IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 19, 2017, accepted November 9, 2017, date of publication November 15, 2017, date of current version December 22, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2773825

A Modified Differential Evolution With Distance-based Selection for Continuous Optimization in Presence of Noise

ARKA GHOSH[1,2], SWAGATAM DAS[3], RAMMOHAN MALLIPEDDI[4], ASIT KUMAR DAS[2], AND SUBHRANSU S. DASH[5]

[1]Indian Statistical Institute Kolkata, Kolkata 700108, India
[2]Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, India
[3]Electronics and Communication Sciences Unit, Indian Statistical Institute Kolkata, Kolkata 700108, India
[4]School of Electronics Engineering, Kyungpook National University, Daegu 702 701, South Korea
[5]Department of Electrical and Electronics Engineering, SRM University, Chennai 603203, India

Corresponding author: Rammohan Mallipeddi (mallipeddi.ram@gmail.com)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea by the Ministry of Education, Science and Technology under Grant NRF-2015R1C1A1A01055669.

ABSTRACT The performance of evolutionary algorithms (EAs), suitable for optimization on static functional landscapes, usually degrade in presence of noises with different statistical features. In this paper, we present a simple variant of the differential evolution (DE) algorithm, one of the most competitive EAs of recent interest, to tackle complex optimization problems in the presence of additive noise. The proposed DE variant is equipped with three new algorithmic components. A new population central tendency-based mutation scheme is proposed and it is switched in a probabilistic manner with the difference mean-based perturbation strategy in the mutation step. Instead of the regular binomial or exponential crossover of DE, we adopt a blending crossover during the recombination stage. Finally, a novel distance-based selection mechanism is incorporated to enable the occasional inclusion of a few inferior solutions to future generations, thus making the usual DE selection less greedy. Five different additive noise models namely Gaussian, Poisson, Rayleigh, Exponential, and Random are considered with a variety of noise amplitudes to simulate the noisy behavior of the objective functions. In total, 79 benchmark functions from traditional, as well as modern (IEEE CEC 2013 and 2017) test-suites, are used to extensively compare and contrast the proposed method with the other state-of-art evolutionary optimization algorithms, tailor-made for noisy function optimization. Experimental results, supported with the non-parametric statistical tests, indicate that our proposed method is very competitive against the noise-resilient variants of classical as well as very recent evolutionary optimizers, including the winners of the recent IEEE CEC competitions of real parameter optimization on the complex fitness landscapes.

INDEX TERMS Differential evolution, noisy optimization, selection, gaussian noise, poisson noise.

I. INTRODUCTION

Noise can be a disruptive factor for the real world and non-convex optimization problems for which Evolutionary Algorithms (EAs) are employed. Noise can be generated from the output of sensors, numerical simulators, actuators, and other measuring equipment of the decision variables and the objective (cost) function [1], [2]. Optimization under noisy scenarios occurs frequently in power systems [3], direct search policies in reinforcement learning [2], game theory [4] and so on.

For almost last two decades, researchers have been preferring EAs as a promising tool to tackle the noisy optimization problems due to their derivative-free nature and population-based search mechanism, see, for example, the recent survey by Rakshit *et al.* [2] and the references therein for a detailed account. If the objective function value is corrupted with noise, the normal evolutionary process can easily become unstable and successful search for the global optimum can be adversely affected [1]. This is more specifically true for EAs that employ a greedy selection operator which always prefer a fitter individual according to the cost function value, as due to additive noise in the cost function, an actually fitter individual may present a worse cost and get discarded. The reverse also can happen, i.e. a worse individual may be admitted to subsequent generations due to deceivingly better fitness appearance caused by noise.

In current EA literature, Differential Evolution (DE) [5]–[7] stands at a very competitive position among other state-of-the-art algorithms for optimization in continuous parameter (decision variable) spaces. Many DE variants have been proposed over the past decade for moderate to high dimensional function optimization. Some of the well known DE variants involve clever mutation and recombination strategies for offspring generation (like Pro-DE [8], MDE-pBX [9], DE with eigenvector based crossover [45], DE with multiple exponential crossover [48]) success-history based strategy and parameter adaption strategies (e.g., SaDE [10], JADE [11], SHADE [41], L-SHADE [42]), and combination of multiple offspring generation techniques (CoDE [12], EPSDE [13], DE with multiple variant coordination [50] etc.). DE has also been modified for solving continuous optimization problems corrupted with additive noise. A deterministic choice of the scale factor and the usual greedy selection method of DE adversely affect its performance on the noisy optimization problems [14]. Das and Konar [15] suggested two modifications to improve the performance of DE on noisy optimization problems, firstly by sampling the scale factor uniform at random between 0.5 and 1 and secondly by incorporating two not-so-greedy selection mechanisms (threshold-based selection and stochastic selection) in DE. Rahnamayan *et al.* [16] proposed a DE variant based on the concept of opposite numbers for superior search over noisy functions. A hybrid algorithm involving DE and Simulated Annealing (SA) was proposed as DEOSA [17] for solving noisy optimization problems. Mininno and Neri [18] presented a DE variant based on the noise analysis principle. This scheme comprised of three algorithmic tweaks to cope up with the uncertainties induced due to noise. The modifications included a controlled randomization of the scale factor and crossover rate, local search procedures to determine the scale factor, and an online statistical test that can ensure sampling at only the most important points of the feasible search region.

Resampling tries to estimate the 'true' fitness value of an individual by evaluating it repeatedly a number of times [2]. Resampling incurs higher computational burden, especially if the evaluation of the candidate solutions is expensive, which is typical for real-world applications. Iacca *et al.* [19] presented a compact DE with the mechanism of storing only a probabilistic representation of the population instead of the entire population and an automatic resampling process to optimize under noisy environments. A noise intensity estimation based Rolling Tide Evolutionary Algorithm (RTEA) was proposed in [20] for noisy multi-objective optimization problems. A biogeography-based optimization along with blended migration strategy was proposed in [21] for noisy constrained optimization problems. Noise compensation via sequential dynamic resampling based Genetic Algorithm (GA) for multi-objective optimization problems was presented in [22]. Chiu *et al.* [23] presented a DE with the 1.01n resampling, targeting noisy problems where the standard deviation of noise is as large as the variation of the fitness value. Rakshit and Konar [24] proposed a DE variant with local neighborhood based adaptive sample size selection for noisy multi-objective optimization.

It can be seen that to handle growing hardship of the real-world problems, DE has been modified in many different ways, but in most of the cases, the resulting DE variants have lost their simplicity, which is one of the main reasons for DE's popularity among the practitioners. The present work is driven by the quest to improve DE for noisy, multivariate, and multi-modal search problems by using simple parameter control techniques along with very simple and intuitively appealing modifications to the basic DE steps without necessitating serious computational overhead (likely to be caused by external achieves, rank-based parent selection schemes, additional local search schemes, statistical test on sampled points, keeping the long records of successful individuals etc.).

The proposed DE variant incorporates a population centrality based mutation strategy and couples it with the difference mean based mutation [25] in a stochastically switchable manner. Alongside it uses a blending crossover as the recombination strategy and a novel distance-based stochastic selection mechanism to tackle the noisy nature of the objective function by keeping the provision for occasionally accepting poorer offspring for upcoming generations. Note that a related work (by a group of the co-authors of this paper) has recently appeared as the conference article in [26]. However, the current paper is significantly extended and different from the conference article from the following aspects. In [26], two classical DE mutation strategies, namely the explorative DE/rand/1 and the more exploitative DE/best/1 were combined in a uniform randomly switched manner (with equal probabilities). However, in the current work, two completely different and non-conventional mutation schemes are used, of which the population centrality based scheme is first proposed here. The non-conventional mutation strategies are used to make the algorithm robust to noises of widely different statistical characteristics. The work of [26] coupled both blending and binomial crossovers in a switching manner, but the algorithm, developed here, makes use of only the blending crossover, owing to its greater flexibility towards different kinds of noise distributions. Finally, while [26] used an adaptive threshold-based selection, the current work employs a novel distance-based selection with reduced greediness. Moreover, the NRDE (Noise Resilient DE) algorithm in [26] was tested only on the Gaussian noise. But the current algorithm has been extensively tested on 21 standard benchmarks using 5 different and practical noise models. In addition, the algorithm is tested on the test-suites comprising of 28 and 30 functions respectively proposed for the IEEE CEC (Congress on Evolutionary Computation) 2013 and 2017 competitions on real parameter optimization. These 58 functions are much more complex with a shift of the optima, rotation of the search space coordinates (inducing more non-separability among the decision variables) and mixing of the properties of many basic functions (which are included in the first set of 21 functions). Noisy versions of these functions are simulated by adding standard Gaussian noise with each evaluation of the functions. As will be evident from our experimental results, the proposed algorithm performs significantly better than NRDE on a wide variety of test functions.

Rest of the paper is organized in the following way. Section II outlines the canonical DE algorithm in sufficient details. Section III describes the proposed DE method. Section IV presents and discusses the experimental results. Finally, the paper is concluded in Section V.

II. THE CANONICAL DE ALGORITHM

A standard DE algorithm starts with a set of candidate solutions, each of which is represented as a vector of real numbers. Components of the solution vectors are also called decision variables. The set of solutions is also called a population and the initial population is generated through random sampling from a uniform probability distribution within the prescribed bounds for each decision variable. Subsequently, the algorithm evolves this population by applying the variation (mutation followed by crossover or recombination) and selection operators in each generation (iteration). The generations are terminated when some predefined condition (like exhaustion of a prescribed maximum number of Function Evaluations (FEs)) is met. In typical DE literature, the population size is denoted by Np. Each iteration of DE is called a generation following the standard evolutionary computing terminology. A standard way to represent the i^{th} vector of the current generation G is $\vec{X}_{i,G}= [x_{1,i,G}, x_{2,i,G}, \ldots, x_{D,i,G}]^T$ where D is the dimensionality of the search space under consideration.

A. INITIALIZATION

From a coarse knowledge of the problem at hand, we first determine the bounds for the solution space as: vector of minimum bounds for each decision variable $\vec{X}_{min} = [x_{1,min}, x_{2,min}, \ldots, x_{D,min}]^T$ and vector of maximum bounds for each decision variable $\vec{X}_{max} = [x_{1,max}, x_{2,max}, \ldots, x_{D,max}]^T$. Then, j^{th} dimension of the i^{th} population member can be initialized as:

$$x_{i,j} = x_{j,min} + rand_{i,j} \times \left(x_{j,max} - x_{j,min}\right), \tag{1}$$

where $rand_{i,j}$ is a uniformly distributed random number from the range [0, 1] and it is freshly drawn for each ordered pair (i, j).

B. MUTATION

Any individual (say the i^{th}) of the current population is known as the target vector. The mutation in DE is not exactly nature inspired and differs markedly from the same operation in GA. During mutation, for each target vector, another vector from the same population (known as the base vector) is perturbed with the scaled difference vector(s) of the form $\left(\vec{X}_{r_1,G} - \vec{X}_{r_2,G}\right)$ (created from the current generation vectors) to produce a new vector, known as the mutant or donor vector. Usually, there is a scale factor F, lying between [0.4, 2], which scales the difference vector(s), thus controlling the perturbation step-size. The base vector can be a random one from current population, the best one (yielding the smallest objective function value for a minimization problem), a point on the line joining the current target and the best vectors, and so on. Depending on the nature of the base vector and number of difference vectors used for perturbation, many mutation strategies have been proposed in DE literature. Below we show two of the most commonly used strategies [6].

$$\text{DE/rand/1: } \vec{V}_{i,G} = \vec{X}_{r_1,G} + F.(\vec{X}_{r_2,G} - \vec{X}_{r_3,G}), \tag{2a}$$
$$\text{DE/best/1: } \vec{V}_{i,G} = \vec{X}_{best,G} + F.(\vec{X}_{r_1,G} - \vec{X}_{r_2,G}), \tag{2b}$$

where r_1, r_2, r_3 are randomly selected from $\{1,2,\ldots, N_{\text{p}}\}$, they are mutually exclusive and all of them are different from the current running index i. These indices are stochastically generated for each donor vector. $\vec{X}_{best,G}$ is the best-performing solution vector of the current generation and F is the scale factor already mentioned. The DE/rand/1 scheme is usually more explorative, which increases population diversity fast, whereas DE/best/1 promotes exploitation or intensification of the search around the best performing individual.

