Differential Evolution With Domain Transform

Sheng Xin Zhang, Yi Nan Wen, Yu Hong Liu, Li Ming Zheng, and Shao Yong Zheng, *Senior Member, IEEE*

***Abstract*—Although a significant advancement of differential evolution (DE) for global optimization has been witnessed in the past two decades, the problems of premature convergence and stagnation are still open questions that hinder the performance. Both are likely to occur on complicated multimodal functions, but the phenomena differ. Premature convergence refers to a rapid loss of population diversity when attracted to a local minimum while stagnation happens even though the population is diverse. To deal with these problems, this article proposes a domain transform (DT) methodology. Different from existing fitness analysis which mainly utilizes the original fitness landscape information, DT yields a transformed fitness landscape with transform operation to the frequency domain and inverse transform operation back to the solution domain, between which the first few highest frequencies are removed. With the deletion operation, the transformed fitness landscape becomes smoother and facilitates the escape from local minima and stagnation on complicated multimodal functions. Simulation results show that DT significantly improves the population successful update rate and population convergence. The constructed DTDE algorithm consequently exhibits remarkable improvements on the baseline algorithm and outperforms several state-of-the-art DE variants. DT has also been extended for noisy optimization and it performs better than the baseline, the classic resampling method, state-of-the-art DE variants, as well as several popular noisy evolutionary optimization algorithms.**

***Index Terms*—Differential evolution (DE), domain transform (DT), evolutionary optimization, fitness landscape, noisy optimization.**

I. Introduction

EVOLUTIONARY algorithms (EAs) [1] have been demonstrated as a kind of effective approach for solving complicated optimization problems. With a distributed population structure, EAs are more robust to local minima when compared with traditional single-point searching methods, such as the gradient descent [2]. In the past two decades, many research interests have been devoted to EAs and fruitful advancements have been achieved. Among them, differential evolution (DE) [3], proposed by Price and Storn, has been one of the most popular EAs. In the DE literature, a clear step-by-step development from the classic to the state-of-the-art could be observed [4], [5], [6] and DE has been applied in various scientific and engineering fields, such as machine scheduling [7], protein structure prediction [8], combinatorial [9], and constrained multiobjective optimization [10].

As for black-box optimization, it is crucial to collect the evolutionary information to guide the search process. The fitness landscape which depicts a glance of the problem structure, provides essential observations and evidence for the adjustment of search strategies. Methods for utilizing fitness landscape information could be implicit or explicit.

Implicit methods usually take advantage of fitness comparison or fitness ranking. For DE, on the one hand, the one-to-one fitness comparison of parent and offspring in the DE selection has been widely adopted to develop strategy adaptation, such as memory [11], probability [12], multipopulation [13], and stage-based [14] adaptation mechanisms. It has also been employed for adjusting control parameters, such as the individual-encoded [15], distribution-dependent [16], and history-based [17] parameter control methods. On the other hand, the observation from fitness ranking has been considered to allocate strategy or parameters. For instance, the individual-dependent mutation [18], individual-dependent topology [19], fitness-based similarity selection [20], and individual-dependent parameters [18].

Explicit methods first analyze the features of the fitness landscape and then decide the search strategy based on the obtained features. Several fitness landscape metrics have been proposed, such as the fitness distance correlation (FDC) [21], dispersion (DISP) [22], population evolvability [23], and new measures with the assist of frequency analysis [24]. Some of them have been utilized to determine search strategies. For instance, the selection of CMA-ES variants [22], distribution strategies [25], DE strategies [26], [27], [28], [29], and different EAs [23].

Although the performance of DE has been significantly advanced with the explicit and implicit utilizations of fitness information, the problems of premature convergence [30] and stagnation [31], [32], [33] are still challenging for DE. Premature convergence refers to the attraction of a population to a local minimum, which usually happens when the information of fit solutions is overused. While when stagnation occurs, the population is still diverse, but the solutions stop generating successful offspring [31]. As pointed by Lampinen and Zelinka [31], DE suffers from stagnation due to the limited number of trial vectors.

Manuscript received 26 April 2022; revised 25 July 2022 and 28 September 2022; accepted 22 October 2022. Date of publication 8 November 2022; date of current version 3 October 2023. This work was supported in part by the Applied Science and Technology Research and Development Special Fund Project of Guangdong Province under Grant 2016B010126004; in part by the National Special Project Number for International Cooperation under Grant 2015DFR11050; and in part by the National Natural Science Foundation of China under Grant 62201227 and Grant 62071503. *(Corresponding authors: Sheng Xin Zhang; Shao Yong Zheng.)*

Sheng Xin Zhang, Yu Hong Liu, and Li Ming Zheng are with the College of Information Science and Technology and the College of Cyber Security, Jinan University, Guangzhou 510632, China (e-mail: zhangsx@jnu.edu.cn).

Yi Nan Wen is with the College of Information Science and Technology, Jinan University, Guangzhou 510632, China.

Shao Yong Zheng is with the School of Electronics and Information Technology, Guangdong Provincial Key Laboratory of Optoelectronic Information Processing Chips and Systems, Sun Yat-sen University, Guangzhou 510006, China (e-mail: zhengshaoy@mail.sysu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TEVC.2022.3220424.

Digital Object Identifier 10.1109/TEVC.2022.3220424

To deal with the issues of premature convergence and stagnation, this article proposes the domain transform (DT)-based analysis of the fitness landscape to guide the evolution of DE. At each generation, the original fitness landscape which consists of parent and offspring solutions are first transformed from the solution domain to the frequency domain to separate the components of the original landscape. Subsequently, the top highest frequencies are removed in the frequency domain, followed by an inverse transform operation which transforms the remaining frequencies from the frequency domain back to the solution domain. A transformed fitness landscape is consequently generated by the DT method.

DT has been incorporated with a competitive baseline, i.e., the SCSS-L-SHADE [20] algorithm. With the transformed fitness, the DE population yields higher population successful update rate and higher population convergence, which helps escape from the situation of stagnation and premature convergence on complicated multimodal functions. Further, to take advantages of both the transformed and original fitness landscapes on multimodal and unimodal functions, respectively, an evolution difficulty-based utilization mechanism is adopted. The effectiveness of the proposed methods is confirmed by experiments performed for solving traditional noiseless as well as noisy optimizations [34].

The remainder of this article is organized as follows. Section II presents the background, including the advancement of DE for noiseless optimization, the related works on noisy optimization, and fitness landscape analysis. Section III introduces the theory of discrete Fourier transform (DFT). Section IV describes the proposed method while Section V validates its effectiveness by simulation comparisons. Finally, Section VI concludes this work and makes a prospect for future research.

II. Background

A. Traditional Noiseless Evolutionary Optimization With DE

Due to its simple structure and competitive performance, DE[1] has attracted substantial attention from researchers. In the past two decades, the performance of DE has been consistently improved with contributions mainly attributed to the enhancements of strategy [35] and parameter [36] controls.

Since DE is featured by the differential mutation and there are plenty of mutation strategies available for solving distinct types of problems, researchers have paid attention to strategy adjustments for performance improvements. On the one hand, new mutation strategies have been constructed, for instance, the "current-to-*p*best/1" mutation [16], the collective information-based mutation [37], and the level-based mutation [38]. On the other hand, to utilize strategies, various methods have been developed, including strategy combination, strategy adaptation, and multistrategy utilization. For example, the global and local mutation-based DE (DEGL) [39] combines global mutation with local mutation using a weighting factor. Strategy adaptive DE (SaDE) [11], ensemble of parameters and strategies DE (EPSDE) [40], multipopulation-based ensemble of strategies DE (MPEDE) [13], DE with adaptive mutation (ZEPDE) [41], and explicitly adaptive DE (EaDE) [42] adapt strategies according to distinct mechanisms. Except for the strategy adaptation, research on employing multiple strategies to simultaneously generate multiple candidates and filtering one as offspring has also been conducted, such as the composite DE (CoDE) [43], cheap surrogate model-based DE (CSM-DE) [44], underestimation-based DE (UMDE) [45], and selective candidate with similarity selection rule-based DE (SCSS-DE) [20].

The performance of DE also heavily depends on the control parameters, namely, the scaling factor F, the crossover rate CR, and the population size NP. To adapt these parameters for different problems, deterministic and adaptive parameter control schemes have been developed. Deterministic methods adjust the parameters based on progress-dependent values, such as the linear population size reduction in L-SHADE [46] and the weighting-based F setting in jSO [47], or observations, such as the fitness rankings in IDE [18]. While adaptive methods allocate parameters based on successful experience, such as the adaptation of F and CR in jDE [15], JADE [16], CoBiDE [48], SHADE [17], L-SHADE_cnEpSin [49], PaDE [50], and the adaptation of NP in ADDE [51].

B. Noisy Evolutionary Optimization

Noise imposes difficulties to the selection operation of EAs [52], [53]. For demonstration, two solutions A and B with real fitness of $f_A < f_B$ are considered. With the injected noise η_A and η_B, respectively, the final noisy fitness might be $f_A^{\text{noisy}} > f_B^{\text{noisy}}$ and hard to distinguish the real fitness of A and B. It consequently influences the fitness ranking of the population or the fitness comparison between parent and offspring and as a result, superior solutions might be declined while inferior solutions might mistakenly pass into the next generation. To resolve this problem and improve the performance of noisy EAs (NEAs), researchers have put forward plenty of techniques. Rakshit et al. [34] presented a comprehensive survey on NEAs, where the noise handling techniques were divided into five categories: 1) sampling strategy-based explicit averaging; 2) fitness estimation from multiple samplings; 3) dynamic population sizing-based implicit averaging; 4) enhanced evolutionary strategies; and 5) improved robust selection.

First, since the reevaluation of a same solution for n times could reduce the standard error of the mean fitness by a factor of $\sqrt{n}$ [34], sampling [54], [55] has been widely investigated in the literature. The relevant open questions include: 1) how many samplings should be performed; 2) when to perform the samplings; and 3) samplings perform on which solutions. To handle these issues, sample size settings according to time-related variables [56] such as generation number or function evaluations, noise strength [57], fitness variance [58], and reinforcement learning-based autonomous selection [59], have been developed. Noise analysis deciding whether to sample [60] and dynamic resampling [57] deciding to sample on which solutions have also been introduced.

Second, to overcome the weakness in traditional resampling strategies in which all the sampled fitness are usually

[1] Due to the space limitation, a brief introduction of the classic DE is presented in the supplementary file.

assumed to have equal occurrence probability, improved fitness estimation from multiple samplings using uniform fitness interval [61], nonuniform fitness interval [62], and type-2 fuzzy set [62] have been proposed.

Third, the contamination effect incurred by noise can be compensated by increasing the population size [63], which takes advantages of the inherent characteristics of population-based metaheuristics to frequently sample the promising area in the search space and therefore to produce a large number of similar trial solutions. However, similar to the sampling strategy, the population sizing strategy also increases the computational complexity since more solutions have to be evaluated at each generation.

Fourth, how to exploit the local area of a solution and how to explore the search space without being trapped into premature convergence are also major challenges for noisy optimization, which has been investigated with enhanced evolutionary strategies, such as opposition-based learning [64], adaptive mutation strength [65], data mining-based crossover [66], chaotic jump [67], local model- [68] and memetic-based [69] searches, and weighted search center-based learning [70].

Finally, the judicious selection of solutions for resampling and better solutions for the next generation can offset the adverse effect caused by noise. Quite a lot of robust selection strategies have been developed, such as Kalman formulation-assisted reevaluation [71], rolling tide selection-based resampling [72], and restricted Boltzmann machine-based likelihood correction [73].

C. Fitness Landscape Analysis

The fitness landscape, which consists of the fitness values of individuals, is the basic description of a problem. Through the analysis of the fitness landscape, useful information on optimization problems can be obtained. Due to this benefit, the fitness landscape has been widely investigated and applied to predict appropriate strategies or optimizers to suit the optimization of different problems. Representative methods include FDC [21], evolvability [74], dispersion (DISP) [22], ruggedness of information entropy (RIE) [75], exploratory landscape analysis (ELA) [76], and local optima networks (LONs) [77].

