1.85E+03	2.02E+02	7.56E+03	6.86E+01
	Std	8.90E+02	2.27E+02	8.74E+02	2.03E+02	2.56E+01	7.05E+03	6.08E+03	1.87E+02	3.53E+01	6.99E+02	6.46E+01	4.68E+03	4.54E+01
	Rank/Stat	10 +	5 +	8 +	6 +	1 ≈	12 +	13 +	7 +	2 ≈	9 +	4 +	11 +	3
F21	Avg error	3.54E+06	8.45E+04	7.37E+04	2.13E+05	9.62E+03	1.27E+03	4.66E+06	9.15E+05	7.83E+02	2.29E+04	4.12E+03	1.79E+05	2.31E+04
	Std	8.75E+05	6.21E+04	4.09E+04	1.27E+05	6.71E+03	4.25E+02	3.17E+06	4.25E+05	2.55E+02	5.61E+03	2.05E+03	1.41E+05	1.51E+04
	Rank/Stat	12 +	8 +	7 +	10 +	4 -	2 -	13 +	11 +	1 -	5 ≈	3 -	9 +	6
F22	Avg error	9.64E+02	7.86E+02	5.42E+02	2.87E+02	5.20E+02	4.96E+02	1.87E+03	3.86E+02	5.10E+02	1.40E+03	5.79E+02	9.58E+02	4.40E+02
	Std	1.80E+02	1.65E+02	1.30E+02	1.35E+02	1.15E+02	1.16E+02	5.64E+02	1.47E+02	1.88E+02	5.50E+02	2.54E+02	3.29E+02	1.44E+02
	Rank/Stat	11 +	9 +	7 +	1 -	6 +	4 ≈	13 +	2 ≈	5 ≈	12 +	8 +	10 +	3
F23	Avg error	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	7.76E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02
	Std	1.83E-13	2.80E-12	4.26E-13	1.97E-13	1.97E-13	1.79E-13	1.07E+02	2.63E-13	2.42E-13	2.68E+01	5.20E-13	2.73E-13	1.88E-13
	Rank/Stat	9 ≈	3.5 -	3.5 -	9 ≈	9 ≈	9 ≈	13 +	3.5 ≈	3.5 +	3.5 +	12 +	3.5 ≈	9
F24	Avg error	2.63E+02	2.73E+02	2.79E+02	2.65E+02	2.68E+02	2.74E+02	4.03E+02	2.69E+02	2.75E+02	2.01E+02	2.66E+02	2.72E+02	2.74E+02
	Std	5.08E+00	5.91E+00	4.91E+00	3.38E+00	2.49E+00	2.57E+00	2.36E+01	2.31E+00	2.20E+00	1.56E-01	6.84E+00	7.20E+00	1.71E+00
	Rank/Stat	2 -	8 ≈	12 +	3 -	5 ≈	9 ≈	13 +	6 -	11 +	1 -	4 -	7 ≈	10
F25	Avg error	2.46E+02	2.03E+02	2.28E+02	2.09E+02	2.07E+02	2.24E+02	2.07E+02	2.10E+02	2.08E+02	2.00E+02	2.12E+02	2.12E+02	2.14E+02
	Std	6.02E+00	4.02E+00	8.55E+00	1.67E+00	1.16E+00	2.23E+00	1.76E+01	2.03E+00	6.38E+00	4.32E-09	5.96E+00	4.10E+00	8.86E+00
	Rank/Stat	13 +	2 -	12 +	6 -	3 -	11 +	4 -	7 -	5 -	1 -	9 ≈	8 ≈	10
F26	Avg error	1.05E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.04E+02	1.88E+02	1.74E+02	1.17E+02	2.00E+02	1.51E+02	1.00E+02	1.90E+02
	Std	2.28E+01	6.11E-02	5.71E-02	4.27E-02	2.95E-02	1.82E+01	3.21E+01	4.51E+01	3.78E+01	1.55E-02	1.08E+02	5.67E-02	3.04E+01
	Rank/Stat	7 +	2 +	4 +	1 +	3 +	6 +	11 +	10 ≈	8 -	13 +	9 ≈	5 -	12