C. CROSSOVER/RECOMBINATION

The crossover step combines selected components of the target and donor vectors into a single vector, commonly known as the trial vector (final offspring) $U_{i,G} = [u_{1,i,G}, u_{2,i,G}, \ldots, u_{D,i,G}]^T$. Most commonly, DE uses the binomial crossover strategy [6], [7].

Binomial crossover involves fixing the value of a parameter called crossover rate (Cr) in the range [0, 1]. D independent numbers between 0 and 1 are sampled uniform at random and compared with Cr to decide which component is to be a part of the trial vector. The method can be expressed in the following way.

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & if\ rand_{i,j} \le Cr or j = j_r, \\ x_{j,i,g}, & otherwise, \end{cases} \tag{3}$$

where j_r is an integer index chosen randomly from $\{1, 2, \ldots, D\}$ and it makes sure that at least one component from the mutant vector passes on to the trial vector produced.

D. SELECTION

The one-to-one competition based selection process between the target and the trial vectors is performed as the final stage of a DE generation to maintain the population size constant.

The selection process can be outlined in the following way.

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_i, & if\ f\left(\vec{U}_{i,G}\right) \leq f\left(\vec{X}_{i,G}\right), \\ \vec{X}_{i,G}, & otherwise, \end{cases} \tag{4}$$

where $f(.)$ is the cost or objective function to be minimized. Therefore, if the trial vector gives equal or small objective function value compared to that of the target vector, the trial vector replaces the donor. The equality in "$\leq$" of (4) helps the DE population to navigate over the flat portions of a fitness landscape and to reduce the possibility of the population becoming stagnant. Note that the original DE selection, as outlined in (4) is quite greedy in nature, as it always allows a better or equivalent solution to enter into population, and does not make room for a worse individual which may be close to the boundary of an optimal basin of attraction (and this can be a common scenario for noisy optimization problems).

III. THE PROPOSED METHOD

Let us refer to the DE variant, proposed in this paper, as Modified DE with a Distance-based Selection (MDE-DS). MDE-DS introduces a population centrality based mutation strategy and couples it with the Difference Mean based Perturbation (DMP) [25] in a probabilistically switchable manner. It also uses the blending crossover as recombination strategy and a novel distance-based stochastic selection mechanism to tackle noisy nature of the objective function. MDE-DS does not contain any explicitly tuneable control parameter, except for the population size Np, which is kept constant in majority of the DE variants [7]. In what follows, we discuss the modifications proposed at different stages of the MDE-DS algorithm.

A. PARAMETER CONTROL

In MDE-DS there is no need to fix values for F and Cr, as F is randomly switched between 0.5 and 2 for each mutation operation and Cr is sampled uniform at random from the continuous interval [0.3, 1] for each target vector. Switching of F between two extreme corners of the feasible range is conducive to attain a balance between diversification and intensification of the search. The utility of such switching scheme has been earlier discussed by us in a recent work in the context of large scale static optimization [27]. In blending crossover, there is a parameter b (the blending rate), whose value is also randomly chosen from among three distinct values: a low value of 0.1, a medium value of 0.5 and a high value of 0.9.

B. MUTATION

In the population centrality based mutation, the entire population is sorted based on the objective function values and 50% best performing individuals are selected to design a temporary subpopulation of size $Np/2$. Now, we compute $\tilde{\vec{X}}_{best,G}$ as the arithmetic mean (centroid) of the subpopulation members. We mutate the i^{th} population member using the following expression:

$$\vec{V}_{i,G} = \vec{X}_{r1,G} + F\left(\tilde{\vec{X}}_{best,G} - \vec{X}_{r2,G}\right), \tag{5}$$

where $\vec{X}_{r1,G}$ and $\vec{X}_{r2,G}$ are two individuals corresponding to randomly chosen indices r_1 and r_2 and $\vec{V}_{i,G}$ is the newly generated mutant vector corresponding to current target vector for present generation G.

In the DMP-based mutation scheme, the best performing individual of the current generation ($\vec{X}_{best,G}$) is selected and dimension-wise average is taken for both $\vec{X}_{best,G}$ and the current target member $\vec{X}_{i,G}$. Now, a D-dimensional vector $\vec{M}_{i,G}$ is generated having each element within the range [0, 1]. Using the random directional vector $\vec{M}_{i,G}$, the mutant is generated in the following way:

$$\vec{V}_{i,G} = \vec{X}_{i,G} + \Delta_m \cdot \left(\frac{\vec{M}_{i,G}}{\left\|\vec{M}_{i,G}\right\|}\right), \tag{6}$$

where $\Delta_m = \left(X_{best_{dim},G} - X_{i_{dim},G}\right)$, with $X_{best_{dim},G} = \frac{1}{D}\sum_{k=1}^{D} x_{best_k,G}$ and $X_{i_{dim},G} = \frac{1}{D}\sum_{k=1}^{D} x_{i_k,G}$.

The significance of the population centrality based mutation scheme is that it preserves greediness while still maintaining some level of diversity, i.e. it is less greedy than the DE/best/1 scheme and hence there is less chance of getting stuck in the local optima. On the other hand, the DMP-based mutation scheme prefers exploration (see [25] for a detailed explanation), and thus, in absence of any feedback about nature of the function, we use an unbiased combination of these two methods. The population centrality based mutation scheme is illustrated in Fig. 1, which shows the top 50% of a sample DE population in 2-D decision space of the sphere function and the formation of a possible donor (mutant) point.

Each individual is perturbed either with the population centrality based mutation of with the DMP based mutation with equal probability. Note that the difference mean Δ_m, in (6) acts as a scaling coefficient for the randomized direction and provides high degree of explorative power while still maintaining attraction towards the currently best individual.

C. CROSSOVER

Crossover plays a crucial role in the generation of new promising search point from two or more existing points within the function landscape. We use a blending crossover in MDE-LS and it can be described in the following way:

$$u_{j,i,G} = \begin{cases} b.x_{j,i,G} + (1-b).v_{j,i,G}, & if\ rand_{i,j} \leq 0.5\ or\ j = j_r, \\ x_{j,i,G}, & otherwise \end{cases} \tag{7}$$

where $u_{j,i,G}$ and $v_{j,i,G}$ are the j^{th} dimensions of the trial and donor vectors respectively corresponding to current index i in generation G and $x_{j,i,G}$ is the j^{th} dimension of the current population member $\vec{X}_{i,G}$. Blending recombination has one

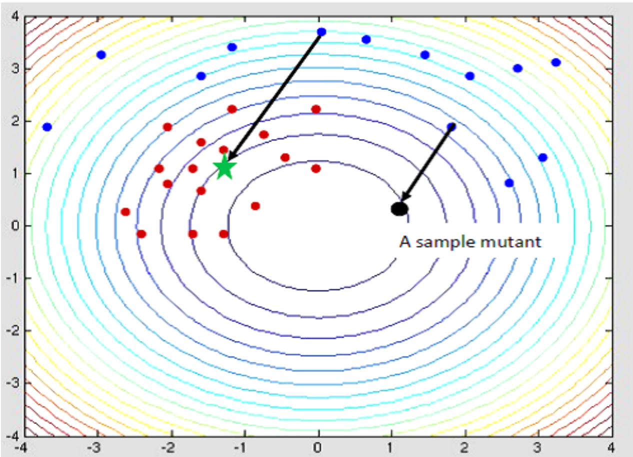


FIGURE 1. **Illustrating the population centrality based mutation scheme on a 2-D decision space with iso-contours of the sphere function.**

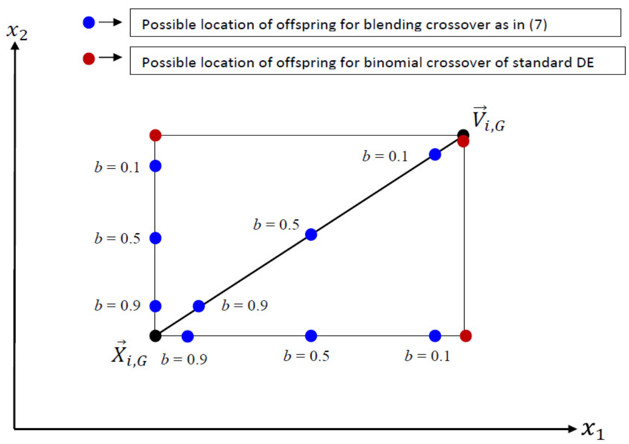


FIGURE 2. **Effect of blending crossover on a 2-D decision space.**

parameter b, which here can take three values: 0.1, 0.5, and 0.9. Reasons behind such selection of values are, when $b = 0.1$, the new point is generated near the donor vector, whereas $b = 0.9$ gives new point near the target vector and new point is at middle of the line joining the donor and the target points when b is set to 0.5. In absence of any prior knowledge or informative feedback from the functional landscape (and such feedback will obviously incur more computations), intuitively, a random switching of b over these three values for each crossover operation can be very effective, as was discussed by Das *et al.* in [28] for the two other crucial parameters of DE (F and Cr).

Idea of the blending crossover is illustrated for a two-dimensional decision variable space in Fig. 2, where we show possible location of the offspring for blending crossover as well as regular binomial crossover of DE. It is evident that if both the components become a convex combination of the corresponding components from $\vec{X}_{i,G}$ and $\vec{V}_{i,G}$ then the resulting offspring are formed on the diagonal connecting the target and the donor. These solutions are essentially rotationally invariant as even when the reference coordinate axes of the search space rotate, the principal diagonal of the rectangle formed with $\vec{X}_{i,G}$ and $\vec{V}_{i,G}$ at two opposite corners remain unchanged. However, Fig. 2 also shows that when one component is formed as a convex combination and the other is directly taken from $\vec{X}_{i,G}$ (i.e. for this one, $rand_{i,j} > 0.5$), the resulting offspring are axis-parallel for different values of b. Such offspring simulate coordinate-wise search which is effective for separable problems, whereas the rotationally invariant offspring are more effective for the rotated multi-modal landscapes. Thus, the blending crossover yields these two types of offspring resulting in more diversified search moves, instead of forming offspring only at the corners of the simplex formed with $\vec{X}_{i,G}$ and $\vec{V}_{i,G}$ as two opposite points. Note that Mukherjee *et al.* [29] recently showcased the efficacy of blending crossover for dynamic optimization problems (which are, although, different from noisy optimization dealt in the current context) and here the term "blending crossover" has been used in a sense completely different from the usual BLX crossover (see for example, the crossover strategy of [28]).

D. SELECTION

The general DE selection, as detailed in Section IVE, accepts an offspring into the population if it is no worse than its parent at the same population index. However, if the fitness landscape gets corrupted with noise, such greedy selection method suffers a lot because in this case the original fitness of parent and offspring are unknown and it can be well nigh impossible to infer when an offspring is absolutely superior or absolutely inferior to its parent. To handle this situation, a novel distance-based selection mechanism is introduced without any extra parameter (like the threshold value in a threshold-based selection [35]). There are three cases of the proposed selection mechanism which are described subsequently.

Case 1: If the offspring cost is equal or less than the parent cost, then offspring replaces the parent and survives to the next generation as:

$$\vec{X}_{i,G+1} = \vec{U}_{i,G}, \quad if\ \frac{f(\vec{U}_{i,G})}{f(\vec{X}_{i,G})} \leq 1.$$

Case 2: If the offspring cost is greater than parent cost, then offspring can replace the parent based on a stochastic principle. A probability is calculated by $e^{-\frac{\Delta f}{Dist}}$ where Δf is the absolute difference of objective function value between parent (target) and offspring (trial) and *Dist* is the Manhattan distance between those two vectors. We have used the Manhattan distance because of its simplicity and computational efficiency. The scheme can be outlined in the

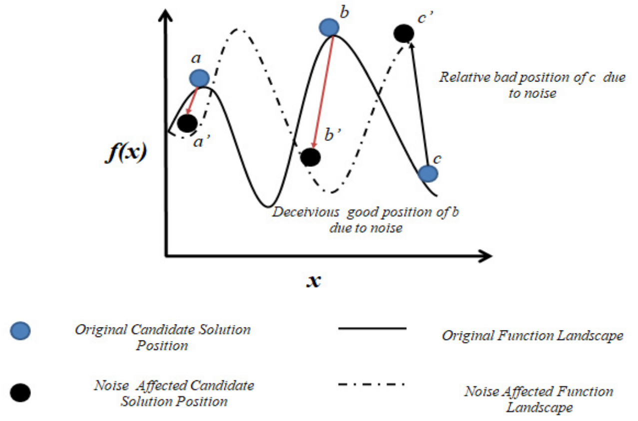


FIGURE 3. A sample fitness landscape before and after being affected by noise.

following way:

$$\vec{X}_{i,G+1} = \vec{U}_{i,G}, \quad if\ \frac{f(\vec{U}_{i,G})}{f(\vec{X}_{i,G})} > 1 \text{ and } p_s \le e^{-\frac{\Delta f}{Dist}},$$

where $\Delta f = \left| f\left(\vec{U}_{i,G}\right) - f(\vec{X}_{i,G}) \right|$ and $Dist = \sum_{k=1}^{D} \left| u_{i,k} - x_{i,k} \right|$. Here, p_s is a random number generated in the range [0, 1].