FDC [21], a method for quantifying the relationship between fitness and distance was designed by Jones and Forrest to measure the difficulty of optimization problems. This indicator is reliable in many cases even with a few samples, but it has counterexamples. For instance, the "GA-easy" fitness function which shows no relationship between fitness and Hamming distance from the global optimum [21]. The initial FDC indicator cannot be used to predict unknown problems because it needs the knowledge of the global optimum. While this weakness has been overcome in the later developments and FDC has been employed to assist the selection of appropriate DE strategies [29]. Evolvability [78], a measurement of parent solutions' ability to produce fitter offspring is effective in problems in which FDC is invalid. But it also has counterexamples, such as the long path problem [78]. In [74], the features of fitness landscape were revealed by the average evolvability over the offspring of individuals sampled randomly or online. While this metric may not be effective in search space with heterogeneous anisotropic nature to make a prediction of global space properties. Lunacek and Whitley proposed DISP [22], which is obtained by calculating the average pairwise distance in solution space between all points in a sample, to estimate the presence of funnels. DISP was adopted to choose suitable CMA-ES variants for solving different types of problems [22]. Malan and Engelbrecht [75] adopted entropic measures for continuous problems to measure the ruggedness, which refers to the number and distribution of local optima. ELA [76], [79], proposed by Mersmann et al., includes several methods like expected running time (ERT) and multidimensional scaling (MDS), which can be employed to obtain the macro-properties of an unknown problem. They later suggested some low-level features in [80] to estimate the properties relatively cheaply. LONs [77], [81] were modified to obtain a continuous fitness landscape, of which nodes and edges were achieved by recording and aggregating the local minima visited by several trajectories of the Basin-Hopping algorithm. It can make the global structure of high-dimensional continuous functions be visualized while it is not computationally efficient.

III. Theory of Discrete Fourier Transform

A. Discrete Fourier Transform

Fourier series (FS) [82] was used to analyze continuous periodic functions, which has also been further extended to the analysis of continuous aperiodic functions on the assumption that the period is infinite. The continuous Fourier transform (CFT) is an integral transformation, with which a complicated signal is decomposed into the superposition of several sine functions. In this way, the analysis of the original time series can be transformed to that of the amplitudes and phases of various harmonic components at different frequencies. The calculation of CFT can be written as follows:

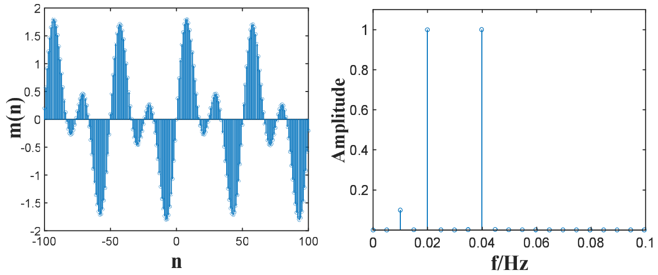
$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \tag{1}$$

where $f(t)$ denotes the continuous aperiodic function of time, ω represents the angular frequency of sine function, and $j = \sqrt{-1}$. The inverse operation of CFT can be obtained by

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)e^{j\omega t}d\omega. \tag{2}$$

The frequency distribution, amplitudes, and phases of various frequencies can be obtained by CFT, which is an easy approach for analyzing and processing the time series. However, although it is applicable to utilize (1) to calculate the CFT of closed-form functions, it is not the case in industrial applications since only discrete data can be processed by computers. To this end, the DFT was developed, which reflects the mapping relationship between the discrete time and discrete frequency. The DFT of a uniformly sampled and finite sequence $m(n)$ with N sampling points is defined as follows:

$$M(k) = \sum_{n=0}^{N-1} m(n)W_N^{nk}, k = 0, 1, \ldots, N-1 \tag{3}$$



(a) (b)

Fig. 1. Time series $m(n)$ and its DFT $M(k)$. (a) Time domain. (b) Frequency domain.

where $M(k)$ denotes the output data obtained by DFT and W_N^{nk} is equal to $(W_N)^{nk}$. W_N, the Nth root of unity, is written as

$$W_N = e^{-\frac{j2\pi}{N}}. \tag{4}$$

From (3), we can see that N multiplications and $N-1$ additions are required to obtain an $M(k)$ and obviously the number of multiplications and additions for calculating all $M(k)$ is proportional to N^2, which is of high computational complexity along with the increase of N. To reduce both computational efforts and round-off mistakes of the DFT substantially, fast Fourier transform (FFT) was developed by Cooley and Tukey in 1965 [83], which takes advantages of the time-series decomposition and some inherent properties, such as symmetry, periodicity, and reducibility of coefficient W_N^{nk}.

B. Inverse Discrete Fourier Transform

DFT transforms the signal from the time domain to the frequency domain, while it is necessary to convert it back to the time domain when the signal processing is finished. The definition of inverse DFT (IDFT) is as follows:

$$m(n) = \frac{1}{N}\sum_{k=0}^{N-1} M(k)W_N^{-nk}, n = 0, 1, \ldots, N-1 \tag{5}$$

where $W_N^{-nk} = (W_N)^{-nk}$ and W_N is the same as (4).

There exist a lot of problems difficult to obtain feasible solutions straightforwardly, while they become achievable with the use of DT. Such kind of problems can be solved by transforming the original problem to a transform-domain problem and converting them back to the original domain via a reciprocal operation after finding appropriate solutions. For demonstration, a finite time series $m(n) = \sin(n/4) + \sin(n/8) + 0.1 \times \sin(n/16)$ in time and frequency domains are shown in Fig. 1, where n is an integer ranging from -100 to 100 with the sampling interval set to be 1. Owing to the fact that the time series $m(n)$ is obtained by linear superposition of three basic sine functions, $\sin(n/4)$, $\sin(n/8)$, and $\sin(n/16)$, each of which has a single frequency, that is, $1/(8\pi)$, $1/(16\pi)$, and $1/(32\pi)$, respectively, only three frequency values are required to depict the time series $m(n)$ in the frequency domain, as shown in Fig. 1(b) and the amplitude is 1, 1, and 0.1, respectively. Therefore, the main components of the original signal, which are not easily distinguishable in the time domain could be clearly observed in the frequency domain. Moreover, we can directly operate the frequencies, i.e., the features of the original signal in the frequency domain.

Algorithm 1 DTEO

```
Input: $\boldsymbol{P}_G$: parent population at generation $G$; $\boldsymbol{FP}_G$: fitness of $\boldsymbol{P}_G$; $\boldsymbol{O}_G$: offspring population at generation $G$; $\boldsymbol{FO}_G$: fitness of $\boldsymbol{O}_G$; $D$: problem dimensionality; $NP$: population size
Output: $\boldsymbol{TFP}_G$ and $\boldsymbol{TFO}_G$: the transformed fitness of $\boldsymbol{P}_G$ and $\boldsymbol{O}_G$ respectively
1: Perform DFT to transform the solution sequence to frequency domain as frequency series $\boldsymbol{Fs}$ according to Component 1;
2: In the frequency domain, remove frequency series with the top highest frequencies from $\boldsymbol{Fs}$ according to Component 2 and the remaining series is denoted as $\boldsymbol{Fs}$';
3: Perform IDFT on $\boldsymbol{Fs}$'to transform the frequency series back to solution domain as $\boldsymbol{TF}_G$ according to Component 3;
4: Separate $\boldsymbol{TF}_G$ into transformed fitness $\boldsymbol{TFP}_G$ of $\boldsymbol{P}_G$ and transformed fitness $\boldsymbol{TFO}_G$ of $\boldsymbol{O}_G$.
```

IV. Proposed Method

In this article, we introduce the DT from signal processing and communication fields to evolutionary computation with the aims of: 1) providing a new insight on function landscape analysis with DT; 2) alleviating the problems of premature convergence and stagnation which frequently occur on complicated multimodal function landscape; and 3) constructing a new searching paradigm based on DT. This section will first describe the DT-based evolutionary optimization (DTEO) technique and then introduce its implementation on DE for noiseless and noisy optimization, respectively.

A. DT for Evolutionary Optimization

Unlike the existing methodology which mainly focuses on the detection and utilization of original landscape information, this article proposes the DT of landscape methodology to guide the optimization process of an EA. The general idea of DTEO is shown in Algorithm 1, which consists of three components: 1) **T-to-F:** transform the solution sequence from the solution domain to the frequency domain (line 1); 2) **O-in-F:** operate the frequency series in the frequency domain (line 2); and 3) **T-to-S:** transform the processed frequency series back to the solution domain (line 3). These three components are realized by Components 1–3, respectively.

In Component 1 (**T-to-F**), given the parent population $\boldsymbol{P}_G$, the offspring population $\boldsymbol{O}_G$, and their fitness $\boldsymbol{FP}_G$ and $\boldsymbol{FO}_G$ at the Gth generation, line 1 first combines them as a union solution population $\boldsymbol{UP}_G$ and a union fitness set $\boldsymbol{UF}_G$, respectively. This operation will facilitate the comparison of parent and offspring fitness in the same context. Afterward, **T-to-F** is performed for each dimension of the problem (lines 3–6). Specifically, the dth dimension values of $\boldsymbol{UP}_G$ is first set as $\boldsymbol{UP}_{d,G}$ (line 3); second, $\boldsymbol{UP}_{d,G}$ is sorted from the smallest to the largest according to the variable values and the associated fitness $\boldsymbol{UF}_G$ is reindexed as $\boldsymbol{UF}_{d,G}$ accordingly (line 4), which enables the comparison of fitness within neighborhoods; third, the sorted fitness sequence

Component 1 T-to-F (DFT-Based Transformation to Frequency Domain)

Input: $\boldsymbol{P}_G$*:* parent population at generation G; $\boldsymbol{FP}_G$*:* fitness of $\boldsymbol{P}_G$; $\boldsymbol{O}_G$*:* offspring population at generation G; $\boldsymbol{FO}_G$*:* fitness of $\boldsymbol{O}_G$; D*:* problem dimensionality; NP*:* population size
Output: $\boldsymbol{Fs}$**:** the frequency series in frequency domain

```
1: Combine $\boldsymbol{P}_G$ and $\boldsymbol{O}_G$, $\boldsymbol{FP}_G$ and $\boldsymbol{FO}_G$ as a union solution
   population $\boldsymbol{UP}_G$ and a union fitness set $\boldsymbol{UF}_G$ respectively;
2: For $d = 1{:}D$
3:     $\boldsymbol{UP}_{d,G} \leftarrow$ the $d$-th dimension of $\boldsymbol{UP}_G$;
4:     Sort $\boldsymbol{UP}_{d,G}$ according to the variable values from the
       smallest to the largest and reindex the associated $\boldsymbol{UF}_G$ as
       $\boldsymbol{UF}_{d,G}$ accordingly;
5:     Let $m(n) = \boldsymbol{UF}_{d,G}$, $k = 0, 1, \cdots, N - 1$ with $N = 2 \times NP$ and perform DFT calculation to obtain frequency
       series $M$ according to (3);
6:     $\boldsymbol{Fs}_d \leftarrow M$;
7: End For
```

Component 2 O-in-F (Operation in Frequency Domain)

Input: $\boldsymbol{Fs}$**:** the original frequency series in frequency domain; D*:* problem dimensionality
Output: $\boldsymbol{Fs'}$**:** the remaining frequency series in frequency domain

```
1: For $d = 1{:}D$
2:     Remove the top $r \times 100\%$ ($r \in [0,1)$) highest frequencies
       from $\boldsymbol{Fs}_d$;
3: End For
```

$\boldsymbol{UF}_{d,G}$ is treated as the time series, i.e., $m(n)$ in (3) is set as $\boldsymbol{UF}_{d,G}$ and the indices k are 0, 1, …, $N - 1$, where $N = 2 \times NP$ is the series size (i.e., union population size). Then, DFT is performed to obtain frequency series M according to (3) (line 5); and finally, the obtained frequency series M for the dth dimension is stored in $\boldsymbol{Fs}_d$ (line 6).

In Component 2 (**O-in-F**), for each of the D frequency series, the top $r \times 100\%$ ($r \in [0,1)$) highest frequencies are removed from $\boldsymbol{Fs}$ (lines 1–3) to obtain the remaining frequency series $\boldsymbol{Fs'}$.