F27	Avg error	3.47E+02	1.57E+03	1.20E+03	3.47E+02	4.01E+02	4.55E+02	2.00E+03	3.72E+02	4.87E+02	3.16E+02	1.23E+03	1.19E+03	4.11E+02
	Std	1.66E+01	5.94E+01	3.24E+02	2.36E+01	4.87E+01	6.03E+01	1.27E+02	4.03E+01	4.76E+01	2.31E+01	1.11E+02	1.26E+02	4.04E+01
	Rank/Stat	3 -	12 +	10 +	2 -	5 ≈	7 +	13 +	4 -	8 +	1 -	11 +	9 +	6
F28	Avg error	1.09E+03	3.90E+02	4.15E+02	1.06E+03	1.13E+03	1.13E+03	7.18E+03	1.11E+03	1.16E+03	1.51E+03	2.23E+03	1.74E+03	1.19E+03
	Std	1.02E+02	9.80E+00	2.57E+01	3.71E+01	5.82E+01	4.86E+01	1.07E+03	5.42E+01	6.93E+01	9.33E+02	5.58E+02	3.82E+02	3.92E+01
	Rank/Stat	4 -	1 -	2 -	3 -	6 -	7 -	13 +	5 -	8 -	10 ≈	12 +	11 +	9
F29	Avg error	2.38E+04	2.24E+02	2.19E+02	1.98E+03	1.00E+03	8.84E+02	3.46E+08	1.73E+03	8.04E+02	1.20E+03	1.41E+07	8.22E+06	1.45E+03
	Std	1.24E+04	4.16E+00	3.69E+00	3.64E+02	1.65E+02	8.34E+01	1.28E+08	3.45E+02	4.13E+01	3.77E+02	1.76E+07	1.52E+07	2.57E+02
	Rank/Stat	10 +	2 -	1 -	9 +	5 -	4 -	13 +	8 +	3 -	6 -	12 ≈	11 +	7
F30	Avg error	8.40E+03	1.03E+03	6.91E+02	8.30E+03	8.64E+03	1.00E+04	3.32E+06	9.26E+03	9.66E+03	5.44E+04	1.04E+04	9.83E+03	9.03E+03
	Std	4.77E+02	1.55E+02	1.99E+02	3.74E+02	5.07E+02	8.87E+02	2.24E+06	4.39E+02	8.00E+02	9.47E+03	1.91E+03	1.75E+03	6.58E+02
	Rank/Stat	4 -	2 -	1 -	3 -	5 -	10 +	13 +	7 ≈	8 +	12 +	11 +	9 +	6
	#(+)	21	19	25	17	11	15	29	18	12	15	18	26	
	#(-)	6	7	4	10	9	10	1	6	11	8	5	1	
	#(≈)	3	4	1	3	10	5	0	6	7	7	7	3	

Table S4: Comparison of solution accuracy on 50-dimensional CEC 2017 problems

Problem	Result	ODE	EPSDE	SaDE	SinDE	jDE	JADE	NBOLDE	TVDE	MPEDE	MLGSA	RSDE	DLABC	MIDE
F1	Avg error	4.22E+03	2.78E-08	1.98E+03	4.73E+03	6.20E-08	5.75E-13	5.44E+10	3.93E+03	2.61E-14	6.12E+03	2.04E+03	1.21E+04	8.51E+01
	Std	5.15E+03	9.36E-08	2.30E+03	5.05E+03	1.64E-07	2.70E-12	1.06E+10	3.83E+03	5.39E-15	6.35E+03	3.51E+03	1.05E+04	2.11E+02
	Rank/Stat	9 +	3 -	6 +	10 +	4 -	2 -	13 +	8 +	1 -	11 +	7 +	12 +	5
F2	Avg error	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	0.00E+00	0.00E+00	0.00E+00
	Std	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	0.00E+00	0.00E+00	0.00E+00
	Rank/Stat	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7 ≈	7
F3	Avg error	7.04E+04	4.49E+04	2.92E+02	1.61E+04	1.26E+02	1.48E+04	1.78E+03	1.33E+04	1.17E-04	6.51E+03	1.65E+04	2.19E+05	1.74E+02