Case 3: If the offspring cost is highly greater than the parent cost, then offspring is removed and the parent persists to next generation.

$$\vec{X}_{i,G+1} = \vec{X}_{i,G}, \quad otherwise.$$

This selection process is further illustrated in Fig. 3. In Fig. 3, a sample function landscape scenario for any given instant is shown where a is the position of target member and b and c are the two donor members. Now selection among a, b and c should give us c as it has better objective function value than target member a. However, suppose this landscape is affected by noise, as shown in Fig. 3 as dotted line. In this noise affected scenario, if we use traditional greedy selection mechanism, b' will be selected for its decisive objective function value. In order to overcome this kind of situation, we propose the distance-based stochastic selection scheme, which gives us some probabilistic flexibility to select worse solutions as in noise affected landscapes; the original objective function value may become masked.

This selection mechanism is controlled in probabilistic manner otherwise it would affect convergence of the proposed algorithm. Maintaining diversity throughout entire search process is one of the most crucial aspects of any population-based optimization algorithm. Mutation is the key step to introduce diversity in the population during the search process. DE/rand/1 is one of the most popular DE mutation strategies, and it is well appreciated in the literature for its capability of making diverse solutions [6]. MDE-DS switches between two mutation strategies randomly to achieve a good balance between population diversity and convergence so that it can avoid local optima but does not overshoot and finds

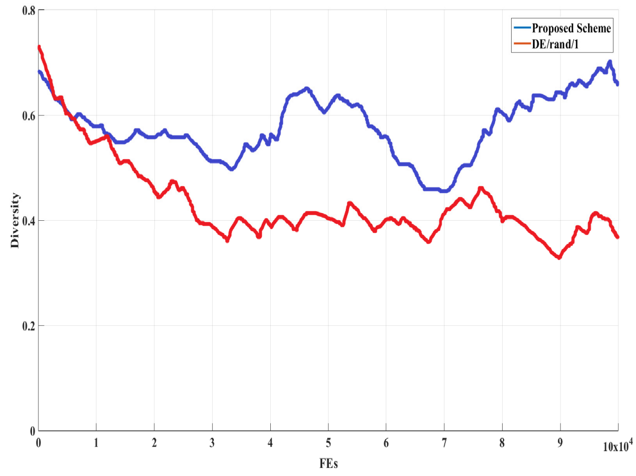


FIGURE 4. Population diversity comparison of MDE-DS and DE/rand/1/bin for Gaussian noise with amplitude 0.2, on function f_3 of set Bench I in Table AI (Appendix).

global optima successfully. Fig. 4 shows diversity comparison between MDE-DS and DE/rand./1/bin. Standard way to measure the diversity of a population P_G at generation G, is the average distance of the population from the mean vector, as proposed in [30], in the following way:

$$diversity(P_G) = \frac{1}{Np \times L} \sum_{i=1}^{Np} \sqrt{\sum_{j=1}^{D} (x_{j,i} - \bar{x}_j)^2},$$

where L is the length of the longest diagonal in the search space of dimension D and $\bar{x}_j = \frac{1}{Np} \sum_{i=1}^{Np} x_i$. is the average value of j^{th} dimension of the population members. Variation of population diversity with respect to the number of generations, for the population of MDE-LS (indicated by the blue line) and DE/rand/1/bin (shown by the red line and $F = 0.8$, $Cr = 0.9$) are plotted in Fig. 4. A complete pseudo-code of the proposal is provided as Algorithm 1.

IV. EXPERIMENTS & RESULTS

A. BENCHMARK SUITE & COMPUTATIONAL PROTOCOLS

The testing of any optimization algorithm on some well-known benchmark suites is an acceptable way to showcase the efficiency of the algorithm. These benchmark functions should have diverse characteristics, like multi-modality, non-separability, lack of global structure etc. In order to compare the performance of MDE-DS, a benchmark suite comprising of 21 well-known functions is used. Following the recommendations from existing literature [18], [37], 13 functions are grouped in a sub-suite called Bench-1 and the remaining 8 functions are grouped as Bench-2, based on the nature of the peer algorithms against which we made the comparative study. A summary of the 21 standard functions, divided into Bench-I and Bench-II sets, can be found in Table AI at the Appendix. Further to evaluate the capability of the proposed algorithm on more complex fitness landscapes, we

Algorithm 1 Pseudo Code of mde-ds

Procedure ***MDE-DS***
Create initial population by generating Np individuals uniform at random within given bounds for the decision variables;
Initialize FEs to 0, generation counter G to 1; Set value of maximum number of generations G_{max} as $MAXFEs/Np$ where $MAXFEs$ is the maximum budget of Function Evaluations (FEs).

    for $i = 1{:}Np$ **do**
        compute $f\left(\vec{X}_{i,0}\right)$;
    end for
    while $G \leq G_{max}$ **do**
        for $i = 1{:}Np$ **do**
            generate a random number $rand_m$ between 0 and 1; *%Mutation%*
            if $rand_m \leq 0.5$
                sort current population based on objective function values $f\left(\vec{X}_{i,}\right)$;
                select $Np/2$ best performing individuals from sorted population;
                compute $\tilde{\vec{X}}_{best,G} = \frac{2}{N_p}\sum_{k=1}^{\frac{Np}{2}} \tilde{\vec{X}}_{k,G}$;
                select two other members $\vec{X}_{r1,G}$ and $\vec{X}_{r2,G}$ from current population where $i \neq r_1 \neq r_2$;
                set value of F either to 0.5 or to 2 randomly;
                compute $\vec{V}_{i,G} = \vec{X}_{r1,G} + F(\tilde{\vec{X}}_{best,G} - \vec{X}_{r2,G})$;
            else
                select $\vec{X}_{best,G}$ which yields lowest objective function value for current iteration;
                select current population member $1\vec{X}_{i,G}$;
                compute $X_{best_{dim},G} = \frac{1}{D}\sum_{k=1}^{D} x_{best_k,G}$ and $X_{i_{dim},G} = \frac{1}{D}\sum_{k=1}^{D} x_{i_k,G}$;
                generate a D dimensional random vector, $\vec{M}_{i,G}$ with components in [0, 1];
                compute $\vec{V}_{i,G} = \vec{X}_{i,G} + (X_{best_{dim},G} - X_{i_{dim},G})(\frac{\vec{M}_{i,G}}{\left\|\vec{M}_{i,G}\right\|})$;
            end if
            sample the value ofCr from the interval [0.3, 1] uniform at random; *% Recombination %*
            select value for b randomly from the set from {0.1, 0.5, 0.9};
            initialize j_r to a random integer between 1 and D;
            for $j = 1{:}D$ **do**
                $u_{j,i,G} = \begin{cases} b.x_{j,i,G} + (1-b)\,.v_{j,i,G},\ if\, r\ and_{i,j} \leq CRorj = j_r \\ x_{j,i,G}, otherwise. \end{cases}$
            end for
            compute $f(\vec{U}_{i,G})$; *% Selection %*
            $FEs = FEs + 1$;
            compute $\Delta f_i = \left|f\left(\vec{U}_{i,G}\right) - f(\vec{X}_{i,G})\right|$;
            compute $Dist = \sum_{k=1}^{D}\left|u_{i,k} - x_{i,k}\right|$;
            generate a random number p_s between 0 and 1;
            $\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G},\ if\frac{f(\vec{U}_{i,G})}{f(\vec{X}_{i,G})} \leq 1, \\ \vec{U}_{i,G},\ if\frac{f(\vec{U}_{i,G})}{f(\vec{X}_{i,G})} > 1 and p_s \leq e^{-\frac{\Delta f_i}{Dist}}, \\ \vec{X}_{i,G}, otherwise; \end{cases}$
        end for
        $G{=}G + 1$
    end while

used all functions from the IEEE Congress on Evolutionary Computation (CEC) 2013 [38] and 2017 [39] test suites for competition on real parameter optimization in 50D and simulated their noisy behavior by adding Gaussian noise of zero mean and variance of 0.2, this being a standard noise model used in many of the previous works. Since these suites comprise respectively 28 and 30 functions each, we test MDE-DS extensively on 79 (28+30+21) functions in total.

IEEE CEC 2005 [31] benchmark suite for real parameter optimization contains a function (f_{17}) with noise corruption and the same is included in our tests also. For a detailed description of the IEEE CEC benchmark functions, refer to the respective technical reports [38], [39], [31].

Simulation is carried out in 30D and 100D for Bench-1 functions while for Bench-2, simulation is undertaken in 30D except for f_2, which is tested in 2D as it is non-scalable. For all test-suites, the results are reported in terms of the mean *best-of-the-run* error value over 30 independent runs over each noise strength. The error is measured as $|f_{best} - f^*|$, where f_{best} is the best function value returned by the algorithm in a run and f^* is the actual optimum value of the function. In the simulation process, a result less than or equal to 10^{-16} is rounded to 0.0.

In order to judge the statistical significance of our results with respect to the other competitors, the non-parametric Wilcoxon's rank sum test [32] is conducted for independent samples at 5% significance level. Throughout all the result tables, the statistical test results are summarized in the following way. If the set of errors yielded by an algorithm over 30 independent runs differ statistically significantly from that of the best performing algorithm on a particular function, then the mean error of the former is marked with a † symbol. If the difference of the error values found by one algorithm is not significant from the best algorithm, then the mean of the algorithm is marked with the sign ≈. The best performing algorithm in each case is marked with boldface.

A workstation equipped with Intel Xeon E5-2630 V3 processor running at 2.40 GHz and 32GB RAM is used to execute all simulations related to this work. The operating system used is Microsoft Window 10 and all the codes are written in MATLAB R2016a. Based on the mean error of functions given by different algorithms, a ranking for each algorithm is presented. In addition, for each group of problems (based on number of dimensions), an average rank for each competing algorithm is presented. Moreover, a Win/Tie/Loss analysis is provided in the result tables, where a win depicts that the algorithm is the sole best performer. If the best mean error is achieved by more than one method, than the tie count for all those methods is increased and the win count is kept unaltered. For every other case, a loss is declared.

B. NOISE MODELS

The original optimization problem is reformulated to incorporate noise in the following way:

$$f_{noisy}\left(\vec{X}\right) = f_{original}\left(\vec{X}\right) + \tau, \tag{8}$$

where $f_{original}\left(\vec{X}\right)$ *is* the original cost is function value corresponding to trial solution $\vec{X}$, $f_{noisy}\left(\vec{X}\right)$ is the noisy version of the cost function for the same trial solution $\vec{X}$ and τ represents the stochastic noise amplitude which is injected into the original cost function to emulate noise. This noise amplitude τ follows a certain Probability Distribution Function (PDF). The following five distributions of τ are considered here:

1) *Poisson*: τ follows a Poisson PDF, which is given by,

$$pdf\ (\tau) = \frac{\lambda^{\tau} \times e^{\tau}}{\tau!}, \tag{9}$$

where λ is the mean as well as variance of the Poisson distribution.

2) *Gaussian:* τ follows a Gaussian PDF, which is given by,

$$pdf\ (\tau) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\tau-m)^2}{2\sigma^2}}, \tag{10}$$

where m and σ^2 are mean and variance of the Gaussian PDF.

3) *Rayleigh*: τ follows a Rayleigh PDF, which is given by,

$$pdf\,(\tau) = \begin{cases} \dfrac{\tau}{\alpha^2}.e^{\frac{-\tau^2}{2\alpha^2}} & if\ \tau \geq 0, \\ 0 & otherwise, \end{cases} \tag{11}$$

where $m = \alpha\sqrt{\frac{\pi}{2}}$ is the mean and $\sigma = \alpha^2(4-\pi)/2$ variance of the noise distribution.

3) *Exponential*: τ follows an exponential PDF, which is given by,

$$pdf\,(\tau) = \begin{cases} ae^{(-a\tau)} & if\ \tau \geq 0, \\ 0 & otherwise, \end{cases} \tag{12}$$

where the mean of this distribution is $m = \frac{1}{a}$ and variance is $\sigma^2 = \frac{1}{a^2}$.