In Component 3 (**T-to-S**), for each of the D processed frequency series $\boldsymbol{Fs'}_d$ in the frequency domain, they are transformed back to the solution domain with IDFT (lines 2–4). Specifically, $M(k)$ in (5) is first set as $\boldsymbol{Fs'}_d$ and the indices n are 0, 1, ⋯, $N - 1$, where $N = 2 \times NP$ is the series size (i.e., union population size). Then, IDFT is performed to obtain the solution (time) series m according to (5) (line 2); second, the obtained solution series m is reindexed as m' according to the sorting indices in line 4 of Component 1 (line 3); third, the series m' are stored in m'_d (line 4). Finally, the transformed population fitness $\boldsymbol{TF}_G$ in the solution domain is calculated as the average fitness of the D m'_d series

$$TF_G = \sum_{d=1}^{D} m'_d / D. \tag{6}$$

Component 3 T-to-S (IDFT-Based Transformation Back to Solution Domain)

Input: $\boldsymbol{Fs'}$**:** the remaining frequency series in frequency domain; D*:* problem dimensionality;
Output: $\boldsymbol{TF}_G$*:* the transformed fitness in solution domain

```
1: For $d = 1{:}D$
2:     Let $M(k) = \boldsymbol{Fs'}_d$, $n = 0, 1, \cdots, N - 1$ with $N = 2 \times NP$ and perform IDFT calculation to obtain the solution
       series $m$ according to (5);
3:     Reindex $m$ as $m'$ according to the sorting indices in line
       4 of Component 1;
4:     $m'_d \leftarrow m'$;
5: End For
6: $\boldsymbol{TF}_G = \sum_{d=1}^{D} m'_d / D$
```

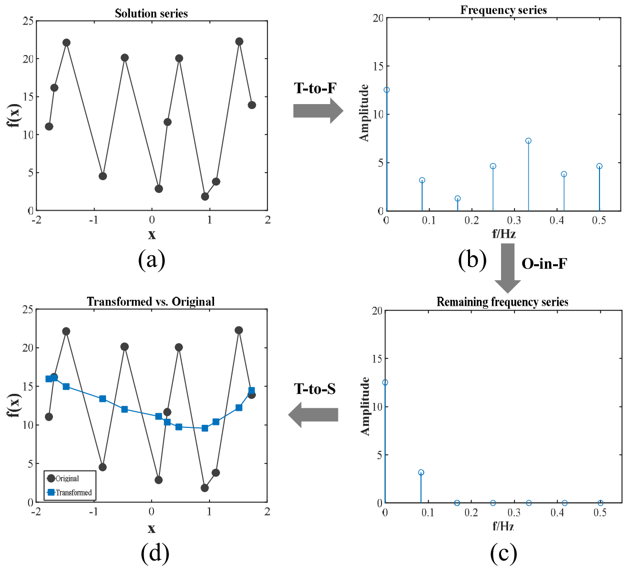


Fig. 2. Illustrative example of DTEO with unilateral amplitude spectrum in the frequency domain. (a) Solution series. (b) Frequency series. (c) Remaining frequency series. (d) Transformed series.

Following the three components, in line 4 of Algorithm 1, $\boldsymbol{TF}_G$ is separated into the transformed parent fitness $\boldsymbol{TFP}_G$ of $\boldsymbol{P}_G$ and the transformed offspring fitness $\boldsymbol{TFO}_G$ of $\boldsymbol{O}_G$, respectively. In the **O-in-F** operation, if there is not any frequency removed (i.e., $r = 0$), $\boldsymbol{TFP}_G$ and $\boldsymbol{TFO}_G$ are the same as $\boldsymbol{FP}_G$ and $\boldsymbol{FO}_G$, respectively, and DTEO degrades to EO.

To illustrate DTEO, a union population, including 12 individuals with one dimension, i.e., $\boldsymbol{UP}_G = \{-0.85, 0.92, 1.51, -0.47, 0.47, -1.69, -1.78, 0.12, 1.11, 1.73, -1.48, 0.27\}$ is considered as an example, and the corresponding fitness are $f(\boldsymbol{UP}_G) = \{4.52, 1.84, 22.25, 20.13, 20.05, 16.17, 11.05, 2.86, 3.81, 13.88, 22.11, 11.65\}$. First, reindex the fitness series $f(\boldsymbol{UP}_G)$ according to the variable values of all individuals from the smallest to the largest to obtain the sorted fitness series $f^{sorted}(\boldsymbol{UP}_G) = \{11.05, 16.17, 22.11, 4.52, 20.13, 2.86, 11.65, 20.05, 1.84, 3.81, 22.25, 13.88\}$ in the solution domain, as shown in Fig. 2(a). Then, the **T-to-F** operation is performed to transform the sorted fitness series into the frequency domain with the utilization of DFT as depicted in Fig. 2(b). Subsequently, to smooth the original fitness landscape with much fluctuations, the **O-in-F** operation is performed to remove the top five highest frequencies, as shown

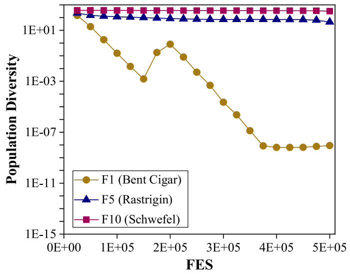


Fig. 3. Illustration of the population diversity over the evolution process with L-SHADE on unimodal Bent Cigar function and multimodal Rastrigin's and Schwefel's functions.

in Fig. 2(c). Finally, the **T-to-S** operation is carried out on the remaining frequency series to transform it back to the solution domain, as shown in Fig. 2(d) with blue line.

B. DT for Traditional Noiseless Optimization

As is known, there is a dilemma between convergence and diversity in EAs. To improve the performance, both problem structure and algorithmic strategy should be simultaneously investigated. On the one hand, explicit analysis of function landscape has been performed to predict the problem difficulty and characteristic to select suitable strategies or algorithms [22], [28], [29]. On the other hand, implicit utilization of function landscape information, such as the successful experience in the searching history has also been employed to adaptively adjust the search strategies [35] as well as the control parameters [36] in an online manner. The significant advance of DE to be state-of-the-art in the past two decades has witnessed the success of the above methodology. Nevertheless, the problems of premature convergence and stagnation are still challenging for DE.

Premature convergence, which usually accompanies with a rapid loss of population diversity is widely known. Herein, we would like to illustrate the stagnation with an example given in Fig. 3, where the population diversity *div* over the evolution process achieved by the state-of-the-art L-SHADE [46] on 50-D unimodal Bent Cigar function (F1 from the CEC2017 test suite [84]) and multimodal Rastrigin's and Schwefel's functions (F5 and F10 from the CEC2017 test suite) is shown. *div* is calculated as

$$div = \frac{1}{NP}\sum_{i=1}^{NP}\|\mathbf{X}_i - \mathbf{C}\| \tag{7}$$

where $\|\mathbf{X}_i - \mathbf{C}\|$ represents the Euclidean distance between $\mathbf{X}_i$ and $\mathbf{C}$, and $\mathbf{C} = (1/NP)\sum_{i=1}^{NP}\mathbf{X}_i$ is the center of the population.

From Fig. 3, L-SHADE could achieve a small $div <$ 1E−08 on the unimodal function, indicating that the algorithm is well converged on the function. However, on the two multimodal functions, the finally obtained *div* is about 45 and 319 and not small enough. As is known, L-SHADE adopts a linear population reduction scheme to accelerate the convergence and the population size is only 4 at the final generation. While Fig. 3 shows that on the multimodal functions, the population converges slowly, and the final population is still quite diverse with insufficient convergence.

Algorithm 2 DT Methodology for DE

```
1:  Initialize the population size $NP$, mutation factor $F = 0.5$ and crossover rate $CR = 0.5$; initialize the population $\boldsymbol{P}_G$; denote the fitness of $\boldsymbol{P}_G$ as $\boldsymbol{FP}_G$; set generation counter $G = 0$;
2:  Perform DTEO (Algorithm 1) with $\boldsymbol{P}_G$ and $\boldsymbol{FP}_G$ to obtain the transformed population fitness $\boldsymbol{TF}_G$ (Herein, $\boldsymbol{FO}_G$ and $\boldsymbol{O}_G$ are empty and $N = NP$); ⇐
3:  While stopping condition is not met
4:      Sort the population based on $\boldsymbol{TF}_G$; ⇐
5:      Perform DE's mutation to generate a mutant population $\boldsymbol{V}_G$;
6:      Perform DE's crossover on $\boldsymbol{V}_G$ and $\boldsymbol{X}_G$ to obtain an offspring population $\boldsymbol{O}_G$;
7:      Evaluate the fitness of $\boldsymbol{O}_G$ and denote it as $\boldsymbol{FO}_G$;
8:      Perform DTEO (Algorithm 1) with $\boldsymbol{P}_G$, $\boldsymbol{TF}_G$, $\boldsymbol{O}_G$, and $\boldsymbol{FO}_G$ to obtain the transformed parent fitness $\boldsymbol{TFP}_G$ and the transformed offspring fitness $\boldsymbol{TFO}_G$; ⇐
9:      Perform DE's selection to select solutions as $\boldsymbol{P}_G$ for the next generation based on $\boldsymbol{TFP}_G$ and $\boldsymbol{TFO}_G$; ⇐
10:     Update the transformed fitness $\boldsymbol{TF}_G$ of $\boldsymbol{P}_G$; ⇐
11:     $G = G + 1$;
12: End While
```

This article incorporates DT with DE to deal with the above problems. The pseudocode of the DT methodology for DE is shown in Algorithm 2. Comparing Algorithm 2 with the original DE, the differences, as marked with "⇐" lie in: 1) DTEO (Algorithm 1) is performed on the initial population (line 2) and the combined population of parent and offspring at each generation (line 8) to obtain the transformed fitness; 2) the sorting process of the population is based on the transformed fitness (line 4) while in the original DE, it is based on $\boldsymbol{FP}_G$; and 3) the selection operation of DE is based on the transformed fitness $\boldsymbol{TFP}_G$ and $\boldsymbol{TFO}_G$ (line 9) while in the original DE, it is based on $\boldsymbol{FP}_G$ and $\boldsymbol{FO}_G$. Overall, the main idea of DT-assisted DE is to use the transformed fitness to replace the original fitness, which could be utilized to construct mutation strategies such as the fitness ranking process in the "current-to-*p*best/1" mutation strategy [16], to reduce population size, such as the population size reduction based on fitness in L-SHADE or to determine the selection.

The motivation of the DT methodology is to remove the high-frequency elements of challenging function landscape to make it smooth to help escape from local optima and stagnation. For demonstration, Fig. 4 depicts an example of the original and DT-based landscapes. Compared the transformed landscape with the original one, it is observed that DT eliminates the large variations of the original landscape. With the transformed fitness, it is easy to jump out from the local minima. To illustrate this, Fig. 4 presents a parent solution (black circle) and its associated offspring solution (green circle) and their transformed fitness are the black and green squares, respectively. According to the selection operation of DE in the original landscape, the offspring is declined since it has

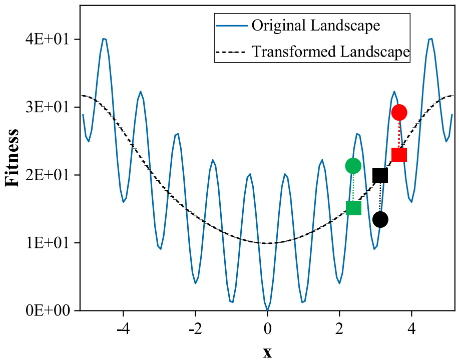


Fig. 4. Illustration of the motivation of the proposed DT methodology.

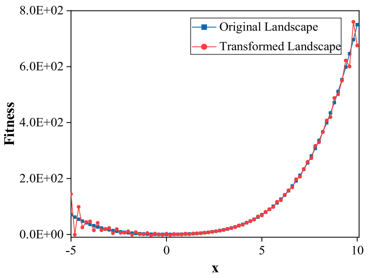


Fig. 5. Effect of DT on a smooth unimodal landscape.

worse fitness. While according to the transformed fitness, this offspring, which is beneficial to basin-jumping is accepted for the next generation since its transformed fitness (green square) is better. Therefore, it is seen that the utilization of the transformed fitness influences the selection of solutions, which is also a critical operation of DE. The original fitness comparison always exhibits a greedy property by preferring the fitter solutions while the transformed fitness comparison could also preserve potential solutions. Note that this process is different from the random acceptance of a worse solution. For explanation, another possible offspring, the red circle solution is also added in Fig. 4. The random acceptance manner might select the red circle offspring for the next generation while the DT method would not as the transformed offspring fitness (red square) is worse than the transformed parent fitness (black square). The reason is that the transformation in DT is based on the original fitness and thus the tendency of the transformed landscape correlates with the original landscape. Experimental studies about the benefit of DT on alleviating stagnation and improving population convergence will be presented in Section V-A4.

While for the originally smooth unimodal function, DT might adversely generate the rugged landscape, as shown in Fig. 5. To resolve this problem, we merge the advantages of both original and transformed landscapes and DT is performed only when the optimization stage becomes difficult.