	Std	8.53E+03	3.86E+04	3.87E+02	3.87E+03	1.52E+02	3.40E+04	1.57E+03	3.67E+03	4.57E-04	2.04E+03	3.50E+03	2.14E+04	2.22E+02
	Rank/Stat	12 +	11 +	4 ≈	9 +	2 ≈	8 ≈	5 +	7 +	1 -	6 +	10 +	13 +	3
F4	Avg error	5.17E+01	4.42E+01	8.21E+01	6.36E+01	6.61E+01	5.01E+01	1.08E+04	7.83E+01	1.60E+02	1.34E+02	7.44E+01	1.05E+02	6.14E+01
	Std	4.03E+01	4.93E+01	5.13E+01	5.12E+01	4.35E+01	5.04E+01	3.46E+03	4.52E+01	4.36E+01	2.97E+01	4.20E+01	5.18E+01	4.02E+01
	Rank/Stat	3 ≈	1 ≈	9 ≈	5 ≈	6 ≈	2 ≈	13 +	8 +	12 ≈	11 +	7 ≈	10 +	4
F5	Avg error	3.59E+02	1.51E+02	9.20E+01	6.75E+01	9.32E+01	5.46E+01	4.46E+02	1.62E+02	5.96E+01	1.03E+02	1.81E+02	1.75E+02	5.18E+01
	Std	1.37E+01	1.62E+01	1.35E+01	1.47E+01	1.20E+01	6.45E+00	3.37E+01	2.34E+01	1.16E+01	5.69E+01	3.95E+01	3.36E+01	7.16E+00
	Rank/Stat	12 +	8 +	5 +	4 +	6 +	2 ≈	13 +	9 +	3 +	7 +	11 +	10 +	1
F6	Avg error	2.16E-13	1.52E-13	3.34E-03	1.59E-13	1.21E-13	1.52E-13	5.78E+01	1.40E-07	9.49E-04	1.39E-01	6.94E-02	6.05E-04	1.55E-13
	Std	3.47E-14	5.45E-14	1.08E-02	5.66E-14	2.88E-14	9.12E-14	5.42E+00	3.32E-07	4.45E-03	1.62E-01	3.84E-02	2.44E-03	5.57E-14
	Rank/Stat	6 +	2.5 ≈	10 +	5 ≈	1 -	2.5 ≈	13 +	7 +	9 +	12 +	11 +	8 +	4
F7	Avg error	4.20E+02	2.02E+02	1.47E+02	1.17E+02	1.49E+02	1.00E+02	1.10E+03	2.38E+02	1.03E+02	6.78E+01	3.08E+02	2.20E+02	9.68E+01
	Std	1.48E+01	1.64E+01	2.12E+01	1.27E+01	1.05E+01	8.16E+00	2.30E+02	2.27E+01	1.21E+01	2.85E+00	4.45E+01	5.13E+01	5.30E+00
	Rank/Stat	12 +	8 +	6 +	5 +	7 +	3 +	13 +	10 +	4 +	1 -	11 +	9 +	2
F8	Avg error	3.56E+02	1.53E+02	9.65E+01	6.40E+01	8.94E+01	5.46E+01	4.52E+02	1.61E+02	1.94E+02	9.55E+01	1.83E+02	1.57E+02	5.32E+01
	Std	2.00E+01	1.68E+01	1.53E+01	1.11E+01	9.50E+00	8.63E+00	4.38E+01	2.39E+01	1.23E+01	5.39E+01	3.92E+01	3.18E+01	6.38E+00
	Rank/Stat	12 +	7 +	6 +	3 +	4 +	2 ≈	13 +	9 +	11 ≈	5 +	10 +	8 +	1
F9	Avg error	1.14E-13	1.97E+00	5.66E+01	1.14E-13	1.36E-01	1.53E+00	1.18E+04	1.51E-02	9.57E-01	1.19E-02	2.55E+03	6.40E+02	3.33E-02
	Std	0.00E+00	9.59E+00	8.53E+01	0.00E+00	3.34E-01	1.62E+00	1.91E+03	8.29E-02	8.03E-01	3.10E-02	1.35E+03	5.49E+02	1.16E-01
	Rank/Stat	1.5 ≈	9 +	10 +	1.5 ≈	6 +	8 +	13 +	4 ≈	7 +	3 ≈	12 +	11 +	5
F10	Avg error	1.25E+04	7.36E+03	6.72E+03	4.37E+03	5.02E+03	3.84E+03	9.40E+03	8.95E+03	4.97E+03	5.42E+03	5.89E+03	5.77E+03	3.20E+03