4) *Random*: In this case, τ is random noise which has maximum amplitude of ±25% of the original cost $f_{original}\left(\vec{X}\right)$.

Different parameter settings for the noise PDFs are summarized below. The parameters for Poisson, Rayleigh, and Exponential noise have been set and the noises have been injected into the objective function following [24] which, however, focuses on multi-objective noisy optimization only. Parameters of the Gaussian noise are standard and have been recommended in works like [18], [19], [26]. Uniform random noise, as such, has no parameter to set.

I. *Poisson Noise:* τ follows a Poisson PDF with mean and variance equal to 0.25.

II. *Gaussian Noise:* In Bench-1 τ follows a Gaussian PDF with zero mean and variance equal to 0.04, 0.1, and 0.2 and for Bench-2 τ follows a Gaussian PDF with zero mean and variance equal to 0.2, 0.4, 0.6, 0.8, and 1 following existing works [18], [37].

III. *Rayleigh Noise:* τ follows a Rayleigh PDF with mean equal to 0.3 and variance equal to 0.025.

IV. *Exponential Noise:* τ follows an exponential PDF with mean equal to 0.86 and variance equal to 0.75.

TABLE 1. Cost ± standard deviation for 30D Bench-1 problems for Gaussian noise.

Function	Noise Amplitude	DE/rand/1/bin	jDE	GADS	DERSFTS	OBDE	NADE	MUDE	MDE-DS
f_1	0.04	2.94e+00† 1.91e-01	1.13e-01† 2.44e-02	1.91e+00† 2.58e-01	1.92e-01† 3.51e-02	2.28e+00† 7.81e-01	6.96e-02† 1.90e-02	5.39e-02† 9.53e-03	**0.00e+00** **0.00e+00**
	0.1	3.53e+00† 9.64e-02	2.27e-01† 3.76e-02	1.89e+00† 2.60e-01	4.35e-01† 8.94e-02	2.89e+00† 5.85e-01	1.54e-01† 3.90e-02	1.35e-01† 4.22e-02	**0.00e+00** **0.00e+00**
	0.2	3.67e+00† 3.26e-02	4.59e-01† 7.41e-02	1.95e+00† 2.26e-01	1.10e+00† 4.77e-01	3.35e+00† 3.40e-01	2.99e-01† 8.88e-02	2.59e-01† 6.99e-02	**0.00e+00** **0.00e+00**
f_2	0.04	2.15e+01† 3.58e+00	7.09e+00† 1.96e+00	1.89e+01† 3.31e+00	1.48e+01† 2.75e+00	1.91e+01† 3.49e+00	3.42e+00† 1.34e+00	3.02e+00† 1.79e+00	**1.13e-03** **2.32e-01**
	0.1	3.02e+01† 4.77e+00	2.15e+01† 4.16e+00	2.09e+01† 3.73e+00	3.52e+01† 4.49e+00	3.39e+01† 8.94e+00	1.28e+01† 4.21e+00	1.25e+01† 4.42e+00	**3.64e-03** **1.24e-01**
	0.2	5.89e+01† 1.10e+01	4.25e+01† 7.54e+00	3.24e+01† 1.88e+01	5.67e+01† 9.79e+00	5.69e+01† 1.13e+01	3.29e+01† 6.70e+00	2.69e+01† 6.98e+00	**8.53e-03** **2.84e-02**
f_3	0.04	2.52e+03† 4.66e+02	8.94e+02† 2.23e+02	1.98e+03† 5.16e+02	1.36e+03† 3.31e+02	2.89e+03† 7.36e+02	6.22e+02† 2.14e+02	5.39e+02† 2.01e+02	**1.46e-04** **1.86e-05**
	0.1	3.87e+03† 1.06e+03	2.04e+03† 6.14e+02	2.16e+03† 4.78e+02	3.39e+03† 1.27e+03	5.70e+03† 1.60e+03	1.59e+03† 5.20e+02	1.02e+03† 2.94e+02	**4.02e-04** **2.98e-01**
	0.2	6.56e+03† 1.65e+03	4.27e+03† 1.08e+03	7.44e+03† 3.89e+03	6.84e+03† 2.21e+03	8.23e+03† 1.85e+03	3.32e+03† 1.17e+03	2.36e+03† 1.06e+03	**7.08e-04** **8.56e-05**
f_4	0.04	4.21e+01† 1.15e+01	1.23e+01† 2.63e+00	3.16e+01† 8.28e+00	2.30e+01† 7.32e+00	4.75e+01† 1.40e+01	8.55e+00† 2.84e+00	5.96e+00† 1.74e+00	**1.57e-02** **5.45e-06**
	0.1	6.44e+01† 1.69e+01	3.06e+01† 1.05e+01	4.48e+01† 2.09e+01	4.49e+01† 1.25e+01	8.89e+01† 2.51e+01	2.12e+01† 7.28e+00	1.61e+01† 4.50e+00	**4.20e-02** **6.25e-05**
	0.2	1.02e+02† 2.71e+01	5.82e+01† 1.82e+01	6.49e+01† 3.54e+01	1.06e+02† 3.61e+01	1.45e+02† 2.68e+01	4.53e+01† 2.14e+01	3.68e+01† 1.33e+01	**7.92e-02** **2.56e-06**
f_5	0.04	7.16e-02† 1.96e-02	7.80e-01† 6.62e-02	3.56e-02† 6.05e-03	8.17e-01† 5.69e-02	7.99e-02† 1.91e-02	8.43e-01† 7.21e-02	7.61e-01† 5.07e-02	**8.94e-04** **3.56e-05**
	0.1	5.24e-02† 1.86e-02	7.90e-01† 2.75e-02	3.79e-02† 1.29e-02	8.18e-01† 6.08e-02	7.19e-02† 2.47e-02	8.13e-01† 5.78e-02	7.63e-01† 6.96e-02	**2.53e-04** **4.78e-04**
	0.2	9.98e-03† 7.58e-03	7.67e-01† 6.55e-02	2.76e-02† 1.06e-02	7.84e-01† 5.88e-02	4.16e-02† 2.51e-02	8.30e-01† 6.87e-02	7.69e-01† 4.65e-02	**5.41e-04** **9.45e-08**
f_6	0.04	1.33e+02† 3.25e+01	4.53e+01† 1.29e+01	1.17e+02† 2.56e+01	7.87e+01† 2.16e+01	1.77e+02† 4.60e+01	3.65e+01† 1.33e+01	2.86e+01† 7.62e+00	**1.34e-03** **5.65e-01**
	0.1	2.38e+02† 5.07e+01	1.15e+02† 2.80e+01	1.53e+02† 3.96e+01	2.08e+02† 5.53e+01	3.48e+02† 8.17e+01	7.45e+01† 1.98e+01	7.49e+01† 2.42e+01	**1.08e-03** **2.65e-04**
	0.2	4.18e+02† 7.79e+01	1.99e+02† 4.28e+01	4.75e+02† 2.48e+02	3.95e+02† 1.37e+02	5.21e+02† 7.50e+01	1.65e+02† 6.29e+01	1.46e+02† 4.45e+01	**1.41e-03** **1.01e+00**
f_7	0.04	1.24e+01† 1.30e+00	2.78e+01† 4.65e-01	1.31e+01† 1.02e+00	2.58e+01† 8.04e-01	1.55e+01† 1.58e+00	2.69e+01† 7.07e-01	2.80e+01† 3.27e-01	**2.01e-01** **2.15e-01**
	0.1	1.09e+01† 1.50e+00	2.63e+01† 5.27e-01	1.35e+01† 9.14e-01	2.39e+01† 1.11e+00	1.45e+01† 1.65e+00	2.64e+01† 6.65e-01	2.69e+01† 6.32e-01	**2.04e+00** **2.45e+00**
	0.2	6.34e+00† 2.07e+00	2.08e+01† 2.04e+00	1.29e+01† 1.06e+00	8.52e+00† 2.30e+00	9.27e+00† 1.97e+00	2.46e+01† 1.35e+00	2.32e+01† 1.27e+00	**2.02e+00** **5.45e+00**
f_8	0.04	6.59e+03† 1.66e+03	1.82e+03† 5.69e+02	4.97e+03† 1.52e+03	3.16e+03† 8.14e+02	8.67e+03† 2.55e+03	1.42e+03† 5.62e+02	1.05e+03† 3.66e+02	**1.41e-01** **7.45e-02**
	0.1	8.60e+03† 2.46e+03	4.33e+03† 1.53e+03	5.96e+03† 1.59e+03	7.35e+03† 2.34e+03	1.37e+04† 3.33e+03	3.29e+03† 1.40e+03	2.27e+03† 8.58e+02	**3.63e-01** **2.14e-01**
	0.2	1.65e+04† 4.30e+03	9.19e+03† 2.06e+03	1.05e+04† 6.43e+03	1.66e+04† 4.53e+03	2.15e+04† 4.89e+03	6.84e+03† 1.79e+03	5.28e+03† 2.12e+03	**7.81e-01** **2.21e-01**
f_9	0.04	5.91e+00† 6.59e-01	6.86e+00† 1.67e+00	4.01e+00† 1.58e+00	6.49e+00† 2.35e+00	5.28e+00† 7.93e-01	6.42e+00† 1.43e+00	6.71e+00† 1.39e+00	**1.44e+00** **2.12e-01**
	0.1	5.84e+00† 5.33e-01	5.68e+00† 9.76e-01	3.60e+00† 2.01e+00	5.63e+00† 1.63e+00	4.64e+00† 7.38e-01	6.17e+00† 1.49e+00	6.58e+00† 1.37e+00	**1.42e+00** **2.54e-01**
	0.2	5.74e+00† 5.60e-01	4.65e+00† 1.31e+00	3.81e+00† 2.12e+00	3.92e+00† 1.40e+00	4.25e+00† 8.73e-01	5.05e+00† 1.16e+00	5.41e+00† 1.36e+00	**1.39e+00** **2.15e+00**
f_{10}	0.04	1.93e+02† 2.85e+01	5.31e+01† 9.75e+00	1.96e+02† 2.73e+01	8.23e+01† 1.39e+01	1.45e+02† 2.42e+01	3.73e+01† 8.15e+00	3.34e+01† 6.96e+00	**2.07e-03** **2.01e-04**
	0.1	2.20e+02† 3.30e+01	1.42e+02† 2.51e+01	2.03e+02† 2.52e+01	2.22e+02† 3.33e+01	2.05e+02† 3.44e+01	6.99e+01† 2.07e+01	9.43e+01† 2.45e+01	**2.35e-02** **2.47e+00**
	0.2	3.91e+02† 6.50e+01	2.90e+02† 3.82e+01	2.31e+02† 6.27e+01	3.83e+02† 5.63e+01	3.78e+02† 4.66e+01	1.96e+02† 5.00e+01	2.00e+02† 4.30e+01	**1.89e-02** **2.34e-01**
f_{11}	0.04	1.73e+03† 5.70e+02	8.49e+02† 2.25e+02	1.15e+03† 3.31e+02	1.34e+03† 5.51e+02	2.36e+03† 7.04e+02	7.23e+02† 3.05e+02	7.19e+02† 3.84e+02	**5.66e+00** **2.35e+00**
	0.1	3.16e+03† 1.05e+03	1.78e+03† 7.01e+02	3.01e+03† 1.51e+03	3.03e+03† 1.15e+03	4.73e+03† 1.41e+03	1.75e+03† 8.31e+02	1.46e+03† 8.20e+02	**2.06e+01** **6.52e+00**
	0.2	6.36e+03† 2.09e+03	4.12e+03† 1.68e+03	7.50e+03† 2.12e+03	6.12e+03† 3.03e+03	7.26e+03† 2.18e+03	3.76e+03† 2.28e+03	2.49e+03† 1.71e+03	**2.60e+01** **2.32e+00**
f_{12}	0.04	6.48e+03† 3.64e+02	1.26e+03† 4.17e+02	5.19e+03† 6.01e+02	3.04e+03† 4.47e+02	6.36e+03† 3.77e+02	2.29e+03† 5.32e+02	1.07e+03† 3.50e+02	**0.00e+00** **0.00e+00**
	0.1	7.05e+03† 2.17e+02	3.97e+03† 8.44e+02	5.73e+03† 5.21e+02	6.43e+03† 9.07e+02	6.96e+03† 3.34e+02	2.94e+03† 5.91e+02	2.59e+03† 7.05e+02	**0.00e+00** **0.00e+00**
	0.2	7.89e+03† 5.77e+02	8.22e+03† 9.54e+02	6.09e+03† 1.03e+03	1.01e+04† 8.28e+02	8.03e+03† 5.42e+02	5.60e+03† 8.80e+02	6.00e+03† 9.99e+02	**0.00e+00** **0.00e+00**
f_{13}	0.04	2.03e+00† 1.11e-01	2.29e+00† 4.40e-02	**1.99e+00** **1.03e-01**	2.19e+00† 7.22e-02	2.14e+00† 9.83e-02	2.36e+00† 4.00e-02	2.38e+00† 4.10e-02	1.10e+01† 2.35e+00
	0.1	1.81e+00† 1.77e-01	1.92e+00† 1.32e-01	1.88e+00† 1.09e-01	**1.53e+00** **2.00e-01**	1.89e+00† 1.67e-01	2.29e+00† 7.30e-02	2.25e+00† 8.53e-02	1.07e+01† 3.56e-01
	0.2	1.18e+00† 3.02e-01	9.91e-01† 3.19e-01	1.46e+00† 5.39e-01	**5.89e-01** **3.01e-01**	1.08e+00† 2.99e-01	1.82e+00† 2.52e-01	1.57e+00† 2.80e-01	1.03e+01† 4.23e+00
W/T/L		0/0/39	0/0/39	1/0/38	2/0/37	0/0/39	0/0/39	0/0/39	**36/0/3**
Avg .Rank		5.85	4.54	4.70	6.23	6.20	4.18	3.54	**1.54**

TABLE 2A. Cost ± standard deviation for 100D Bench-1 problems for Gaussian noise.