With these considerations, the pseudocode of the proposed DTDE is shown in Algorithm 3. At the beginning of the optimization, the original DE is performed, while when the optimization enters a relatively difficult stage, DT is performed. To detect the optimization difficulty, we adopt our previously proposed mechanism in [42], where the total fitness improvements Imp_s and Imp_i of the superior and inferior solutions every ten generations are accumulated according to (8) and (9). If Imp_s is larger than Imp_i, the optimization becomes relatively difficult

Algorithm 3 DTDE

```
Initialize the population size $NP$, mutation factor $F$ and crossover factor $CR$; initialize the population $\boldsymbol{P}_G$; evaluate the fitness of $\boldsymbol{P}_G$ and denoted it as $\boldsymbol{FP}_G$; set generation count $G = 0$;
While stopping condition is not met
    If the optimization process becomes difficult then
        Perform DT assisted DE, i.e. (lines 4−11) of Algorithm 2;
    Else
        Perform original DE;
    End If
End While
```

$$\mathrm{Imp}_s = \sum_{g=1}^{10} \sum_{rank(i)=1}^{\lfloor NP/2 \rfloor} \Delta f_i \tag{8}$$

$$\mathrm{Imp}_i = \sum_{g=1}^{10} \sum_{rank(i)=\lceil NP/2 \rceil+1}^{NP} \Delta f_i \tag{9}$$

where

$$\Delta f_i = \begin{cases} f(\mathbf{X}_{i,G}) - f(\mathbf{U}_{i,G}), & \text{if } f(\mathbf{U}_{i,G}) < f(\mathbf{X}_{i,G}) \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

rank(i) denotes the fitness ranking of the ith individual. The smaller, the better. $\lfloor\cdot\rfloor$represents a floor function and $\lceil\cdot\rceil$is a ceiling function.

To demonstrate the performance of the DT methodology and the effectiveness of the difficulty detection mechanism, SCSS-L-SHADE [20] is adopted as the original algorithm and Fig. 6 shows the convergence graphics of the original, DT-based, and DTDE algorithms on five selected 50-D CEC2017 functions, including three multimodal functions (F4: shifted and rotated Rosenbrock's function; F5: shifted and rotated Rastrigin's function; and F10: shifted and rotated Schwefel's function) and two unimodal functions (F1: shifted and rotated Bent Cigar function and F3: shifted and rotated Zakharov function).

It is observed from Fig. 6 that the original SCSS-L-SHADE performs better on the two unimodal functions while the DT-based algorithm is superior on the three multimodal functions. Noticeably, the difficulty detection mechanism effectively takes advantages of both landscapes. Specifically, DTDE outperforms the original algorithm on the multimodal functions F4, F5, and F10 and is competitive on the unimodal functions F1 and F3. While compared with the single DT-based algorithm, DTDE has better performance on the unimodal functions F1 and F3 and multimodal function F4 and comparable performance on the multimodal functions F5 and F10. On F5 and F10, DTDE converges slower than DT-based algorithm since it involves the difficulty detection mechanism. DT is not triggered from the beginning and has a delay. While comparing DTDE with the original algorithm, it is observed

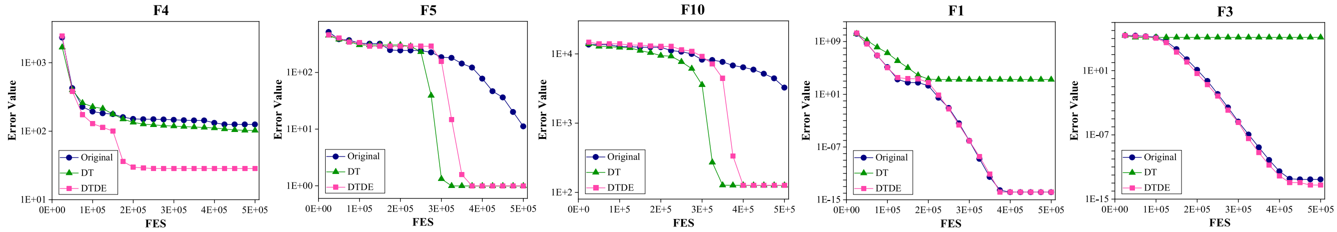


Fig. 6. Convergence graphics of the original, DT-based, and DTDE algorithms on five selected 50-D CEC2017 functions in the trial with the median error value. The maximum number of function evaluations is set as 10 000×D and 51 trials are performed.

that DTDE converges comparably at the early stage. This is because DT has not been triggered at this stage. Furthermore, at the early stage, the stagnation phenomenon in the original algorithm is not as severe as at the late stage since the population is relatively diverse to generate large differential vectors.

With the DT technique, DTDE could find significantly better solutions on several CEC2017 functions that state-of-the-art DEs could merely achieve. More detailed comparisons and discussions will be presented in Section V.

C. DT for Noisy Optimization

The efficiency of an EA in noisy optimization scenario depends on both the performance of the baseline algorithm and the noise handling method. From the reviewed works in Section II-A, a significant advance in DE has been achieved. However, when encountered with noise, mechanisms which are remarkably effective in noiseless optimization might become inefficient. As have been introduced previously, noise hides the real function fitness and hampers the selection which is one of the core operations of EAs. Besides, it also influences the genetic operations. For instance, in the state-of-the-art DEs [5], the population is usually first ranked according to fitness and then mutates toward the top-fittest solutions. If this sorting process is disturbed by noise, so is the effectiveness of the mutation. To construct an effective approach for noisy optimization, these issues need addressing. From Section II-B, existing works on noisy evolutionary optimization mainly paid attention to the sampling and population sizing strategies to compensate for the adverse effect induced by noise. These methods usually require extra function evaluations and increase the computational budget.

Herein, we propose the DT-based noise canceling method. As we have known, DT obtains the transformed fitness by deleting the top highest frequencies in the frequency domain. This is beneficial for handling significant variations and large ruggedness in a noisy fitness landscape. For demonstration, we consider the 1-D sphere function [as shown in (11)] with severe multiplicative Gaussian noise. The resultant noisy fitness f_{noisy} is given by (12)

$$f(\mathbf{x}) = \mathbf{x}^2, \mathbf{x} \in [-100, 100] \tag{11}$$

$$f_{\text{noisy}}(\mathbf{x}) = f(\mathbf{x}) \times e^{\text{Nor}(0,1)} \tag{12}$$

where Nor(0, 1) represents a normal distribution with mean 0 and variance 1.

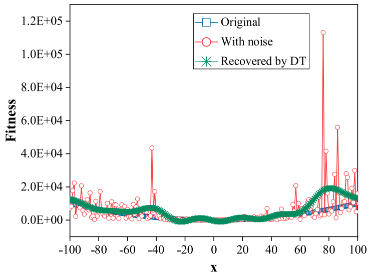


Fig. 7. Plots of the original, noise-induced, and DT-recovered landscapes for the 1-D sphere function.

Algorithm 4 DTDEn

```
1: Initialize the population size $NP$, mutation factor $F$ and crossover factor $CR$; initialize the population $\boldsymbol{P}_G$; evaluate the fitness of $\boldsymbol{P}_G$ and denoted as $\boldsymbol{FP}_G$; set generation count $G = 0$;
2: **While** stopping condition is not met
3:     **If** the noise is severe **then**
4:         Perform DT assisted DE, i.e. (lines 4−11) of **Algorithm** 2;
5:     **Else**
6:         Perform DTDE (**Algorithm** 3);
7:     **End If**
8: **End While**
```

Fig. 7 depicts the original, noise-induced, and DT-recovered (with $r = 0.9$ in **O-in-F**) landscapes. It is seen that for the large fitness values, noise makes the sphere function multimodal shaped and imposes significant difficulties on optimization. While with DT, the noise is suppressed, and the fitness landscape could be partly recovered.

With severe noise, the noisy landscape of the originally smooth unimodal landscape becomes rough. Therefore, in this case, DT should be performed from the beginning for the unimodal functions. The pseudocode of the proposed DTDEn algorithm for noisy optimization is shown in Algorithm 4. When the noise is severe, DT is performed for the entire evolution process (line 4). Otherwise, Algorithm 3 is performed (line 6), in which DT is triggered only when the optimization process becomes difficult.

To detect the strength of noise, a solution is randomly generated and evaluated for K times ($K = 30$) in the presence of noise and the noisy fitness is stored in NF(k), $k \in \{1, 2, 3, \cdots, K\}$. With NF, a noisy strength associated parameter sp is

TABLE I
PERFORMANCE COMPARISONS OF DTDE WITH THE ORIGINAL SCSS-L-SHADE ON 50-D AND 100-D CEC2017 TEST FUNCTIONS

	50-D			100-D				50-D			100-D		
	Original		DTDE	Original		DTDE		Original		DTDE	Original		DTDE
	mean (std)	sig	mean (std)	mean (std)	sig	mean (std)		mean (std)	sig	mean (std)	mean (std)	sig	mean (std)
F1	0.00E+00 (0.00E+00)	=	0.00E+00 (0.00E+00)	0.00E+00 (0.00E+00)	=	0.00E+00 (0.00E+00)	F17	2.21E+02 (8.30E+01)	+	**4.93E+01** **(4.91E+01)**	1.07E+03 (2.20E+02)	+	**8.87E+01** **(9.96E+01)**
F3	0.00E+00 (0.00E+00)	=	0.00E+00 (0.00E+00)	1.50E-04 (1.97E-04)	=	1.33E-04 (2.49E-04)	F18	**2.64E+01** **(4.26E+00)**	−	3.65E+01 (6.78E+00)	2.17E+02 (5.17E+01)	=	2.30E+02 (5.54E+01)
F4	9.80E+01 (4.80E+01)	+	**6.07E+01** **(5.15E+01)**	1.97E+02 (1.89E+01)	=	2.01E+02 (1.11E+01)	F19	1.53E+01 (2.63E+00)	+	**9.41E+00** **(2.34E+00)**	1.71E+02 (2.01E+01)	+	**1.62E+02** **(2.14E+01)**
F5	1.17E+01 (2.41E+00)	+	**1.50E+00** **(1.26E+00)**	2.82E+01 (4.54E+00)	+	**2.11E+00** **(1.46E+00)**	F20	1.73E+02 (6.64E+01)	+	**2.32E+01** **(3.69E+00)**	1.46E+03 (2.16E+02)	+	**1.80E+02** **(5.87E+01)**
F6	4.28E-08 (1.07E-07)	=	1.14E-07 (3.53E-07)	1.92E-03 (1.70E-03)	=	1.52E-03 (1.21E-03)	F21	2.13E+02 (2.58E+00)	+	**2.02E+02** **(1.99E+00)**	2.52E+02 (4.02E+00)	+	**2.22E+02** **(3.64E+00)**
F7	6.29E+01 (2.48E+00)	+	**5.65E+01** **(9.22E-01)**	1.32E+02 (4.20E+00)	+	**1.11E+02** **(1.40E+00)**	F22	2.86E+03 (1.55E+03)	+	**2.97E+02** **(1.34E+02)**	1.09E+04 (5.20E+02)	+	**1.06E+03** **(3.25E+02)**
F8	1.15E+01 (2.38E+00)	+	**1.54E+00** **(1.29E+00)**	2.88E+01 (3.88E+00)	+	**2.01E+00** **(1.39E+00)**	F23	4.27E+02 (5.28E+00)	+	**4.19E+02** **(5.60E+00)**	5.62E+02 (1.08E+01)	+	**5.37E+02** **(8.10E+00)**
F9	0.00E+00 (0.00E+00)	=	0.00E+00 (0.00E+00)	8.68E-02 (1.50E-01)	=	9.56E-02 (1.67E-01)	F24	5.05E+02 (3.16E+00)	+	**4.98E+02** **(2.08E+00)**	9.05E+02 (7.36E+00)	+	**8.84E+02** **(4.54E+00)**
F10	3.17E+03 (2.66E+02)	+	**1.61E+02** **(8.83E+01)**	9.83E+03 (5.30E+02)	+	**4.03E+02** **(2.73E+02)**	F25	4.82E+02 (4.26E+00)	=	4.81E+02 (3.52E+00)	7.31E+02 (3.85E+01)	=	7.27E+02 (4.21E+01)
F11	3.18E+01 (3.66E+00)	+	**3.01E+01** **(4.48E+00)**	**1.67E+02** **(5.18E+01)**	−	1.95E+02 (5.93E+01)	F26	1.12E+03 (4.97E+01)	+	**9.84E+02** **(4.04E+01)**	3.21E+03 (9.08E+01)	+	**3.05E+03** **(7.90E+01)**
F12	2.02E+03 (4.70E+02)	=	2.06E+03 (4.42E+02)	1.80E+04 (6.78E+03)	=	1.83E+04 (8.75E+03)	F27	**5.25E+02** **(1.64E+01)**	−	5.34E+02 (2.05E+01)	6.22E+02 (2.09E+01)	=	6.26E+02 (2.04E+01)
F13	4.96E+01 (2.62E+01)	=	5.39E+01 (2.99E+01)	1.70E+02 (5.04E+01)	=	1.58E+02 (4.47E+01)	F28	4.62E+02 (1.10E+01)	=	4.65E+02 (1.59E+01)	5.26E+02 (2.27E+01)	=	5.28E+02 (2.17E+01)
F14	2.53E+01 (2.40E+00)	=	2.47E+01 (4.49E+00)	8.46E+01 (1.59E+01)	+	**6.52E+01** **(1.49E+01)**	F29	3.63E+02 (1.44E+01)	+	**3.00E+02** **(7.55E+00)**	1.29E+03 (1.59E+02)	+	**8.13E+02** **(1.31E+02)**
F15	2.61E+01 (3.06E+00)	=	2.62E+01 (3.60E+00)	2.56E+02 (4.41E+01)	=	2.56E+02 (5.16E+01)	F30	6.52E+05 (5.92E+04)	=	6.58E+05 (6.40E+04)	2.34E+03 (1.52E+02)	=	2.30E+03 (1.14E+02)
F16	3.47E+02 (1.23E+02)	+	**1.44E+02** **(4.66E+01)**	1.54E+03 (2.31E+02)	+	**2.00E+02** **(1.23E+02)**	**W/T/L**	**16**/11/2			**15**/13/1		

calculated as

$$sp = \frac{\max(\mathrm{NF}) - \min(\mathrm{NF})}{\max(\mathrm{NF})} \tag{13}$$

where $\min(\cdot)$ and $\max(\cdot)$ represent the minimum and maximum values, respectively. When sp is larger than a preset threshold value T, the noise strength is identified as severe.