	Std	4.78E+02	7.70E+02	1.41E+03	6.57E+02	2.68E+02	2.56E+02	1.15E+03	7.58E+02	6.64E+02	8.13E+02	8.99E+02	1.20E+03	3.24E+02
	Rank/Stat	13 +	10 +	9 +	3 +	5 +	2 +	12 +	11 +	4 +	6 +	8 +	7 +	1
F11	Avg error	1.69E+02	6.21E+01	9.73E+01	3.74E+01	4.97E+01	1.20E+02	4.20E+03	5.02E+01	9.98E+01	1.24E+02	2.59E+02	3.57E+02	4.05E+01
	Std	1.07E+01	2.12E+01	3.01E+01	4.42E+00	9.13E+00	3.35E+01	2.39E+03	1.00E+01	1.98E+01	9.33E+00	7.04E+01	6.71E+01	4.79E+00
	Rank/Stat	10 +	5 +	6 +	1 -	3 +	8 +	13 +	4 +	7 +	9 +	11 +	12 +	2
F12	Avg error	4.03E+07	2.70E+04	1.03E+05	1.25E+06	6.43E+04	6.34E+03	1.32E+10	1.01E+06	1.14E+04	2.28E+05	7.04E+04	5.08E+05	7.36E+04
	Std	1.11E+07	4.15E+04	4.88E+04	7.27E+05	3.96E+04	4.66E+03	4.54E+09	4.97E+05	9.15E+03	8.36E+04	5.31E+04	3.35E+05	5.29E+04
	Rank/Stat	12 +	3 -	7 +	11 +	4 ≈	1 -	13 +	10 +	2 -	8 +	5 ≈	9 +	6
F13	Avg error	1.16E+04	2.60E+03	2.19E+03	3.33E+03	1.91E+03	3.16E+02	2.80E+09	2.50E+03	9.31E+01	4.02E+02	5.18E+02	1.26E+04	1.50E+03
	Std	9.58E+03	3.90E+03	2.79E+03	3.66E+03	1.56E+03	2.08E+02	2.45E+09	3.19E+03	5.57E+01	3.53E+02	4.15E+02	1.36E+04	1.85E+03
	Rank/Stat	11 +	9 ≈	7 ≈	10 +	6 ≈	2 -	13 +	8 ≈	1 -	3 -	4 -	12 +	5
F14	Avg error	3.68E+04	1.00E+03	2.68E+03	8.70E+03	6.07E+01	7.17E+03	1.90E+06	3.64E+04	5.65E+01	1.09E+04	3.01E+02	4.84E+04	7.75E+02
	Std	1.50E+04	1.30E+03	3.66E+03	9.03E+03	2.39E+01	3.81E+04	4.14E+06	2.39E+04	1.07E+01	2.70E+03	1.01E+02	4.52E+04	7.99E+02
	Rank/Stat	11 +	5 ≈	6 +	8 +	2 -	7 +	13 +	10 +	1 -	9 +	3 -	12 +	4
F15	Avg error	3.26E+03	4.42E+02	2.59E+03	1.84E+03	7.64E+01	2.47E+02	6.29E+07	2.94E+03	6.91E+01	1.01E+04	2.04E+02	9.99E+03	3.15E+02
	Std	1.35E+03	8.07E+02	1.96E+03	2.86E+03	8.91E+01	1.31E+02	8.50E+07	3.32E+03	2.60E+01	1.93E+03	9.09E+01	1.04E+04	5.92E+02
	Rank/Stat	10 +	6 ≈	8 +	7 +	2 -	4 ≈	13 +	9 +	1 -	12 +	3 ≈	11 +	5
F16	Avg error	2.77E+03	8.98E+02	8.27E+02	6.76E+02	8.95E+02	8.56E+02	2.70E+03	7.63E+02	1.04E+03	1.57E+03	1.01E+03	1.61E+03	7.55E+02
	Std	1.68E+02	2.29E+02	1.83E+02	2.05E+02	2.08E+02	1.88E+02	4.33E+02	2.73E+02	2.69E+02	3.73E+02	3.56E+02	3.81E+02	1.56E+02
	Rank/Stat	13 +	7 +	4 ≈	1 ≈	6 +	5 ≈	12 +	3 ≈	9 +	10 +	8 +	11 +	2
F17	Avg error	1.36E+03	6.71E+02	5.08E+02	3.73E+02	7.42E+02	5.81E+02	1.92E+03	5.64E+02	5.30E+02	1.13E+03	6.95E+02	1.33E+03	5.43E+02