Function	Noise Amplitude	DE/rand/1/bin	jDE	GADS	DERSFTS	OBDE	NADE	MUDE	NRDE	MDE-DS
f_1	0.04	3.56e+00† 1.07e-01	2.14e-01† 6.12e-02	2.94e+00† 8.29e-02	4.58e-01† 9.38e-02	3.29e+00† 4.95e-01	8.69e-02† 2.05e-02	8.82e-02† 2.45e-02	**0.00e+00** **0.00e+00**	**0.00e+00** **0.00e+00**
	0.1	3.62e+00† 8.38e-03	4.40e-01† 1.08e-01	2.20e+00† 1.23e-01	8.13e-01† 1.41e-01	3.59e+00† 2.06e-01	1.63e-01† 4.56e-02	2.17e-01† 6.29e-02	**0.00e+00** **0.00e+00**	**0.00e+00** **0.00e+00**
	0.2	3.67e+00† 1.18e-02	8.13e-01† 1.31e-01	3.01e+00† 8.21e-02	1.34e+00† 1.81e-01	3.61e+00† 2.08e-02	3.65e-01† 1.04e-01	3.35e-01† 7.49e-02	**0.00e+00** **0.00e+00**	**0.00e+00** **0.00e+00**
f_2	0.04	1.18e+02† 1.25e+01	2.53e+01† 4.00e+00	1.19e+02† 8.91e+00	5.36e+01† 6.73e+00	1.07e+02† 1.14e+01	1.83e+01† 3.14e+00	8.60e+00† 1.91e+00	4.05e-01† 1.63-01	**1.62e-02** **2.21e-02**
	0.1	1.28e+02† 1.24e+01	7.59e+01† 8.56e+00	1.45e+02† 1.39e+01	1.05e+02† 1.39e+01	1.21e+02† 1.24e+01	3.56e+01† 6.36e+00	2.52e+01† 5.13e+00	5.14e-01† 3.98e-01	**2.33e-01** **2.24e-03**
	0.2	1.71e+02† 1.86e+01	1.36e+02† 1.54e+01	1.52e+02† 1.43e+01	1.64e+02† 1.86e+01	1.99e+02† 2.44e+01	7.82e+01† 1.55e+01	6.18e+01† 9.87e+00	1.25e+00† 6.36e-01	**6.46e-01** **5.21e+00**
f_3	0.04	4.72e+04† 6.28e+03	1.26e+04† 2.20e+03	6.24e+04† 7.01e+03	1.83e+04† 3.33e+03	4.79e+04† 6.94e+03	6.24e+03† 1.18e+03	6.29e+03† 1.14e+03	2.97e-01† 2.58e-01	**7.35e-02** **2.35e-05**
	0.1	5.59e+04† 7.41e+03	2.32e+04† 4.65e+03	6.44e+04† 6.53e+03	2.92e+04† 4.67e+03	7.64e+04† 9.24e+03	1.53e+04† 4.23e+03	1.31e+04† 2.51e+03	1.11e+00† 1.47e+00	**3.34e-01** **6.54e-01**
	0.2	7.89e+04† 8.59e+03	4.17e+04† 6.64e+03	8.35e+04† 2.67e+04	4.98e+04† 1.22e+04	1.02e+05† 1.35e+04	3.15e+04† 7.64e+03	2.61e+04† 7.26e+03	2.61e+00† 2.79e+00	**4.41e-01** **8.48e-01**
f_4	0.04	2.51e+02† 3.11e+01	5.97e+01† 8.40e+00	3.54e+02† 3.54e+01	9.99e+01† 1.16e+01	2.54e+02† 3.11e+01	2.81e+01† 7.04e+00	2.79e+01† 5.66e+00	1.67e-01† 1.73e-01	**1.45e-02** **1.45e-02**
	0.1	2.98e+02† 4.41e+01	1.05e+02† 1.56e+01	3.68e+02† 3.31e+01	1.53e+02† 2.50e+01	3.22e+02† 5.93e+01	5.97e+01† 1.67e+01	5.88e+01† 1.16e+01	2.17e-01† 1.51e-01	**3.72e-02** **5.45e-01**
	0.2	4.03e+02† 6.03e+01	2.01e+02† 2.95e+01	3.87e+02† 3.90e+01	2.35e+02† 4.45e+01	5.59e+02† 7.46e+01	1.24e+02† 3.30e+01	1.06e+02† 2.16e+01	5.91e-01† 5.03e-01	**6.68e-02** **1.24e-02**
f_5	0.04	1.25e-02† 2.24e-03	1.50e-01† 3.76e-02	6.09e-03† 7.76e-04	2.66e-01† 8.22e-02	1.23e-02† 2.30e-03	3.63e-01† 7.52e-02	1.71e-01† 4.13e-02	1.03e-03† 1.22e-02	**7.31e-04** **6.45e-01**
	0.1	8.63e-03† 2.47e-03	1.45e-01† 3.78e-02	5.52e-03† 8.76e-04	2.63e-01† 8.80e-02	9.58e-03† 3.00e-03	3.92e-01† 9.54e-02	1.61e-01† 3.25e-02	2.55e-03† 2.38e-02	**3.47e-04** **2.31e-04**
	0.2	3.29e-03† 1.74e-03	1.44e-01† 4.27e-02	4.85e-03† 5.85e-04	2.26e-01† 6.38e-02	5.79e-03† 8.33e-04	3.75e-01† 8.11e-02	1.70e-01† 4.36e-02	1.54e-03† 6.03e-02	**4.82e-04** **2.45e+00**
f_6	0.04	8.65e+02† 1.63e+02	2.04e+02† 3.04e+01	1.25e+03† 9.32e+01	3.94e+02† 5.77e+01	9.38e+02† 1.09e+02	9.13e+01† 1.74e+01	9.58e+01† 1.86e+01	1.14e+00† 1.26e-01	**2.63e-02** **2.20e-07**
	0.1	1.08e+03† 1.17e+02	4.15e+02† 8.48e+01	1.28e+03† 1.13e+02	5.32e+02† 1.00e+02	1.26e+03† 2.23e+02	2.19e+02† 4.27e+01	1.97e+02† 4.05e+01	1.53e+00† 3.41e-01	**5.51e-02** **1.45e-01**
	0.2	1.48e+03† 1.91e+02	7.29e+02† 1.06e+02	1.43e+03† 2.36e+02	8.43e+02† 2.09e+02	2.00e+03† 2.19e+02	4.50e+02† 1.55e+02	4.00e+02† 1.05e+02	1.76e+00† 7.00e-01	**7.73e-02** **4.54e-01**
f_7	0.04	1.87e+01† 2.60e+00	7.91e+01† 2.26e+00	2.04e+01† 1.84e+00	6.53e+01† 2.88e+00	2.49e+01† 2.25e+00	7.04e+01† 2.91e+00	8.15e+01† 1.86e+00	7.21e+00† 1.19e+00	**2.21e-01** **2.31e+00**
	0.1	1.75e+01† 1.61e+00	7.62e+01† 3.40e+00	1.95e+01† 1.70e+00	5.97e+01† 3.78e+00	2.38e+01† 2.57e+00	6.93e+01† 3.36e+00	7.95e+01† 2.51e+00	6.88e+00† 1.37e+00	**2.12e-01** **1.35e+00**
	0.2	1.04e+01† 1.99e+00	5.81e+01† 3.41e+00	1.80e+01† 1.12e+00	2.98e+01† 5.34e+00	1.85e+01† 2.77e+00	6.53e+01† 3.43e+00	7.20e+01† 2.22e+00	7.10e+00† 1.34e+00	**1.24e+00** **2.14e-01**
f_8	0.04	1.29e+05† 1.51e+04	3.11e+04† 5.23e+03	1.79e+05† 1.68e+04	4.89e+04† 6.33e+03	1.26e+05† 2.10e+04	1.46e+04† 3.62e+03	1.47e+04† 2.83e+03	8.22e-01† 2.19e+00	**2.32e-02** **1.37e-01**
	0.1	1.68e+05† 2.58e+04	5.93e+04† 1.00e+04	1.51e+05† 2.20e+04	7.75e+04† 1.10e+04	1.73e+05† 2.69e+04	3.53e+04† 9.32e+03	2.85e+04† 6.83e+03	4.29e-01† 5.19e-01	**3.25e-02** **1.45e+00**
	0.2	2.18e+05† 3.05e+04	1.00e+05† 1.56e+04	2.05e+05† 4.21e+04	1.36e+05† 3.55e+04	2.92e+05† 4.01e+04	6.60e+04† 1.89e+04	6.11e+04† 1.82e+04	1.09e+00† 1.53e+00	**3.69e-02** **1.45e+00**
f_9	0.04	1.21e+01† 1.92e+00	1.57e+01† 2.85e+00	2.56e+01† 3.99e+00	1.44e+01† 3.55e+00	8.46e+00† 2.51e+00	1.43e+01† 3.70e+00	1.56e+01† 2.79e+00	4.93e+01† 7.30e-02	**1.31e+00** **2.31e-01**
	0.1	1.31e+01† 1.40e+00	1.32e+01† 2.35e+00	2.47e+01† 3.51e+00	1.56e+01† 3.45e+00	8.05e+00† 2.58e+00	1.34e+01† 2.36e+00	1.51e+01† 2.75e+00	4.90e+01† 2.14e-02	**5.21e+00** **3.25e+00**
	0.2	1.53e+01† 2.08e+00	1.04e+01† 3.18e+00	2.48e+01† 3.89e+00	9.37e+00† 3.91e+00	7.65e+00† 2.98e+00	1.29e+01† 2.03e+00	1.28e+01† 2.26e+00	4.85e+01† 5.78e-02	**6.22e+00** **1.21e+00**
f_{10}	0.04	9.01e+02† 6.32e+01	2.39e+02† 2.65e+01	1.13e+03† 5.42e+01	3.97e+02† 3.64e+01	8.20e+02† 7.65e+01	2.72e+02† 3.07e+01	1.20e+02† 2.35e+01	3.14e-02† 2.26e-02	**1.75e-02** **1.21e-02**
	0.1	1.01e+03† 6.71e+01	5.04e+02† 4.86e+01	1.13e+03† 3.98e+01	7.08e+02† 8.50e+01	9.13e+02† 9.23e+01	3.98e+02† 4.05e+01	2.74e+02† 3.31e+01	6.90e-02† 9.52e-02	**2.45e-02** **1.25e-02**
	0.2	1.21e+03† 1.02e+02	1.08e+03† 7.29e+01	1.15e+03† 6.02e+01	1.18e+03† 7.04e+01	1.34e+03† 1.43e+02	5.01e+02† 8.09e+01	5.80e+02† 8.04e+01	1.35e-01† 1.09e-01	**1.54e-02** **4.75e-03**
f_{11}	0.04	8.91e+03† 1.50e+03	3.19e+03† 6.36e+02	1.20e+04† 1.59e+03	4.02e+03† 6.72e+02	9.88e+03† 1.66e+03	2.27e+03† 5.95e+02	1.91e+03† 5.00e+02	9.60e+01† 2.12e+00	**5.42e+01** **2.54e+00**
	0.1	1.33e+04† 2.31e+03	6.00e+03† 1.47e+03	1.28e+04† 1.89e+03	7.23e+03† 2.13e+03	1.67e+04† 3.21e+03	4.34e+03† 1.63e+03	3.93e+03† 1.57e+03	1.01e+02† 1.72e+01	**6.45e+01** **1.02e+00**
	0.2	2.03e+04† 3.06e+03	1.00e+04† 2.52e+03	2.28e+04† 1.01e+04	1.21e+04† 3.78e+03	2.83e+04† 5.26e+03	8.18e+03† 3.20e+03	7.16e+03† 2.46e+03	1.07e+02† 2.16e+01	**9.45e+01** **2.78e+00**
f_{12}	0.04	2.36e+04† 2.21e+02	1.34e+04† 9.57e+02	2.79e+04† 8.15e+02	1.65e+04† 1.37e+03	2.38e+04† 2.94e+02	1.29e+04† 1.01e+03	8.19e+03† 8.53e+02	**0.00e+00** **0.00e+00**	**0.00e+00** **0.00e+00**
	0.1	2.49e+04† 3.35e+02	1.71e+04† 1.32e+03	2.70e+04† 1.03e+03	2.54e+04† 2.06e+03	2.44e+04† 3.22e+02	1.80e+04† 1.15e+03	1.19e+04† 1.41e+03	**0.00e+00** **0.00e+00**	**0.00e+00** **0.00e+00**
	0.2	2.68e+04† 1.17e+03	3.97e+04† 1.81e+03	2.88e+04† 1.07e+03	3.18e+04† 1.35e+03	2.78e+04† 1.19e+03	1.96e+04† 1.64e+03	2.06e+04† 1.74e+03	**0.00e+00** **0.00e+00**	**0.00e+00** **0.00e+00**
f_{13}	0.04	1.61e+00† 9.41e-02	2.14e+00† 4.18e-02	**1.23e+00** **8.07e-02**	1.94e+00† 5.87e-02	1.78e+00† 1.05e-01	2.16e+00† 6.34e-02	2.07e+00† 3.77e-02	2.09e+01† 3.78e+00	1.52e+01† 2.58e-02
	0.1	1.45e+00† 1.09e-01	1.74e+00† 1.14e-01	**1.23e+00** **1.13e-01**	1.33e+00† 1.84e-01	1.53e+00† 1.56e-01	2.02e+00† 7.71e-02	2.16e+00† 1.01e-01	1.91e+01† 2.42e+00	1.68e+01† 2.56e+00
	0.2	1.03e+00† 2.02e-01	6.82e-01† 1.54e-01	1.13e+00† 1.74e-01	**4.61e-01** **1.76e-01**	8.49e-01† 1.76e-01	1.78e+00† 1.12e-01	1.57e+00† 1.90e-01	1.97e+01† 2.46e+00	1.86e+01† 5.46e-01
W/T/L		0/0/39	0/0/39	2/0/37	1/0/38	0/0/39	0/0/39	0/0/39	**0/6/33**	**30/6/3**
Avg Rank		6.33	5.33	6.84	5.92	6.89	4.89	4.41	**2.87**	**1.38**