D. Time Complexity of DT

At one generation, the time complexity of FFT- and IFFT-based transform operations in **T-to-F** and **T-to-S** for one dimension are both $O(2 \times NP \times \log_2(2 \times NP))$. In addition, the time complexity of the sorting of variables for one dimension in **T-to-F** is $O(2 \times NP \times \log_2(2 \times NP))$. Therefore, the computational overhead of the proposed DT method at one generation is $O(6 \times D \times NP \times \log_2(2 \times NP))$.

V. SIMULATION

In this section, experiments are performed to verify the effectiveness of the proposed DT method. The CEC2017 test suite [84], which consists of 29 minimization functions are adopted. These functions are characterized by various mathematical properties, such as the unimodal functions F1 and F3, simple multimodal functions F4–F10, hybrid functions F11–F20, and composition functions F21–F30. To compare the performance, 51 trials are performed for each algorithm on each function and the finally obtained solution error value *SE* is compared. *SE* is defined as $f(x) - f(x^*)$, where $f(x^*)$ is the optimal function value and $f(x)$ is the minimum fitness value achieved with $10^4 \times D$ function evaluations. Note that for the noisy optimization, the minimal real function error value is reported as *SE*. All the algorithms were implemented in MATLAB and run on an Intel Core i9 3.10-GHz PC with 16 GB of RAM in Windows 10 system.

To draw statistically sound conclusions, the Wilcoxon signed-rank test [85] with 5% significance level is employed to compare the 51 entries on each function. The symbols "+," "=," and "−" indicate that the DT-based method is superior (i.e., win, W), similar (i.e., tie, T), and inferior (i.e., lose, L) to the compared algorithm, respectively.

A. Experiments on Traditional Noiseless Optimization

1) Performance Comparison With Competitive Baseline Algorithm: To demonstrate the effectiveness of the proposed DT methodology for traditional noiseless optimization, it is first incorporated with a state-of-the-art baseline algorithm, the SCSS-L-SHADE [20]. As is known, L-SHADE is a competitive DE with linear population size reduction. While selective candidate with similarity selection rule (SCSS) method [20] significantly improves the performance of L-SHADE. Following the procedures shown in Algorithm 2, when incorporating DT with SCSS-L-SHADE, fitness ranking in the "current-to-*p*best" mutation and population size reduction scheme are also based on the transformed fitness. The improved variant and the original SCSS-L-SHADE are denoted as **DTDE** and **Original**, respectively. DTDE adopts the same parameter settings as Original for fair comparisons.

Table S1 in the supplementary file and Table I show the performance comparison of DTDE with the baseline on the 30-D, 50-D, and 100-D CEC2017 benchmark functions. At

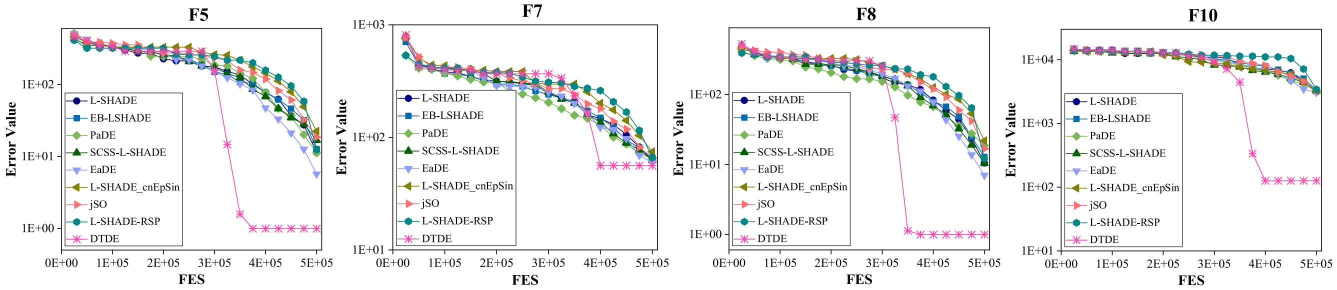


Fig. 8. Convergence graphics of the best error value achieved by the compared DEs on the 50-D CEC2017 functions F5, F7, F8, and F10 in the trial with the median error value.

the first glance, DT significantly enhances the performance of SCSS-L-SHADE on all the considered dimensionalities. Statistically, DTDE performs better in 13, 16, and 15 and loses in 3, 2, and 1 case on 30-D, 50-D, and 100-D functions, respectively.

Considering the function properties, the followings are observed.

1) On the unimodal functions with 6 cases, DTDE performs similarly to the baseline.
2) On the simple multimodal functions F4–F10 with 21 cases, DTDE wins in 14 cases and loses in none case. Particularly, we noticed that on F5, F7, F8, and F10, DTDE achieves the best-so-far performance with the commonly adopted function evaluation budget of $10^4 \times D$ when compared with other state-of-the-art DE variants, such as L-SHADE [46], EB-LSHADE [86], PaDE [50], SCSS-L-SHADE, EaDE [42], L-SHADE_cnEpSin [49], jSO [47], and L-SHADE-RSP [87]. For illustration, Fig. 8 shows the convergence graphics on the 50-D F5, F7, F8, and F10. As seen, DTDE exhibits significant performance superiority when compared with the competitors. For instance, on F5, the median error value obtained by the compared DEs is above 5 while DTDE achieves a value around 0.99. On F10, the median error value achieved by the compared DEs is around 3000, while it is about 126 by DTDE.
3) On the hybrid functions F11–F20, DTDE exhibits better performance in 13 out of the total 30 cases and underperforms in five cases (30-D F14, F15, F18, 50-D F18, and 100-D F11).
4) On the composition functions F21–F30 with 30 cases, DTDE wins in 17 cases and loses in 1 case (50-D F27).

In summary, on the one hand, from the error value perspective, Table S1 and Table I show that most of the improvements are remarkable, despite that the baseline is rather competitive. For instance, on 100-D F5, F16, F17, and F22, the mean error value decreases from 2.82E+01 to 2.11E+00, 1.54E+03 to 2.00E+02, 1.07E+03 to 8.87E+01, and 1.09E+04 to 1.06E+03, respectively. On the other hand, from the statistical "W/T/L" performance, DTDE performs better on the simple multimodal functions, the hybrid functions, and the composition functions and is comparable on the unimodal functions.

2) Performance Comparison With State-of-the-Art DEs: To further assess the significance of the performance enhancement contributed by the DT mechanism, DTDE is compared with six state-of-the-art DE algorithms.

TABLE II
COMPARISON RESULTS OF DTDE WITH STATE-OF-THE-ART DES ON 30-D, 50-D, AND 100-D CEC2017 FUNCTIONS

W/T/L	30-D	50-D	100-D
EB-LSHADE	**15**/11/3	**20**/6/3	**21**/5/3
PaDE	**14**/10/5	**21**/5/3	**22**/3/4
EaDE	**12**/12/5	**16**/11/2	**15**/13/1
L-SHADE_cnEpSin	**12**/11/6	**15**/7/7	**13**/3/13
jSO	**13**/7/9	**15**/6/8	**13**/7/9
L-SHADE-RSP	**14**/6/9	**14**/5/10	**13**/4/12

EB-LSHADE [86]: An improved L-SHADE algorithm with novel mutation strategy.

PaDE [50]: An enhanced L-SHADE algorithm with novel control parameter adaptation schemes.

EaDE [42]: Explicitly adaptive DE with two baseline algorithms, i.e., SCSS-L-SHADE and SCSS-L-CIPDE.

L-SHADE_cnEpSin [49]: An improved L-SHADE algorithm with an ensemble of sinusoidal parameter adaptation and covariance matrix based crossover. It is an enhanced version of the joint winner in the CEC2016 competition.

jSO [47]: An enhanced L-SHADE variant with weighting-based scaling factor and fine-tuning parameter settings at different evolution stages, which is the first ranked DE algorithm in the CEC2017 competition.

L-SHADE-RSP [87]: An improved jSO algorithm with rank-based selective pressure strategy, which is the first ranked DE algorithm in the CEC2018 competition.

Tables S2–S4 in the supplementary file present the comparisons on the 30-D, 50-D, and 100-D functions while Table II summarizes the results.

From Table II, compared with EB-LSHADE, PaDE, and EaDE, DTDE is superior on 56, 57, and 43 functions and inferior on 9, 12, and 8 functions, respectively. Compared with L-SHADE_cnEpSin, DTDE performs better in the 30-D and 50-D cases and is competitive in the 100-D case. From Tables S2–S4, it is found that L-SHADE_cnEpSin mainly shows advantages for solving the hybrid functions with 4, 3, and 7 wins in the 30-D, 50-D, and 100-D cases, respectively. The main contribution to the superior performance on hybrid functions in L-SHADE_cnEpSin is the sinusoidal parameter adaptation according to our component experiment by deactivating each single mechanism. With respect to jSO, the "W/L" metric achieved by DTDE is "13/9," "15/8," and "13/9." Similar to L-SHADE_cnEpSin, jSO also exhibits better

TABLE III
COMPARISON RESULTS OF DTDE-JSO AND DTDE-RSP WITH ORIGINAL BASELINES ON 30-D, 50-D, AND 100-D CEC2017 FUNCTIONS

W/T/L	30-D	50-D	100-D
vs. SCSS-jSO	**14**/10/5	**18**/9/2	**17**/11/1
vs. SCSS-L-SHADE-RSP	**13**/12/4	**18**/10/1	**16**/11/2

TABLE IV
COMPARISON RESULTS OF DTDE-RSP WITH STATE-OF-THE-ART DES ON 30-D, 50-D, AND 100-D CEC2017 FUNCTIONS

W/T/L	30-D	50-D	100-D
EB-LSHADE	**16**/9/4	**25**/4/0	**25**/2/2
PaDE	**19**/6/4	**23**/5/1	**25**/1/3
EaDE	**16**/8/5	**23**/5/1	**24**/5/0
L-SHADE_cnEpSin	**14**/10/5	**20**/4/5	**18**/3/8
jSO	**13**/9/7	**19**/7/3	**21**/5/3
L-SHADE-RSP	**15**/12/2	**20**/8/1	**22**/6/1

TABLE V
OVERALL PERFORMANCE RANKING ON ALL THE CONSIDERED 30-D, 50-D, AND 100-D CEC2017 FUNCTIONS

Algorithm	Ranking	Algorithm	Ranking
EB-LSHADE	7.19	jSO	5.75
PaDE	7.32	L-SHADE-RSP	5.21
SCSS-L-SHADE	6.43	DTDE	4.56
L-SHADE_cnEpSin	6.01	DTDE-jSO	3.77
EaDE	5.90	DTDE-RSP	**2.81**

performance for solving 6, 4, and 4 30-D, 50-D, and 100-D hybrid functions, respectively. The reason is the fine-tuning parameter settings in jSO according to our component experiment. With respect to L-SHADE-RSP, since it is an improved version of jSO, the overall performance superiority of DTDE is less significant than the case against jSO.

3) Flexibility of DT: To demonstrate the flexibility of DT, it is further incorporated into another two algorithms, SCSS-jSO and SCSS-L-SHADE-RSP, resulting in two variants, named DTDE-jSO and DTDE-RSP, respectively. From Table S5 in the supplementary file and Table III, DT also advances the performance of these algorithms, winning in 49 and 47 and losing in 8 and 7 functions, respectively. Table IV shows the comparison results of DTDE-RSP with the state-of-the-art DEs. Comparing Table IV with Table II, it is seen that with a more competitive baseline, the performance of DTDE-RSP is much more attractive. This is also confirmed by the Friedman test given in Table V in which DTDE-RSP achieves the overall best performance with the smallest ranking value of 2.81, followed by DTDE-jSO (3.77) and DTDE (4.56).