	Std	1.69E+02	1.49E+02	1.85E+02	1.58E+02	1.15E+02	1.59E+02	3.59E+02	2.11E+02	1.97E+02	4.38E+02	2.25E+02	3.41E+02	1.37E+02
	Rank/Stat	12 +	7 +	2 ≈	1 -	9 +	6 ≈	13 +	5 ≈	3 ≈	10 +	8 +	11 +	4
F18	Avg error	3.37E+06	1.38E+05	4.68E+04	2.64E+05	2.49E+03	5.77E+03	5.71E+06	7.72E+05	1.11E+02	2.31E+06	4.80E+03	2.27E+05	1.52E+04
	Std	1.05E+06	2.94E+05	4.02E+04	2.14E+05	2.56E+03	3.01E+04	6.42E+06	5.94E+05	8.54E+01	1.62E+06	4.99E+03	2.41E+05	1.16E+04
	Rank/Stat	12 +	7 ≈	6 +	9 +	2 -	4 -	13 +	10 +	1 -	11 +	3 -	8 +	5
F19	Avg error	5.06E+03	6.61E+01	1.24E+04	5.40E+03	2.67E+01	1.31E+02	2.45E+07	1.14E+04	3.76E+01	1.57E+04	1.06E+02	1.32E+04	6.18E+01
	Std	3.80E+03	4.01E+01	3.65E+03	4.44E+03	7.43E+00	4.60E+01	3.77E+07	5.82E+03	1.32E+01	2.91E+03	3.93E+01	1.17E+04	3.00E+01
	Rank/Stat	7 +	4 ≈	10 +	8 +	1 -	6 +	13 +	9 +	2 -	12 +	5 +	11 +	3
F20	Avg error	1.20E+03	4.91E+02	3.20E+02	2.11E+02	4.86E+02	4.47E+02	1.23E+03	3.77E+02	4.54E+02	6.07E+02	4.27E+02	8.59E+02	3.19E+02
	Std	1.43E+02	1.18E+02	1.51E+02	1.44E+02	9.58E+01	1.26E+02	4.84E+02	1.82E+02	2.37E+02	1.73E+02	2.14E+02	3.51E+02	1.21E+02
	Rank/Stat	12 +	9 +	3 ≈	1 -	8 +	6 +	13 +	4 ≈	7 +	10 +	5 +	11 +	2
F21	Avg error	5.59E+02	3.63E+02	2.80E+02	2.62E+02	2.98E+02	2.53E+02	6.78E+02	3.58E+02	2.51E+02	3.60E+02	3.95E+02	3.71E+02	2.55E+02
	Std	1.51E+01	1.80E+01	1.77E+01	1.25E+01	1.14E+01	8.54E+00	4.61E+01	2.79E+01	1.63E+01	3.51E+01	3.96E+01	3.79E+01	7.46E+00
	Rank/Stat	12 +	9 +	5 +	4 +	6 +	2 ≈	13 +	7 +	1 ≈	8 +	11 +	10 +	3
F22	Avg error	1.20E+04	6.57E+03	3.44E+03	4.13E+03	4.33E+03	3.17E+03	8.86E+03	7.99E+03	3.33E+03	1.00E+02	6.74E+03	5.94E+03	1.87E+03
	Std	3.25E+03	3.11E+03	3.03E+03	1.27E+03	2.40E+03	2.02E+03	1.56E+03	2.76E+03	2.75E+03	7.58E-01	9.61E+02	1.03E+03	1.93E+03
	Rank/Stat	13 +	9 +	5 +	6 +	7 +	3 +	12 +	11 +	4 +	1 -	10 +	8 +	2
F23	Avg error	7.78E+02	5.90E+02	5.13E+02	4.80E+02	5.10E+02	4.79E+02	1.31E+03	5.64E+02	4.82E+02	6.06E+02	6.86E+02	6.13E+02	4.92E+02
	Std	1.48E+01	2.20E+01	2.22E+01	1.11E+01	1.15E+01	1.15E+01	1.16E+02	1.97E+01	1.51E+01	5.01E+01	7.22E+01	4.87E+01	1.26E+01
	Rank/Stat	12 +	8 +	6 +	2 -	5 +	1 -	13 +	7 +	3 -	9 +	11 +	10 +	4
F24	Avg error	8.43E+02	6.55E+02	5.90E+02	5.48E+02	5.76E+02	5.39E+02	1.52E+03	6.49E+02	5.43E+02	6.85E+02	7.73E+02	6.50E+02	5.67E+02