TABLE 2B. Cost ± standard deviation for 100D Bench-1 problems for Poisson, Rayleigh, exponential and random noise.

Function	Noise Amplitude	DE/rand/1 /bin with the proposed selection scheme	DERSFTS	NADE	MUDE	MDE-DS
f_1	Poisson	2.95e+00† 2.07e-08	1.22e-01† 1.28e-02	4.23e+00† 1.10e+00	2.45e+00† 2.48e-01	**0.00e+00** **0.00e+00**
	Rayleigh	2.12e+00† 2.38e-07	1.15e-01† 2.41e-01	2.45e+00† 1.54e+00	1.46e+00† 2.78e-05	**0.00e+00** **0.00e+00**
	Exponential	1.57e+00† 2.18e-01	2.22e+00† 2.81e-02	5.65e+00† 1.45e+00	2.75e+00† 1.45e-01	**0.00e+00** **0.00e+00**
	Random	2.33e+01† 3.73e+00	3.10e+00† 1.35e-06	1.45e+00† 2.56e+00	2.54e+00† 1.54e+00	**0.00e+00** **0.00e+00**
f_2	Poisson	1.01e+01† 2.25e+00	6.36e+01† 6.73e+00	1.56e+02† 2.35e+01	1.74e+01† 2.35e+00	**5.56e-05** **2.12e-09**
	Rayleigh	1.11e+01† 1.14e+01	8.05e+02† 5.39e+00	1.86e+02† 8.65e+00	4.99e+01† 2.35e-02	**8.53e-08** **2.65e-04**
	Exponential	2.71e+01† 3.86e+00	2.64e+02† 1.86e+00	2.17e+02† 1.48e+00	2.75e+01† 1.54e-01	**1.14e-02** **2.36e-05**
	Random	4.51e+01† 4.77e-01	2.64e+02† 1.86e+00	1.72e+02† 2.35e+00	3.21e+01† 1.25e+00	**6.08e-02** **1.28e-08**
f_3	Poisson	3.52e+04† 6.28e+02	2.83e+04† 3.33e+03	4.77e+05† 1.02e+02	6.37e+00† 1.24e+00	**1.96e-05** **2.35e-06**
	Rayleigh	1.59e+04† 4.41e+01	3.02e+04† 2.17e+02	7.45e+05† 8.45e+00	1.07e+03† 2.56e+00	**7.55e-06** **2.36e-02**
	Exponential	2.89e+04† 2.59e+01	3.85e+04† 1.02e+01	7.75e+05† 2.45e+01	1.24e+03† 2.32e+01	**3.97e-04** **2.65e-02**
	Random	1.22e+04† 8.59e+00	5.12e+04† 1.22e+01	9.92e+05† 2.35e+01	2.37e+02† 1.24e-04	**9.50e-03** **2.87e-07**
f_4	Poisson	1.21e+01† 2.11e+00	8.89e+01† 2.16e+02	1.69e+02† 2.22e+00	2.79e-01† 1.24e-02	**5.20e-03** **2.35e-08**
	Rayleigh	3.98e+01† 7.41e+00	2.23e+02† 2.50e+00	4.10e+02† 2.12e+00	1.25e+00† 8.64e-00	**2.50e-02** **3.65e-05**
	Exponential	5.03e+01† 5.03e+00	2.85e+02† 4.45e+01	4.44e+02† 2.32e+01	3.47e+00† 1.54e+00	**1.37e-03** **2.68e-07**
	Random	9.74e+01† 1.16e+01	2.39e+02† 4.45e+01	5.25e+02† 2.56e+01	1.45e+00† 2.35e-03	**5.40e-03** **7.65e-04**
f_5	Poisson	1.75e-04† 4.24e-03	2.57e-01† 8.52e-02	5.85e-01† 1.25e-01	6.26e-01† 2.56e+00	**0.00e+00** **0.00e+00**
	Rayleigh	5.63e-04† 1.47e-03	2.23e-01† 8.50e-02	3.92e+02† 1.25e+01	6.02e-01† 4.58e-03	**0.00e+00** **0.00e+00**
	Exponential	2.29e-04† 8.74e-03	3.25e-01† 6.38e-02	2.24e-02† 2.65e-01	6.44e-02† 2.54e-03	**0.00e+00** **0.00e+00**
	Random	7.89e-03† 7.17e+01	2.85e-02† 2.55e-01	1.03e-03† 1.20e+00	7.68e-02† 2.98e-04	**0.00e+00** **0.00e+00**
f_6	Poisson	2.65e+01† 1.63e+02	3.54e+02† 5.87e+01	6.98e+02† 5.65e+01	9.50e+00† 2.32e-02	**3.72e+00** **1.24e-05**
	Rayleigh	1.00e+01† 1.12e+02	5.12e+02† 1.50e+01	1.17e+01† 5.64e+00	3.61e+00† 8.45e-01	**2.18e+00** **5.12e-09**
	Exponential	1.58e+02† 5.91e-03	5.43e+02† 2.55e+01	1.54e+03† 8.65e+00	1.26e+01† 3.01e+00	**3.94e+00** **2.36e-06**
	Random	1.32e+02† 1.00e+01	1.21e+03† 2.55e+01	1.61e+03† 2.35e+01	8.21e+00† 4.54e-01	**2.55e+00** **2.92e-01**
f_7	Poisson	9.87e+01† 1.60e+00	5.52e+01† 3.88e+00	1.95e+01† 2.35e+00	4.26e+01† 1.23e-02	**4.38e+00** **1.25e-01**
	Rayleigh	8.75e+01† 5.61e+00	6.55e+01† 2.78e+00	1.31e+03† 5.65e+01	3.74e+01† 1.24e+00	**4.73e+00** **2.32e+01**
	Exponential	4.04e+01† 5.99e+00	3.85e+01† 5.24e+00	1.89e+01† 5.64e+00	3.65e+01† 1.25e-01	**4.23e+00** **2.25e-02**
	Random	3.98e+01† 2.34e+00	5.22e+01† 5.45e+00	2.42e+01† 5.65e+00	1.45e+01† 2.12e-02	**1.92e+00** **2.45e-04**
f_8	Poisson	1.49e+01† 1.51e+00	4.71e+04† 5.23e+01	4.35e+04† 2.65e+00	1.04e+00† 2.65e-05	**4.13e-05** **4.65e-08**
	Rayleigh	1.58e+01† 2.28e+00	9.45e+04† 1.50e+02	2.13e+01† 2.45e+00	7.44e+01† 2.11e+00	**1.23e-05** **6.23e-02**
	Exponential	2.12e+01† 2.01e+00	3.89e+05† 3.55e+02	8.02e+04† 8.45e+01	4.54e+01† 2.65e+00	**4.07e-02** **2.36e-05**
	Random	1.32e+01† 1.55e+00	1.96e+05† 5.37e+01	1.06e+05† 5.32e+02	8.65e+01† 2.88e-02	**2.76e-05** **1.32e-03**
f_9	Poisson	1.51e+02† 2.92e+00	5.49e+01† 3.75e+00	4.80e+01† 2.35e+01	4.90e+01† 1.89e+00	**4.22e+01** **1.56e+00**
	Rayleigh	1.31e+02† 1.40e+00	4.89e+01† 3.15e+00	8.68e+04† 2.45e+01	4.84e+01† 1.24e+00	**4.55e+01** **2.32e+00**
	Exponential	1.52e+02† 3.08e+00	8.45e+01† 3.11e+00	4.99e+01† 5.54e+01	4.97e+01† 2.54e+00	**4.95e+01** **2.25e-02**
	Random	1.56e+02† 5.45e+00	6.65e+01† 2.58e+00	3.75e+01† 2.35e+00	**3.59e+01** **1.54e+00**	3.71e+01† 1.65e-02
f_{10}	Poisson	8.01e+02† 2.32e+01	3.27e+02† 3.24e+01	1.23e+03† 1.22e+00	3.28e+02† 2.33e+00	**1.66e+02** **2.35e+00**
	Rayleigh	5.01e+02† 5.71e+01	8.08e+02† 1.50e+01	4.95e+01† 2.45e+00	5.89e+02† 3.54e+00	**1.53e+01** **5.65e+00**
	Exponential	3.21e+03† 8.02e+00	5.18e+03† 2.04e+01	1.49e+03† 5.24e+00	5.17e+02† 1.24e+00	**1.80e+02** **2.36e+00**
	Random	5.18e+03† 7.88e+00	**6.85e+01** **2.41e+00**	1.15e+03† 2.35e+00	7.65e+02† 1.28e+00	1.47e+02† 1.54e+00
f_{11}	Poisson	7.91e+03† 4.50e+02	6.82e+03† 4.72e+01	1.68e+04† 1.25e+00	1.05e+02† 2.35e-02	**8.43e+01** **1.25e+00**
	Rayleigh	2.33e+03† 2.55e+01	8.23e+03† 4.13e+02	1.45e+03† 8.54e+02	6.42e+02† 1.23e+01	**7.80e+01** **2.32e-02**
	Exponential	1.01e+03† 5.06e+02	2.21e+04† 3.88e+03	3.84e+04† 2.45e+00	3.49e+02† 1.84e+00	**9.51e+01** **3.25e-02**
	Random	3.21e+03† 2.78e+03	**4.74e+01** **2.35e+00**	3.52e+04† 2.35e+00	3.25e+02† 5.65e+00	7.05e+01† 1.45e+00
f_{12}	Poisson	2.32e+01† 1.21e+02	2.85e+04† 1.37e+03	2.35e-01† 1.86e-02	8.39e-02† 2.65e-06	**0.00e+00** **0.00e+00**
	Rayleigh	2.21e+01† 3.35e+02	3.44e+04† 3.06e+03	3.12e+03† 2.65e+02	7.01e+03† 2.32e+01	**0.00e+00** **0.00e+00**
	Exponential	3.68e+01† 1.17e+03	8.18e+04† 2.35e+02	2.25e+06† 1.35e+00	7.52e+03† 1.25e+00	**0.00e+00** **0.00e+00**
	Random	1.18e+01† 1.25e+02	4.75e+01† 2.25e+00	1.25e+08† 2.35e+00	5.65e+02† 2.35e-02	**0.00e+00** **0.00e+00**
f_{13}	Poisson	3.61e+01† 1.21e-02	2.94e+01† 5.87e-02	3.80e+01† 5.45e+01	3.52e+01† 2.36e-01	**1.90e+01** **2.39e-05**
	Rayleigh	4.45e+01† 1.08e-01	2.33e+01† 1.85e-01	3.36e+01† 8.45e-01	3.89e+01† 1.32e+00	**1.88e+01** **2.45e-02**
	Exponential	3.03e+01† 2.52e-01	4.61e+01† 1.75e-01	3.15e+01† 8.74e+00	3.95e+01† 1.84e+00	**1.95e+01** **8.65e+00**
	Random	3.32e+01† 1.13e-01	3.22e+01† 2.55e+00	3.99e+01† 2.35e+00	4.55e+01† 1.24e-05	**1.78e+01** **1.29e-01**
W/T/L		0/0/52	2/0/50	0/0/52	1/0/51	**49/0/3**
Avg Rank		3.35	3.77	3.94	2.88	**1.06**