To further show the statistical significance, Fig. 9 plots the critical difference diagram [88], where algorithms with no significant difference are connected. It shows that DTDE is statistically better than the baseline while DTDE-RSP is statistically better than all the non-DT-based DEs.

The performance of DTDE-RSP has also been investigated on four real-world problems (RP1–RP4) [89], including the parameter estimation for frequency modulated sound waves problem (RP1), the spread spectrum radar polly phase code design problem (RP2), the "Messenger" spacecraft trajectory optimization problem (RP3) and the "Cassini 2" spacecraft trajectory optimization problem (RP4). From Table S6 in the supplementary file, DTDE-RSP performs better than the baseline on RP2 and RP3 and similarly on RP1 and RP4. While it achieves better performance than jSO, EaDE, and L-SHADE-RSP on 2 (RP2 and RP4), 2 (RP2 and RP3), and 3 (RP1–RP3) problems and loses on none, 1 (RP4), and none problem, respectively.

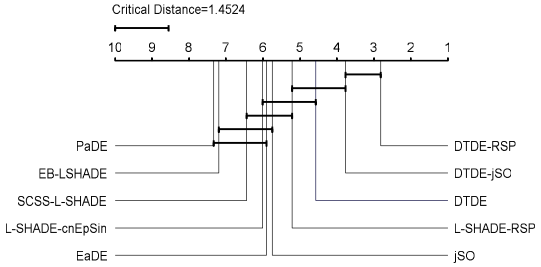


Fig. 9. Comparison of DEs on all the 30-D, 50-D, and 100-D functions with critical difference value.

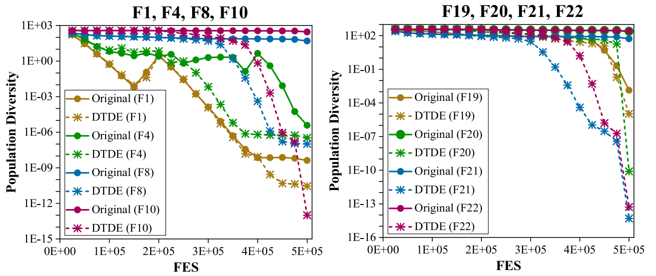


Fig. 10. Convergence graphics of the population diversity achieved by the original baseline and DTDE on the 50-D CEC2017 functions F1, F4, F8, F10, and F19–F22 in the trial with the median error value.

Section IV-D discussed the time complexity of DT. To quantify the overhead, the runtime of DTDE-RSP and the baseline on RP2 and RP3 with the same function evaluations is recorded, respectively, in which the fitness evaluation of RP2 is relatively cheap while it is expensive for RP3. The overhead, calculated by dividing the runtime of DTDE-RSP by that of the baseline is 1.037 and 1.005 on RP2 and RP3 according to experiments, respectively. It shows that the overhead of DT in DTDE-RSP is acceptable.

4) Working Mechanism of DTDE:

a) Population convergence by DTDE: The experiments in Section V-A1 show that DTDE generally yields better performance than the baseline from the solution error perspective. To provide a deeper insight into the optimization process, we consider the population diversity *div*, which is calculated according to (7).

Fig. 10 depicts the convergence of the *div* value achieved the baseline SCSS-L-SHADE and DTDE on eight selected 50-D CEC2017 functions. From Fig. 10, with respect to the baseline algorithm, on the unimodal function F1, it could approach a small diversity value around 1.00E−09. However, on the multimodal functions, for instance, F8, F10, and F20–F22, *div* is above a large value (herein, 50), indicating that the population is not well converged. While with DTDE, the population on all the eight functions converges significantly better with *div* all below 1.00E−05.

b) Population successful update by DTDE: To study the underlying reason that leads to the convergence difference of

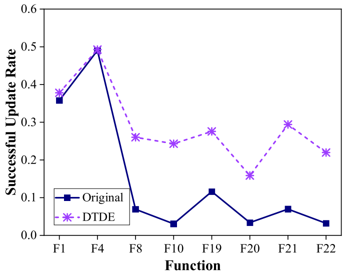
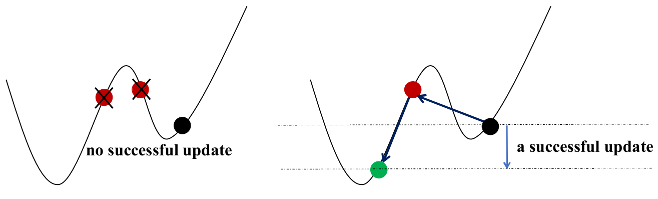


Fig. 11. Average successful update rate achieved by the original baseline and DTDE on the 50-D CEC2017 functions F1, F4, F8, F10, and F19–F22.



(a) (b)

Fig. 12. Rationale behind higher *SR* by DTDE. (a) Original. (b) DTDE.

the population diversity, we also pay attention to the successful update rate *SR* of the population, which is calculated as

$$SR = SU/NP \tag{14}$$

where the successful update counter *SU* is the number of parent solutions that are successfully replaced by the offspring (i.e., the fitness of offspring is better than parent) at each generation and *NP* is the population size.

Fig. 11 shows the average *SR* over the entire evolution by the original baseline and DTDE on the eight considered functions. Note that in DTDE, *SR* is also calculated using the original fitness instead of the transformed fitness in order to compare the evolution differences with the original baseline in the context of the original domain. From Fig. 11, on seven functions F1, F8, F10, and F19–F22, DTDE achieves higher *SR* than the baseline and the ratio $SR_{DTDE}/SR_{Original}$ is about 1.05, 3.76, 7.91, 2.37, 4.69, 4.21, and 6.85, respectively. The rationale behind the higher *SR* is illustrated in Fig. 12, where the evolution situation of a parent solution (black dot) by DTDE and the baseline on a given fitness landscape at two consecutive generations is shown. For the baseline, from Fig. 12(a), the two generated offspring (red dot) are both with worse fitness and are declined. Therefore, there is no successful update at these two generations and the parent stagnates. While for DTDE, as we have known from Fig. 4, it would accept solutions with worse original fitness but better transformed fitness. Fig. 12(b) shows a scenario where it accepts the worse offspring (in terms of original fitness, red dot) at the first generation and then the better offspring (in terms of original fitness, green dot) at the second generation. Therefore, there is a successful update from the black dot to the green dot at these two generations. The successful updates from one basin to another resolve the problem of stagnation and accelerate the convergence of the population.

c) Fitness landscape reconstruction in DTDE: DT reconstructs the fitness landscape by removing the top highest frequencies. To verify the effectiveness of this deletion operation, the following two variants are constructed.

TABLE VI
Comparison Results (W/T/L) of DTDE With the Variants

Variant-L	Variant-R	Variant-DTo
29/0/0	**22**/7/0	**14**/7/8

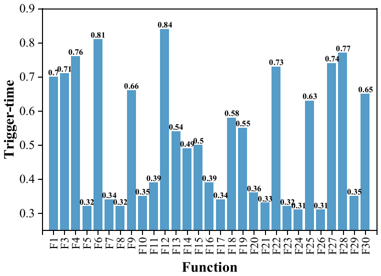


Fig. 13. Trigger-time of the DT mechanism.

Variant-L: Instead of the highest frequencies, in this variant, the *r* lowest frequencies are removed.

Variant-R: In this variant, the *r* frequencies to be removed are randomly selected.

Besides, to investigate the effect of the combination of both original and transformed fitness, Variant-DTo is also considered, in which only the transformed fitness is utilized.

Except the above differences, other settings are kept the same as DTDE. From Table S7 in the supplementary file and Table VI, compared with Variant-L and Variant-R, clear performance superiority could be observed with the "W/T/L" metric of "29/0/0" and "22/7/0," respectively. As we know, Variant-L is exactly opposed to DTDE and it loses in all the cases. While Variant-R removes frequencies randomly and it tends to outperform Variant-L, as is witnessed in Table S7. Compared with Variant-DTo, DTDE wins on 14 functions, including F1, F3, F4, F11–F15, F18, F19, F25, F27, F28, and F30 and loses on 8 functions, including F5, F7, F8, F16, F17, F21, F24, and F26. Statistically, DTDE is superior on the unimodal and hybrid functions while Variant-DTo has strength on the simple multimodal and composition functions. Fig. 13 shows the trigger-time $T_{\text{trig}}(T_{\text{trig}} = \text{FEs/MaxFEs})$ of the DT mechanism in DTDE on the 29 30-D CEC2017 functions, where FEs is the consumed function evaluations before DT is triggered while MaxFEs is the allocated maximum function evaluations, i.e., $10^4 \times D$. From Fig. 13, on the functions that Variant-DTo performs better, including F5, F7, F8, F16, F17, F21, F24, and F26 as seen from Table S7, the trigger-time is relatively early with T_{trig} smaller than 0.4. While on the functions that DTDE exhibits advantages, including F1–F3, F11–F15, F18, F19, F25, F27, F28, and F30, most of the trigger-time is relatively late above 0.5. These confirm the effectiveness of the difficulty detection mechanism.

d) Performance sensitivity to r: The parameter *r* controls the percentages of the highest frequencies removed and it fundamentally affects the performance of DT. To study the sensitivity, nine values from 0.1 to 0.9 with a step of 0.1 are investigated. Among all the considered *r* values, $r = 0.2$ which performs the best is referred as the standard setting. From Table VII, $r = 0.2$ performs similarly to $r = 0.3$ and better

TABLE VII
COMPARISON RESULTS (W/T/L) OF THE REST SETTINGS WITH THE STANDARD SETTING ON 30-D CEC2017 FUNCTIONS

0.1	0.3	0.4	0.5	0.6	0.7	0.8	0.9
4/24/1	**1**/27/1	**2**/26/1	**4**/23/2	**5**/22/2	**6**/21/2	**7**/20/2	**8**/21/0

TABLE VIII
COMPARISON RESULTS WITH THE ORIGINAL SCSS-L-SHADE IN NOISY OPTIMIZATION WITH SLIGHT, MODERATE, AND SEVERE NOISE

W/T/L	30-D	50-D	100-D	**Total**
slight	**18**/9/2	**20**/7/2	**25**/3/1	**63**/19/5
moderate	**19**/7/3	**21**/6/2	**25**/0/4	**65**/13/9
severe	**22**/1/6	**24**/0/5	**25**/0/4	**71**/1/15

than the rest settings. Generally, the performance degenerates as r becomes over-large or over-small. Specifically, when r is larger than 0.3, the superiority of $r = 0.2$ is more significant as r increases. While when r is smaller than 0.3, the superiority of $r = 0.2$ is more significant as r decreases with the "W/L" metric against $r = 0.1$ and $r = 0$ (i.e., the baseline) of "4/1" and "13/3," respectively. The underlying reasons are as follows: when r is over-large, the transformed fitness landscape becomes flat and lacks of selective pressure while when r is over-small, the DT mechanism could not sufficiently take effect.

B. Experiments on Noisy Optimization

In the noisy optimization, we consider the multiplicative Gaussian noise, as shown in (15), where the noise strength ns is set as 0.1, 0.5, and 1.0 to simulate slight, moderate, and severe noise, respectively

$$f_{\text{noisy}}(\mathbf{x}) = f(\mathbf{x}) \times e^{ns\times \text{Nor}(0,1)}. \tag{15}$$

1) Comparison With Baseline Algorithm Without Noise Handling Technique: To demonstrate the effectiveness of the proposed method for noisy optimization, we first consider the original SCSS-L-SHADE as a baseline. Tables S8–S10 in the supplementary file show the comparison results in the cases with slight, moderate, and severe noise, respectively, while Table VIII collects the results. From Table VIII, DTDEn advances the baseline with the "W/L" metric of "63/5," "65/9," and "71/15" in the three cases, respectively.

According to the proposed mechanism in DTDEn, DT is performed from the beginning only when the noise is severe. While DTDE is performed in the slight and moderate noise cases. To show the effectiveness, we consider the following modifications: in the slight and moderate noise cases, DT is performed for the entire evolution process and the resultant variants, are denoted as DTo_slight and DTo_moderate, respectively. From Table S11 in the supplementary file, the comparison results ("W/L" metric) of DTo_slight and DTo_moderate with the baseline are "16/11" and "18/9," respectively. Considering the severe noise case, from Table S10, the "W/L" result of DTDEn (equivalent to DTo_severe) is "22/6." With these results, the following conclusions could be drawn: 1) comparing DTDEn with DTo_slight and DTo_moderate, DTDEn exhibits much better performance as the "W/L" metric against the baseline is "18/2" and "19/3," respectively and 2) as the noise strength increases, the superiority of DT becomes more significant.