	Std	9.98E+00	2.38E+01	1.72E+01	8.89E+00	1.13E+01	1.00E+01	1.30E+02	3.11E+01	1.46E+01	1.26E+02	8.42E+01	2.91E+01	1.28E+01
	Rank/Stat	12 +	9 +	6 +	3 -	5 +	1 -	13 +	7 +	2 ≈	10 +	11 +	8 +	4
F25	Avg error	4.86E+02	5.48E+02	5.71E+02	4.90E+02	5.15E+02	5.20E+02	5.92E+03	5.17E+02	5.14E+02	6.08E+02	5.37E+02	5.23E+02	5.22E+02
	Std	2.09E+01	3.95E+01	2.81E+01	2.61E+01	3.61E+01	4.07E+01	1.39E+03	3.50E+01	3.43E+01	1.41E+01	3.62E+01	3.93E+01	3.65E+01
	Rank/Stat	1 -	10 +	11 +	2 -	4 ≈	6 ≈	13 +	5 ≈	3 ≈	12 +	9 ≈	8 ≈	7
F26	Avg error	4.49E+03	2.38E+03	2.62E+03	1.68E+03	1.92E+03	1.63E+03	1.07E+04	2.18E+03	1.64E+03	3.00E+02	3.95E+03	3.12E+03	1.71E+03
	Std	1.39E+02	1.87E+02	2.47E+02	1.23E+02	6.93E+01	1.11E+02	1.05E+03	2.38E+02	1.81E+02	3.41E-08	6.68E+02	4.60E+02	1.01E+02
	Rank/Stat	12 +	8 +	9 +	4 ≈	6 +	2 -	13 +	7 +	3 ≈	1 -	11 +	10 +	5
F27	Avg error	5.08E+02	5.71E+02	7.00E+02	5.08E+02	5.33E+02	5.67E+02	1.94E+03	5.29E+02	5.48E+02	8.71E+02	7.42E+02	7.00E+02	5.56E+02
	Std	9.60E+00	1.45E+01	4.18E+01	9.95E+00	1.74E+01	3.14E+01	2.95E+02	1.09E+01	2.59E+01	6.48E+01	8.65E+01	8.78E+01	2.33E+01
	Rank/Stat	1 -	8 +	10 +	2 -	4 -	7 ≈	13 +	3 -	5 ≈	12 +	11 +	9 +	6
F28	Avg error	4.62E+02	4.95E+02	5.06E+02	4.64E+02	4.90E+02	4.94E+02	5.21E+03	4.85E+02	4.88E+02	5.21E+02	4.99E+02	4.92E+02	4.90E+02
	Std	1.14E+01	3.22E+01	2.14E+01	1.49E+01	2.16E+01	1.70E+01	1.03E+03	2.48E+01	2.26E+01	3.57E+01	2.11E+01	2.08E+01	2.39E+01
	Rank/Stat	1 -	9 ≈	11 +	2 -	6 ≈	8 ≈	13 +	3 ≈	4 -	12 +	10 ≈	7 ≈	5
F29	Avg error	1.45E+03	5.43E+02	4.75E+02	3.67E+02	5.07E+02	4.77E+02	5.08E+03	5.02E+02	4.87E+02	1.27E+03	7.59E+02	1.06E+03	4.26E+02
	Std	1.61E+02	8.90E+01	9.95E+01	5.26E+01	6.82E+01	5.54E+01	1.19E+03	7.57E+01	1.08E+02	2.60E+02	2.42E+02	2.20E+02	6.76E+01
	Rank/Stat	12 +	8 +	3 +	1 -	7 +	4 +	13 +	6 +	5 ≈	11 +	9 +	10 +	2
F30	Avg error	3.49E+06	7.41E+05	8.07E+05	7.23E+05	5.95E+05	6.51E+05	3.98E+08	6.96E+05	6.86E+05	2.87E+06	6.24E+05	9.81E+05	6.11E+05
	Std	9.85E+05	9.10E+04	8.59E+04	1.09E+05	2.67E+04	6.52E+04	1.66E+08	8.95E+04	1.02E+05	5.84E+05	5.84E+04	2.39E+05	3.65E+04
	Rank/Stat	12 +	8 +	9 +	7 +	1 -	4 +	13 +	6 +	5 +	11 +	3 ≈	10 +	2
	#(+)	24	19	23	15	15	10	29	21	10	24	20	27	NA
	#(-)	3	2	0	9	8	7	0	1	10	4	3	0	NA
	#(≈)	3	9	7	6	7	13	1	8	10	2	7	3	NA