C. COMPARED ALGORITHMS

The proposed MDE-DS is tested and compared with other state-of-the-art algorithms present in literature. The peer algorithms considered in this comparative study for the benchmark suite shown in Table AI are mentioned below.

I. A standard DE/rand/1/bin algorithm with proposed selection mechanism from Section III, here the scale factor is $F = 0.8$ and the cross over rate $Cr = 0.5$.

II. An adaptive DE variant namely jDE with parameter settings proposed in [33]

III. A Durations Sizing Genetic Algorithm (DSGA), a Genetic Algorithm variant for noisy optimization proposed in [34]. All the control parameters are set as per [34].

IV. The DE with Randomized Scale Factor and Threshold based Selection (DERSFTS) as presented in [35]. The scale factor used in this algorithm is randomized and can vary between 0.5 and 1 and cross over rate $Cr = 0.3$. This method employs a threshold based selection mechanism which is depends on a constant value.

V. The Opposition-Based DE (OBDE), with exact parameter settings as presented in [16]. The scale factor and crossover rate are set to $F = 0.7$ and $Cr = 0.3$ respectively and the jump factor is taken as 0.3.

VI. The Noise Analysis DE (NADE) proposed in [36].

VII. The Memetic DE for Noisy Optimization (MUDE), proposed in [18].

VIII. Different Particle Swarm Optimization (PSO) techniques, namely Global Best PSO (PSOgbest), Local Best PSO (PSOlbest), Bare Bone PSO (BBPSO), and Fully Informed PSO (FIPS) equipped with a Chaotic Jump (CJ) mechanism to handle noisy optimization function, as proposed in [37]. All parameters are set following [37].

IX. A Covariance Matrix Adaptation (CMA) based algorithm namely, a Restart CMA Evolution Strategy with Increasing Population Size (IPOP-CMA-ES) as proposed in [38].

X. Noise Resilient DE proposed in [26] (compared only on the $100D$ problems from the Bench - 1 set of Table AI to save space).

Apart from these, for the noisy version of the CEC 2013 functions, we specifically use the Success History based Adaptive DE (SHADE) [41] and a subsequent variant of SHADE with linear population size reduction, called L-SHADE [42]. We also consider the variants of SHADE and L-SHADE using the distance-based selection, proposed here (Section III.D) and mark them as SHADE-DS and LSHADE-DS in the respective result table. Note that SHADE was the top ranking DE variant in the IEEE CEC 2013 competition on real parameter optimization whereas, L-SHADE was the overall winner of the same competition organized under IEEE CEC 2014 conference.

TABLE 3. Cost ± standard deviation for Bench-2 problems.

Function	Noise Amplitude	PSOgbest+CJ	PSOlbest+CJ	BBPSO+CJ	FIPS+CJ	MDE-LS
f_1	0.2	6.18e-01† 7.83e-02	6.63e-01† 7.32e-02	6.25e-01† 6.75e-02	6.29e-01† 7.05e-02	**0.00e+00** **0.00e+00**
	0.4	1.23e+00† 1.64e-01	1.36e+00† 1.24e-01	1.25e+00† 1.46e-01	1.22e+00† 1.24e-01	**0.00e+00** **0.00e+00**
	0.6	1.92e+00† 2.38e-01	2.04e+00† 2.37e-01	1.83e+00† 2.63e-01	1.87e+00† 1.82e-01	**0.00e+00** **0.00e+00**
	0.8	2.56e+00† 2.83e-01	2.68e+00† 3.13e-01	2.54e+00† 3.57e-01	2.51e+00† 2.81e-01	**0.00e+00** **0.00e+00**
	1.0	3.08e+00† 3.48e-01	3.23e+00† 2.87e-01	3.25e+00† 4.20e-1	3.11e+00† 3.24e-01	**0.00e+00** **0.00e+00**
f_2	0.2	6.99e-01† 4.12e-02	7.57e-01† 7.41e-02	7.28e-01† 7.23e-02	6.70e-01† 5.93e-02	**9.04e-02** **2.15e-01**
	0.4	1.26e+00† 1.36e-01	1.34e+00† 1.40e-01	1.28e+00† 1.33e-01	1.23e+00† 1.34e-01	**7.24e-01** **2.54e-02**
	0.6	2.02e+00† 1.72e-01	2.05e+00† 1.78e-01	2.03e+00† 1.81e-01	2.01e+00† 1.65e-01	**4.55e-01** **5.84e-02**
	0.8	2.83e+00† 2.65e-01	2.91e+00† 2.63e-01	2.89e+00† 2.37e-01	2.78e+00† 2.05e-01	**3.47e-01** **1.54e-01**
	1.0	3.69e+00† 3.62e-01	3.67e+00† 2.97e-01	3.54e+00† 2.91e-01	3.63e+00† 3.06e-01	**2.24e-01** **2.54e-02**
f_3	0.2	5.80e-01† 7.59e-02	6.35e-01† 8.32e-02	5.86e-01† 8.77e-02	6.21e-01† 8.15e-02	**7.66e-02** **1.45e-02**
	0.4	1.13e+00† 1.45e-01	1.23e+00† 1.44e-01	1.15e+00† 1.60e-01	1.22e+00† 1.62e-01	**1.47e-02** **5.04e-01**
	0.6	1.74e+00† 3.19e-01	1.84e+00† 2.35e-01	1.75e+00† 2.48e-01	1.89e+00† 1.93e-01	**1.72e-02** **1.46e-01**
	0.8	2.21e+00† 3.88e-01	2.20e+00† 3.94e-01	2.26e+00† 3.43e-01	2.34e+00† 3.45e-01	**2.46e-02** **7.75e-02**
	1.0	2.56e+00† 4.35e-01	2.53e+00† 6.25e-01	2.76e+00† 4.72e-01	3.05e+00† 3.78e-01	**1.51e-02** **1.55e+00**
f_4	0.2	2.75e+01† 2.37e-01	6.28e+01† 5.97e+01	5.13e+01† 6.59e+01	2.79e+01† 1.69e-01	**8.65e-02** **1.25e-03**
	0.4	2.71e+01† 4.01e-01	7.95e+01† 9.67e-01	4.14e+01† 2.90e+01	2.76e+01† 1.69e-01	**1.51e+00** **2.24e+00**
	0.6	2.70e+01† 2.37e-01	5.34e+01† 6.80e+01	7.33e+01† 1.22e+02	2.71e+01† 1.87e-01	**2.66e+00** **1.42e-01**
	0.8	2.61e+01† 3.07e-01	6.86e+01† 5.73e+01	7.18e+01† 9.23e+01	2.64e+01† 2.28e-01	**3.00e+00** **1.24e-01**
	1.0	2.58e+01† 4.62e-01	6.56e+01† 5.52e+01	7.96e+01† 1.29e+02	2.57e+01† 3.96e-01	**3.11e+00** **7.89e+00**
f_5	0.2	6.27e-01† 6.49e-02	7.09e-01† 7.15e-02	6.23e-01† 8.73e-02	6.77e-01† 6.71e-02	**4.34e-01** **2.01e-01**
	0.4	1.39e+00† 1.51e-01	1.45e+00† 1.52e-01	1.27e+00† 1.54e-01	1.33e+00† 1.37e-01	**1.01e+00** **1.54e-01**
	0.6	1.88e+00† 2.16e-01	2.13e+00† 1.73e-01	1.86e+00† 2.37e-01	2.10e+00† 2.15e-01	**1.18e+00** **1.24e-01**
	0.8	2.58e+00† 2.60e-01	2.90e+00† 2.54e-01	2.61e+00† 4.21e-01	2.72e+00† 2.83e-01	**1.77e+00** **1.28e-01**
	1.0	3.23e+00† 3.72e-01	3.67e+00† 2.99e-01	3.21e+00† 5.10e-01	3.33e+00† 3.79e-01	**2.21e+00** **4.58e-01**
f_6	0.2	1.28e-01† 3.89e-01	2.74e-01† 2.30e-01	2.89e-01† 3.65e-01	2.69e-01† 2.42e-01	**2.54e-02** **4.51e-05**
	0.4	2.81e-01† 1.75e-01	2.77e-01† 1.522e-01	3.57e-01† 4.22e-01	2.90e-01† 1.28e-01	**1.65e-02** **2.22e-01**
	0.6	8.81e-01† 2.56e-01	9.47e-01† 2.31e-01	8.80e-01† 3.88e-01	8.85e-01† 1.93e-01	**4.52e-01** **1.20e-01**
	0.8	1.57e+00† 3.02e-01	1.49e+00† 3.03e-01	1.56e+00† 3.24e-01	1.53e+00† 2.23e-01	**2.16e-01** **1.42e-02**
	1.0	2.21e+00† 3.78e-01	2.28e+00† 3.64e-01	2.19e+00† 3.64e-01	2.23e+00† 3.03e-01	**2.06e-01** **1.46e-01**
f_7	0.2	3.03e+00† 1.74e+00	1.83e+00† 1.60e+00	1.63e+00† 1.65e+00	4.25e-01† 4.28e-01	**4.02e-02** **1.28e-02**
	0.4	2.93e+00† 2.45e+00	2.14e+00† 1.77e+00	1.90e+00† 1.36e+00	**7.95e-01** **5.34e-01**	9.12e-01† 1.54e+00
	0.6	2.73e+00† 1.53e+00	1.72e+00† 1.63e+00	1.68e+00† 1.49e+00	**1.07e+00** **8.02e-01**	2.37e+00† 1.87e-01
	0.8	2.25e+00† 1.96e+00	1.64e+00† 1.76e+00	1.91e+00† 1.82e+00	1.65e+00† 8.85e-01	**1.00e+00** **8.98e-05**
	1.0	2.35e+00† 2.10e+00	1.20e+00† 2.01e+00	1.69e+00† 1.75e+00	2.09e+00† 8.17e-01	**9.47e-01** **2.54e+00**
f_8	0.2	1.72e+03† 3.81e+02	1.93e+03† 2.06e+02	4.23e+02† 3.29e+02	2.87e+03† 2.05e+00	**0.00e+00** **0.00e+00**
	0.4	1.73e+03† 3.61e+02	1.95e+03† 2.06e+02	4.12e+02† 3.36e+02	2.89e+03† 2.00e+02	**0.00e+00** **0.00e+00**
	0.6	1.65e+03† 3.57e+02	1.96e+03† 3.22e+02	2.76e+02† 2.46e+02	2.93e+03† 2.08e+02	**0.00e+00** **0.00e+00**
	0.8	1.74e+03† 3.57e+02	1.92e+03† 2.31e+02	3.59e+02† 4.28e+02	2.86e+03† 2.70+02	**0.00e+00** **0.00e+00**
	1.0	1.67e+03† 3.42e+02	1.95e+03† 2.16e+02	4.25e+02† 3.76e+02	2.91e+03† 2.05e+02	**0.00e+00** **0.00e+00**
W/T/L		0/0/40	0/0/40	0/0/40	2/0/38	**38/0/2**
Avg. Rank		3.15	4.15	3.23	3.78	**1.10**

Similarly for the IEEE CEC 2017 benchmarks with additive noise, we consider two top ranking DE-variants called jSO (an improved version for the iL-SHADE [43] with a weighted mutation strategy) [44] and L-SHADE-cnEpSin (ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood) [45] along with their counterparts equipped with the distance-based selection process. Note that jSO and L-SHADE-cnEpSin respectively occupied the 1st and 2nd ranks in the IEEE CEC 2017 competition on real parameter single-objective optimization.