TABLE IX
COMPARISON RESULTS WITH RESAMPLING METHOD IN NOISY OPTIMIZATION WITH SLIGHT, MODERATE, AND SEVERE NOISE

W/T/L	30-D	50-D	100-D	**Total**
slight	**21**/5/3	**23**/6/0	**26**/3/0	**70**/14/3
moderate	**16**/8/5	**17**/6/6	**19**/6/4	**52**/20/15
severe	**21**/1/7	**23**/1/5	**25**/0/4	**69**/2/16

TABLE X
COMPARISON RESULTS WITH STATE-OF-THE-ART DES IN NOISY OPTIMIZATION WITH SLIGHT, MODERATE, AND SEVERE NOISE

	W/T/L	30-D	50-D	100-D	**Total**
slight	EaDE	**18**/8/3	**20**/7/2	**23**/5/1	**61**/20/6
	jSO	**18**/6/5	**22**/3/4	**22**/2/5	**62**/11/14
	L-SHADE-RSP	**17**/5/7	**19**/5/5	**16**/5/8	**52**/15/20
moderate	EaDE	**18**/8/3	**20**/6/3	**25**/1/3	**63**/15/9
	jSO	**22**/5/2	**25**/3/1	**25**/2/2	**72**/10/5
	L-SHADE-RSP	**18**/6/5	**19**/6/4	**25**/0/4	**62**/12/13
severe	EaDE	**22**/2/5	**24**/0/5	**25**/0/4	**71**/2/14
	jSO	**22**/1/6	**24**/0/5	**25**/0/4	**71**/1/15
	L-SHADE-RSP	**20**/2/7	**23**/1/5	**25**/0/4	**68**/3/16

To determine the threshold value T, which is used to identify noise strength, we investigate the performance of DTo and DTDE with ns set from 0.5 to 1.0 with a step of 0.1. As we have known, when ns is set as 0.5, DTDE is better while when ns is set as 1, DTo is better. Therefore, T is set according to the smallest ns value where DTo performs better than DTDE. That is, $ns = 0.7$ and thus $T = 0.93$ according to experiments.

2) Comparison With Resampling Method: The proposed DT methodology is further compared with a classic noise handling technique, i.e., the resampling method, in which each solution is evaluated for ST times and the average noisy fitness is treated as the final noisy fitness. As for the baseline algorithm, we have tested $ST = \{2, 3, 4, 5\}$, from which $ST = 2$ performs the best and is adopted as the preferred setting. Tables S12–S14 in the supplementary file present the comparisons in the slight, moderate, and severe noise cases, respectively. As summarized in Table IX, DTDEn generally yields better performance than the resampling method. As for the resampling method, since extra function evaluations are needed for multiple samplings, the maximum number of generations significantly reduces. While it is not the case in DT. The overall performance ranking of the baseline, resampling, and DTDEn given by the Friedman test is 2.49, 2.28, and 1.23, respectively, from which it is seen that resampling performs better than the baseline while the performance advancement by DT is much more significant.

3) Comparison With State-of-the-Art DEs: It is also interesting to compare the performance variations of state-of-the-art DEs in the presence of noise. To this end, we consider three top-performing DEs, i.e., EaDE, jSO, and L-SHADE-RSP. As shown in Tables S15–S17 in the supplementary file and Table X, DTDEn performs much better with the total "W" count significantly larger than the "L" count in all the three cases. Moreover, comparing Table X with Table II, it is seen that the superiority of DT-based DE over the three DEs is much more significant in the noisy scenario. This further confirms

TABLE XI
COMPARISON RESULTS WITH POPULAR NOISY OPTIMIZATION ALGORITHMS WITH SLIGHT, MODERATE, AND SEVERE NOISE

	W/T/L	30-D	50-D	100-D	**Total**
slight	ODE	**27**/1/1	**27**/1/1	**26**/2/1	**80**/4/3
	MUDE	**29**/0/0	**28**/0/1	**28**/1/0	**85**/1/1
	IPOP-UH-CMA-ES	**29**/0/0	**26**/2/1	**23**/1/5	**78**/3/6
	BBPSO_CJ	**29**/0/0	**29**/0/0	**29**/0/0	**87**/0/0
	DEPSO	**28**/1/0	**29**/0/0	**28**/1/0	**85**/2/0
moderate	ODE	**29**/0/0	**29**/0/0	**29**/0/0	**87**/0/0
	MUDE	**29**/0/0	**29**/0/0	**29**/0/0	**87**/0/0
	IPOP-UH-CMA-ES	**27**/2/0	**26**/0/3	**21**/3/5	**74**/5/8
	BBPSO_CJ	**29**/0/0	**29**/0/0	**29**/0/0	**87**/0/0
	DEPSO	**29**/0/0	**29**/0/0	**29**/0/0	**87**/0/0
severe	ODE	**23**/1/5	**25**/0/4	**27**/0/2	**75**/1/11
	MUDE	**29**/0/0	**25**/1/3	**28**/1/0	**82**/2/3
	IPOP-UH-CMA-ES	**23**/2/4	**24**/0/5	**25**/1/3	**72**/3/12
	BBPSO_CJ	**28**/0/1	**29**/0/0	**29**/0/0	**86**/0/1
	DEPSO	**27**/1/1	**28**/0/1	**29**/0/0	**84**/1/2

the effectiveness and contribution of DT, which makes DTDEn more resistant to noise.

4) Comparison With Popular Noisy Evolutionary Algorithms: DTDEn has also been compared against several popular NEAs to demonstrate its performance. Among them, ODE [64] and MUDE [69] integrate opposition-based learning and memetic search for noisy optimization, respectively. IPOP-UH-CMA-ES [63] is an improved IPOP-CMA-ES algorithm incorporated with an uncertainty handling technique. BBPSO_CJ [67] is a Bare bones PSO variant with chaotic jumps and DEPSO [70] is a dual-environmental PSO with weighted search center. As is evident from Tables S18–S20 in the supplementary file and Table XI, DTDEn outperforms in more than 70 cases and underperforms in no more than 12 cases compared with each of the competitors.

VI. Conclusion

To deal with the premature convergence and stagnation of DE and accelerate the population convergence, this article proposes the DT for fitness landscape analysis. DT first transforms the current evolution information to the frequency domain, which facilitates the easy observation of the components. Afterward, the top highest frequencies are removed in the frequency domain to smooth the landscape. Finally, the remaining frequencies are transformed back to the original domain. With the deletion operation in the frequency domain, the transformed fitness landscape is beneficial for escaping from local minima and stagnation. To merge the advantages of transformed and original fitness on multimodal and unimodal functions, an evolution difficulty-based utilization mechanism has also been developed. DT has been incorporated into several baseline DEs and DT-based variants have been proposed.

Experiments on the test functions show that DT variants yield significant advancements when compared with the baselines. They also exhibit overall better performance than the other six state-of-the-art DEs, namely, EB-LSHADE, PaDE, EaDE, L-SHADE_cnEpSin, jSO, and L-SHADE-RSP. Particularly, the superiority on several functions is remarkable as it achieves the best-so-far DE performance.

The working mechanism of DTDE has also been studied with investigations on the population diversity and population successful update rate. According to the experiments, DT significantly improves the population's successful update rate and accelerates the population convergence. The rationale behind its capability of escaping from stagnation has also been explained.

The DT method has also been extended to noisy optimization. Experiments confirm its superiority over the baseline algorithm, the classic resampling method, which is widely adopted in noisy optimization literature, state-of-the-art DEs, as well as popular NEAs.

In the current study, the Fourier transform is served as the DT method. Further analysis of fitness landscape feature for selecting transformation methods in an ad hoc manner deserves studies. We will investigate other methods, such as the wavelet transform [90]. Besides, future works might also consider the derived fitness landscape feature with the assistance of DT for selecting appropriate strategies or algorithms. Finally, extending the DT method to other types of optimizations, such as multiobjective [91], multitasking [92], [93], and other types of EAs are also interesting directions.

References

[1] J. D. Ser et al., "Bio-inspired computation: Where we stand and what's next," *Swarm Evol. Comput.*, vol. 48, pp. 220–250, Aug. 2019.

[2] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.

[3] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.

[4] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, Apr. 2016.

[5] R. D. Al-Dabbagh, F. Neri, N. Idris, and M. S. Baba, "Algorithm design issues in adaptive differential evolution: Review and taxonomy," *Swarm Evol. Comput.*, vol. 43, pp. 284–311, Dec. 2018.

[6] K. R. Opara and J. Arabas, "Differential evolution: A survey of theoretical analyses," *Swarm Evol. Comput.*, vol. 44, pp. 546–558, Feb. 2019.

[7] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1430–1442, Mar. 2021.

[8] X.-G. Zhou, C.-X. Peng, J. Liu, Y. Zhang, and G.-J. Zhang, "Underestimation-assisted global-local cooperative differential evolution and the application to protein structure prediction," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 536–550, Jun. 2020.

[9] K. Michalak, "Low-dimensional Euclidean embedding for visualization of search spaces in combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 2, pp. 232–246, Apr. 2019.

[10] K. Yu, J. Liang, B. Qu, Y. Luo, and C. Yue, "Dynamic selection preference-assisted constrained multiobjective differential evolution," *IEEE Trans Syst., Man, Cybern., Syst.*, vol. 52, no. 5, pp. 2954–2965, May 2022.

[11] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.

[12] W. Gong, Á. Fialho, Z. Cai, and H. Li, "Adaptive strategy selection in differential evolution for numerical optimization: An empirical study," *Inf. Sci.*, vol. 181, no. 24, pp. 5364–5386, 2011.

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

[14] S. X. Zhang, S. Y. Zheng, and L. M. Zheng, "An efficient multiple variants coordination framework for differential evolution," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2780–2793, Sep. 2017.

[15] 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 Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Dec. 2006.

[16] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.

[17] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 71–78.

[18] L. Tang, Y. Dong, and J. Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 560–574, Aug. 2015.

[19] G. Sun, Y. Cai, T. Wang, H. Tian, C. Wang, and Y. Chen, "Differential evolution with individual-dependent topology adaptation," *Inf. Sci.*, vol. 450, pp. 1–38, Jun. 2018.

[20] S. X. Zhang, W. S. Chan, Z. K. Peng, S. Y. Zheng, and K. S. Tang, "Selective-candidate framework with similarity selection rule for evolutionary optimization," *Swarm Evol. Comput.*, vol. 56, Aug. 2020, Art. no. 100696.

[21] T. Jones and S. Forrest, "Fitness distance correlation as a measure of problem difficulty for genetic algorithms," in *Proc. Int. Conf. Genet. Algorithms (ICGA)*, vol. 95. Pittsburgh, PA, USA, 1995, pp. 184–192.

[22] M. Lunacek and D. Whitley, "The dispersion metric and the CMA evolution strategy," in *Proc. 8th Annu. Conf. Genet. Evol. Comput. (GECCO)*, Seattle WA, USA, 2006, pp. 477–484.

[23] M. Wang, B. Li, G. Zhang, and X. Yao, "Population evolvability: Dynamic fitness landscape analysis for population-based metaheuristic algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 550–563, Aug. 2018.

[24] H. Lu, J. Shi, Z. Fei, Q. Zhou, and K. Mao, "Measures in the time and frequency domains for fitness landscape analysis of dynamic optimization problems," *Appl. Soft Comput.*, vol. 51, pp. 192–208, Feb. 2017.

[25] L. Shen and J. He, "A mixed strategy for evolutionary programming based on local fitness landscape," in *Proc. IEEE Congr. Evol. Comput.*, Barcelona, Spain, 2010, pp. 1–8.

[26] Y. Huang, W. Li, C. Ouyang, and Y. Chen, "A self-feedback strategy differential evolution with fitness landscape analysis," *Soft Comput.*, vol. 22, pp. 7773–7785, Aug. 2018.

[27] W. Li, S. Li, Z. Chen, L. Zhong, and C. Ouyang, "Self-feedback differential evolution adapting to fitness landscape characteristics," *Soft Comput.*, vol. 23, no. 4, pp. 1151–1163, 2019.

[28] Z. Tan, K. Li, and Y. Wang, "Differential evolution with adaptive mutation strategy based on fitness landscape analysis," *Inf. Sci.*, vol. 549, pp. 142–163, Mar. 2021.

[29] K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Landscape-based adaptive operator selection mechanism for differential evolution," *Inf. Sci.*, vols. 418–419, pp. 383–404, Dec. 2017.

[30] M. Yang, C. Li, Z. Cai, and J. Guan, "Differential evolution with auto-enhanced population diversity," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 302–315, Feb. 2015.

[31] J. Lampinen and I. Zelinka, "On stagnation of the differential evolution algorithm," in *Proc. 6th Int. Mendel Conf. Soft Comput.*, Jun. 2000, pp. 76–83.

[32] S.-M. Guo, C.-C. Chang, P.-H. Hsu, and J. S.-H. Tsai, "Improving differential evolution with successful-parent-selecting framework," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 717–730, Oct. 2015.

[33] A. P. Piotrowski, "Differential evolution algorithms applied to neural network training suffer from stagnation," *Appl. Soft Comput.*, vol. 21, pp. 382–406, Aug. 2014.

[34] P. Rakshit, A. Konar, and S. Das, "Noisy evolutionary optimization algorithms—A comprehensive survey," *Swarm Evol. Comput.*, vol. 33, pp.18–45, Apr. 2017.

[35] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Ensemble strategies for population-based optimization algorithms—A survey," *Swarm Evol. Comput.*, vol. 44, pp. 695–711, Feb. 2019.

[36] R. Tanabe and A. Fukunaga, "Reviewing and benchmarking parameter control methods in differential evolution," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1170–1184, Mar. 2020.

[37] L. M. Zheng, S. X. Zhang, K. S. Tang, and S. Y. Zheng, "Differential evolution powered by collective information," *Inf. Sci.*, vol. 399, pp. 13–29, Aug. 2017.

[38] K. Qiao, J. Liang, B. Qu, K. Yu, C. Yue, and H. Song, "Differential evolution with level-based learning mechanism," *Complex Syst. Model. Simulat.*, vol. 2, no. 1, pp. 35–58, Mar. 2022.

[39] S. Das et al., "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

[40] R. Mallipeddi, P. Suganthan, Q. Pan, and M. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1679–1696, 2011.

[41] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 219–232, Jan. 2016.

[42] S. X. Zhang, W. S. Chan, K. S. Tang, and S. Y. Zheng, "Adaptive strategy in differential evolution via explicit exploitation and exploration controls," *Appl. Soft Comput.*, vol. 107, Aug. 2021, Art. no. 107494.

[43] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[44] W. Gong, A. Zhou, and Z. Cai, "A multi-operator search strategy based on cheap surrogate models for evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 746–758, Oct. 2015.

[45] X. Zhou and G. Zhang, "Differential evolution with underestimation-based multimutation strategy," *IEEE Trans. Cybern.*, vol. 49, no. 4, pp. 1353–1364, Apr. 2019.

[46] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proc. IEEE Congr. Evol. Comput.*, Beijing, China, 2014, pp. 1658–1665.

[47] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jSO," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 1311–1318.

[48] Y. Wang, H. X. Li, T. Huang, and L. Li, "Differential evolution based on covariance matrix learning and bimodal distribution parameter setting," *Appl. Soft Comput.*, vol. 18, pp. 232–247, May 2014.

[49] N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 372–379.

[50] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowl.-Based Syst.*, vol. 168, pp. 80–99, Mar. 2019.

[51] Z.-H. Zhan, Z.-J. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4633–4647, Nov. 2020.

[52] B. L. Miller, "Noise, sampling, and efficient genetic algorithms," Ph.D. dissertation, Dept. Comput. Sci., Univ. Illinois Urbana-Champaign, Champaign, IL, USA, 1997.

[53] L. Brevault, M. Balesdent, and J. Morio, *Aerospace System Analysis and Optimization in Uncertainty*. Cham, Switzerland: Springer Int., 2020.

[54] C. Qian, C. Bian, Y. Yu, K. Tang, and X. Yao, "Analysis of noisy evolutionary optimization when sampling fails," *Algorithmica*, vol. 83, pp. 940–975, Jan. 2021.

[55] C. Bian, C. Qian, Y. Yu, and K. Tang, "On the robustness of median sampling in noisy evolutionary optimization," *Sci. China Inf. Sci.*, vol. 64, no. 5, 2021, Art. no. 150103.

[56] F. Siegmund, A. H. C. Ng, and K. Deb, "Hybrid dynamic resampling for guided evolutionary multi-objective optimization," in *Proc. Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Guimarães, Portugal, 2015, pp. 366–380.

[57] A. D. Pietro, "Optimising evolutionary strategies for problems with varying noise strength," Ph.D. dissertation, Dept. Comput. Sci. Softw. Eng., Univ. Western Australia, Crawley WA, Australia, 2007.

[58] P. Rakshit, "Memory based self-adaptive sampling for noisy multi-objective optimization," *Inf. Sci.*, vol. 511, pp. 243–264, Feb. 2020.

[59] P. Rakshit, "Improved differential evolution for noisy optimization," *Swarm Evol. Comput.*, vol. 52, Feb. 2020, Art. no. 100628.

[60] A. Caponio and F. Neri, "Differential evolution with noise analyzer," in *Proc. Workshops Appl. Evol. Comput.*, Apr. 2009, pp. 715–724.

[61] P. Rakshit, A. Konar, S. Das, L. C. Jain, and A. K. Nagar, "Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 7, pp. 922–937, Jul. 2014.

[62] P. Rakshit and A. Konar, "Recent advances in evolutionary optimization in noisy environment—A comprehensive survey," in *Principles in Noisy Optimization* (Cognitive Intelligence and Robotics). Singapore: Springer, 2018.

[63] N. Hansen and A. Auger, "Principled design of continuous stochastic search: From theory to practice," in *Theory and Principled Methods for the Design of Metaheuristics*. Berlin, Germany: Springer, 2014, pp. 145–180.

[64] S. Rahnamayan, H. R. Tizhoosh, and M. M. Salama, "Opposition-based differential evolution for optimization of noisy problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, 2006, pp. 1865–1872.

[65] D. V. Arnold and H. G. Beyer, "Performance analysis of evolutionary optimization with cumulative step length adaptation," *IEEE Trans. Autom. Control.*, vol. 49, no. 4, pp. 617–622, Apr. 2004.

[66] J. Y. Chia, C. K. Goh, V. A. Shim, and K. C. Tan, "A data mining approach to evolutionary optimisation of noisy multi-objective problems," *Int. J. Syst. Sci.*, vol. 43, no. 7, pp. 1217–1247, Oct. 2012.

[67] E. Mendel, R. A. Krohling, and M. Campos, "Swarm algorithms with chaotic jumps applied to noisy optimization problems," *Inf. Sci.*, vol. 181, no. 20, pp. 4494–4514, Oct. 2011.

[68] L. T. Bui, H. A. Abbass, and D. Essam, "Localization for solving noisy multi-objective optimization problems," *Evol. Comput.*, vol. 17, no. 3, pp. 379–409, Sep. 2009.

[69] E. Mininno and F. Neri, "A memetic differential evolution approach in noisy optimization," *Memet. Comput.*, vol. 2, no. 2, pp. 111–135, 2010.

[70] J. Zhang, X. Zhu, Y. Wang, and M. Zhou, "Dual-environmental particle swarm optimizer in noisy and noise-free environments," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2011–2021, Jun. 2019.

[71] P. D. Stroud, "Kalman-extended genetic algorithm for search in nonstationary environments with noisy fitness evaluations," *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 66–77, Feb. 2001.

[72] J. E. Fieldsend and R. M. Everson, "The rolling tide evolutionary algorithm: A multiobjective optimizer for noisy optimization problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 103–117, Feb. 2015.

[73] V. A. Shim, K. C. Tan, J. Y. Chia, and A. A. Mamun, "Multi-objective optimization with estimation of distribution algorithm in a noisy environment," *Evol. Comput.*, vol. 21, no. 1, pp. 149–177, Mar. 2013.

[74] T. Smith, P. Husbands, and M. O'Shea, "Fitness landscapes and evolvability," *Evol. Comput.*, vol. 10, no. 1, pp. 1–34, 2002.

[75] K. M. Malan and A. P. Engelbrecht, "Quantifying ruggedness of continuous landscapes using entropy," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Trondheim, Norway, 2009, pp. 1440–1447.

[76] B. Bischl, O. Mersmann, H. Trautmann, and M. Preuß, "Algorithm selection based on exploratory landscape analysis and cost-sensitive learning," in *Proc. 14th Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2012, pp. 313–320.

[77] S. Verel, G. Ochoa, and M. Tomassini, "Local optima networks of NK landscapes with neutrality," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 783–797, Dec. 2011.

[78] L. Altenberg, "The evolution of evolvability in genetic programming," in *Advances in Genetic Programming*, K. Kinnear, Ed. Cambridge, MA, USA: MIT Press, 1994, ch. 3, pp. 47–74.

[79] O. Mersmann, M. Preuss, and H. Trautmann, "Benchmarking evolutionary algorithms: Towards exploratory landscape analysis," in *Proc. Int. Conf. Parallel Problem Solving Nat. (PPSN)*, 2010, pp. 73–82.

[80] O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph, "Exploratory landscape analysis," in *Proc. 13th Annu. Conf. Genet. Evol. Comput. (GECCO)*, Dublin, Ireland, 2011, pp. 829–836.

[81] J. Adair, G. Ochoa, and K. M. Malan, "Local optima networks for continuous fitness landscapes," in *Proc. Genet. Evol. Comput. Conf. Compan.*, 2019, pp. 1407–1414.

[82] K. R. Rao, D. N. Kim, and J. J. Hwang, *Fast Fourier Transform: Algorithms and Applications*, vol. 32. Dordrecht, The Netherlands: Springer, 2010.

[83] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, pp. 297–301, Apr. 1965.

[84] 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 bound constrained real-parameter numerical optimization," Nanyang Technol. Univ., Singapore, Rep., Dec. 2016.

[85] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*. Boca Raton, FL, USA: Chapman Hall/CRC, 2003.

[86] A. W. Mohamed, A. A. Hadi, and K. M. Jambi, "Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization," *Swarm Evol. Comput.*, vol. 50, Nov. 2019, Art. no. 100455.

[87] V. Stanovov, S. Akhmedova, and E. Semenkin, "LSHADE algorithm with rank-based selective pressure strategy for solving CEC 2017 benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Rio de Janeiro, Brazil, 2018, pp. 1–8.

[88] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

[89] S. Das and P. N. Suganthan, "Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems," Jadavpur Univ., Kolkata, India, School Electr. Electron. Eng., Nanyang Technol. Univ., Rep., 2010.

[90] C. Torrence and G. P. Compo, "A practical guide to wavelet analysis," *Bull. Amer. Meteorol. Soc.*, vol. 79, no. 1, pp. 61–78, 1998.

[91] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[92] Y. Feng, L. Feng, S. Kwong, and K. C. Tan, "A multivariation multifactorial evolutionary algorithm for large-scale multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 248–262, Apr. 2022.

[93] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, and C. Yue, "An evolutionary multitasking optimization framework for constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 263–277, Apr. 2022.

Sheng Xin Zhang was born in Guangdong, China. He received the B.S. degree in electronic science and technology and the M.Sc. degree in communication and information systems from Jinan University, Guangzhou, China, in 2013 and 2016, respectively, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 2020.

He is currently a Lecturer with the Department of Electronic Engineering, School of Information Science and Technology, Jinan University. His research interests include intelligent information processing, evolutionary computation, and the applications to microwave/millimeter-wave components.

Yi Nan Wen was born in Sichuan, China. He received the B.S. degree in electronic and information engineering from the Chengdu University of Information Technology, Chengdu, Sichuan, China, in 2020. He is currently pursuing the M.Sc. degree with Jinan University, Guangzhou, China.

His research interests include evolutionary algorithms, multiobjective optimization, and their applications.

Yu Hong Liu received the B.S. degree in communication engineering from the Guangdong University of Technology, Guangzhou, China, in 2016, and the M.Sc. degree in signal and information processing from Jinan University, Guangzhou, in 2021, where she is currently pursuing the Ph.D. degree.

Her research interests include evolutionary algorithms and multiobjective optimization in noisy environment.

Li Ming Zheng received the B.S. degree in radio electronics from South China Normal University, Guangzhou, China, in 1993, the M.Sc. degree in communication and information systems from Jinan University, Guangzhou, in 2000, and the Ph.D. degree in optics from South China Normal University in 2007.

He is currently a Professor with the Department of Electronic Engineering, School of Information Science and Technology, Jinan University. His research interests include intelligent information processing and microwave/millimeter-wave components.

Shao Yong Zheng (Senior Member, IEEE) was born in Fujian, China. He received the B.S. degree in electronic engineering from Xiamen University, Xiamen, Fujian, China, in 2003, and the M.Sc., M.Phil., and Ph.D. degrees in electronic engineering from the City University of Hong Kong, Hong Kong, in 2006, 2008, and 2011, respectively.

He is currently a Full Professor with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China, and the Deputy Director of the Mobile Communication National Engineering Research Center, SYSU Branch. He has published over 100 internationally refereed journal and conference papers, including 60 IEEE TRANSACTIONS papers. His research interests include microwave/millimeter-wave components and evolutionary algorithms.

Dr. Zheng is currently an Associate Editor of the IEEE ANTENNAS AND WIRELESS PROPAGATION LETTERS and *Microwave and Optical Technology Letters*.