TABLE 4. Cost ± standard deviation for 30D & 50D f_{17} from IEEE CEC 2005 suite.

Function	Dimension	IPOP-CMA-ES	MDE-DS
f_{17}	30	3.01e+02† 1.74e+02	**1.03e+02** **7.59e-04**
	50	2.56e+02† 1.48e+02	**1.10e+02** **4.87e-04**

D. SIMULATION RESULTS ON 30D BENCH-1 PROBLEMS

Bench-1 consists of thirteen well-known numerical functions. Three different noise amplitudes (0.04, 0.1, and 0.2) of Gaussian noise are introduced in this simulation, for the 13 benchmark problems in 30D. Results of the proposed MDE-DS is compared with DE/rand/1/bin with proposed distance based selection, jDE, GADS, DERSFTS, OBDE, NADE, and MUDE for all the 3 noise amplitudes. The maximum number of Function Evaluations (FEs) for 30D Bench-1 problems are limited to 10^5.

Table 1 summarizes the simulation results for 30D Bench-1 problems. We can see that out of the total 39 test cases (13 problems × 3 scenarios), MDE-DS statistically outperforms all other competitors in 36 cases and only for f_{13} (the Tirronen function) it loses. GADS gives best result in f_{13} with noise amplitude 0.04, and DERSFTS gives best result for the said function in other two noise amplitudes. Tirronen function is highly multi-modal in nature and there exists very small gap between two consequent peaks in the functional landscape of this function. This may have led to selection of inappropriate solutions (unable to enter into the narrow basin subsequently) to next generation by the distance-based selection scheme of MDE-DS. MUDE is the closest contender of MDE-DS as per average ranking.

E. SIMULATION RESULTS ON 100D BENCH-1 PROBLEMS

To save space, we consider showing results with all the five different distributions of the noise on 100D functions of Bench-1. For the Gaussian noise model, three noise amplitudes (0.04, 0.1, and 0.2) are introduced again. Other models are Poisson, Rayleigh, exponential, and random noise. Total 50 runs of each problem for each noise model and corresponding to all the algorithms are considered and mean and standard deviation of the absolute *best-of-the-run* error are reported. Maximum number of FEs for 100D Bench-1 problems are limited to $3e{+}05$ following recommendations from existing literature.

Table 2A reports the simulation results for 100D Bench-1 problems for the Gaussian noise model. Out of total 39 test cases, MDE-DS gives the best result for 36 problems. In f_{13} with noise amplitude 0.04 and 0.1, GADS outperform others and for 0.2 noise amplitude of the same problem, DERSFTS gives the best result. For all other problems, MUDE is a close contender of MDE-DS after NRDE. According to the average ranking, MDE-DS is the winner among all other algorithms followed by MUDE and NADE. Consistency of the results over different noise strengths

TABLE 5. Cost ± standard deviation for 50D CEC 2013 problems for Gaussian noise with 0.2 noise strength.

	Algorithms												
Func	DE/rand/1/bin	jDE	GADS	DERSFTS	OBDE	NADE	MUDE	NRDE	SHADE	L-SHADE	SHADE-DS	L-SHDAE-DS	MDE-DS
F1	3.56e-10† 0.00e+00	3.65e-06† 2.32e-01	2.93e-09† 0.00e+00	3.25e-13† 0.00e+00	2.56e-06† 2.12e-02	2.28e-13† 0.00e+00	1.12e-13† 0.00e+00	2.32e-13† 2.32e-15	1.26e-12† 1.23e-01	5.65e-13† 2.36e-03	1.16e-15† 1.53e-02	4.15e-13† 2.26e-03	**0.00e+00** **0.00e+00**
F2	3.25e+09† 1.56e+06	6.32e+06† 3.26e+04	1.89e+08† 1.56e+07	3.25e+07† 3.16e+01	2.64e+07† 3.14e+06	7.59e+07† 2.31e+07	6.56e+07† 2.36e+01	2.36e+04† 1.35e+02	2.65e+04† 1.12e+04	3.12e+05† 1.11e+03	2.22e+04† 1.10e+04	3.10e+05† 1.10e+03	**2.36e+01** **2.36e-02**
F3	3.96e+11† 2.65e+02	4.00e+06† 2.32e-01	3.65e+10† 2.65e+10	2.12e+08† 3.02e+08	1.85e+08† 3.02e+02	2.66e+08† 1.46e+08	2.56e+08† 1.65e+07	2.35e+03† 2.36e+01	8.45e+05† 1.96e+06	2.36e+05† 2.36e+05	8.00e+05† 1.10e+06	2.06e+05† 2.00e+05	**3.85e+02** **1.25e+01**
F4	6.35e+06† 3.23e+02	2.63e+02† 3.23e+01	3.65e+05† 6.36e+04	2.11e+05† 2.95e+04	1.13e-01† 0.00e+00	2.25e+05† 9.99e+03	2.36e+05† 1.25e+04	1.26e+03† 2.36e+01	1.96e-03† 1.65e-03	2.36e-06† 1.23e-02	1.50e-03† 1.00e-03	**2.16e-06** **1.20e-02**	2.96e+02† 1.43e+03
F5	1.25e-10† 1.00e-08	2.36e-03† 5.69e-01	2.32e-12† 2.32e-01	2.13e-01† 2.12e-05	4.56e+01† 6.11e-01	1.13e-06† 0.00e+00	1.02e-03† 0.00e+00	2.65e-15† 2.36e-12	2.36e-10† 2.36e-06	3.36e-11† 3.65e-06	2.10e-10† 2.00e-06	3.30e-11† 3.10e-06	**0.00e+00** **0.00e+00**
F6	5.36e+02† 2.32e+00	5.36e+01† 1.11e+00	6.56e+01† 1.73e-04	5.45e+01† 3.11e-01	9.56e+02† 4.65e+01	4.53e+01† 1.00e+00	3.25e+01† 1.25e+00	2.36e-01† 1.25e-02	4.56e+01† 2.35e+00	3.22e+01† 2.65e-01	4.00e+01† 2.10e+00	3.10e+01† 2.01e-01	**9.56e-12** **1.54e+00**
F7	3.36e+03† 1.25e+02	3.25e+01† 6.36e+00	3.36e+02† 7.56e+01	9.56e+02† 4.34e+01	2.86e+01† 4.96e+00	1.23e+02† 2.36e+00	1.20e+02† 2.36e+00	2.36e-02† 2.36e-02	2.36e+01† 9.35e+00	1.23e+01† 3.36e+00	2.11e+01† 9.02e+00	1.19e+01† 3.11e+00	**2.17e-01** **1.42e+00**
F8	3.56e+01† 3.25e-02	2.33e+01† 7.14e-02	3.21e+01† 7.31e-01	2.85e+01† 4.93e-02	7.25e+01† 1.56e+01	2.15e+01† 1.88e-02	2.12e+01† 0.00e+00	2.36e+01† 1.25e+00	2.10e+01† 1.65e-02	2.00e+01† 1.69e-02	2.19e+01† 1.00e-02	1.95e+01† 1.59e-02	**1.90e+00** **2.25e+00**
F9	7.56e+02† 4.23e+00	6.36e+01† 4.35e+02	7.32e+01† 8.25e-01	7.56e+01† 1.25e+00	7.58e+01† 1.75e+00	7.56e+01† 5.82e-01	7.51e+01† 5.36e-01	2.31e+01† 1.23e+00	3.69e+01† 4.25e+00	2.36e+01† 4.65e+00	3.10e+01† 4.15e+00	2.16e+01† 4.05e+00	**1.30e+00** **2.05e+00**
F10	1.65e+02† 1.81e+01	4.69e-02† 2.23e-01	3.46e+02† 1.48e-02	4.56e-02† 2.14e-02	4.67e-02† 2.14e-02	6.40e-02† 2.59e-02	2.36e-02† 2.36e-02	2.65e-02† 1.25e-02	2.89e-01† 1.26e-01	3.65e-02† 1.64e-01	2.99e-01† 1.16e-01	3.25e-02† 1.14e-01	**6.23e-03** **1.23e-05**
F11	5.62e+01† 1.02e+00	2.35e-05† 3.23e-06	8.21e+01† 1.38e+01	5.12e+01† 4.15e+01	5.65e+01† 4.15e+01	2.32e+01† 2.59e-02	1.23e+01† 2.00e-01	1.65e-12† 1.23e-02	2.59e-12† 2.35e-11	3.65e-11† 3.65e-15	2.10e-12† 2.23e-11	3.05e-11† 3.15e-15	**0.00e+00** **0.00e+00**
F12	5.65e+01† 2.31e+00	1.23e+02† 1.70e+02	5.25e+02† 8.36e+00	4.12e+02† 9.99e+00	3.56e+02† 9.96e+00	3.75e+02† 5.36e+01	3.00e+02† 5.63e+01	1.25e+01† 2.36e+00	6.52e+01† 1.23e+01	5.65e+01† 3.26e+00	6.10e+01† 1.30e+01	5.15e+01† 3.66e+00	**9.60e+00** **2.36e-02**
F13	6.32e+02† 3.38e+01	2.32e+02† 2.79e+01	6.35e+02† 9.65e+00	4.56e+02† 3.25e+01	3.98e+02† 2.56e+01	3.78e+02† 5.73e+00	3.26e+02† 4.62e+00	6.35e+01† 1.64e+01	1.65e+02† 2.36e-01	1.22e+02† 3.36e-01	1.00e+02† 2.12e-01	1.21e+02† 3.32e-01	**5.36e+00** **2.56e+00**
F14	2.63e+03† 2.32e+02	**5.99e-03** **1.32e-02**	3.21e+03† 6.36e+02	1.20e+03† 4.59e+021	1.23e+03† 4.69e+02	9.56e+03† 1.83e+02	9.52e+03† 1.65e+02	8.65e-02† 2.65e-01	3.56e-02† 2.36e-01	2.36e-02† 3.35e-02	3.26e-02† 2.30e-01	2.30e-02† 3.30e-02	9.56e-03† 3.26e+03
F15	2.32e+04† 1.23e+02	9.99e+03† 6.32e+02	2.35e+05† 9.65e+02	1.58e+04† 2.04e+02	1.45e+04† 1.33e+02	1.62e+04† 2.04e+03	1.52e+04† 2.65e+01	**2.39e+02** **1.32e-01**	7.56e+03† 4.25e+02	8.56e+02† 3.48e+01	7.20e+03† 4.00e+02	8.10e+02† 3.41e+01	3.00e+02† 4.96e+01
F16	3.12e+00† 2.32e+00	4.25e+00† 3.65e-01	3.65e+00† 6.64e-02	3.26e+00† 4.25e-01	3.25e+00† 2.79e+00	3.36e+00† 4.22e-01	3.21e+00† 1.20e+00	5.65e+02† 1.25e+01	1.32e+00† 2.56e-01	1.12e+00† 2.65e-01	1.30e+00† 2.06e-01	**1.10e+00** **2.61e-01**	3.25e+00† 425e-02
F17	2.35e+02† 3.25e+00	5.99e+01† 7.89e-12	1.65e+02† 3.56e+01	3.25e+02† 1.65e+01	3.28e+02† 1.63e+01	2.89e+02† 1.23e+00	2.54e+02† 1.25e+00	5.64e+01† 1.25e+00	5.26e+01† 1.23e+01	4.25e+01† 2.36e+00	5.23e+01† 1.20e+01	4.20e+01† 2.30e+00	**2.79e+01** **2.16e-05**
F18	6.35e+02† 2.35e+01	2.92e+02† 3.36e+01	6.56e+02† 1.65e+01	4.56e+02† 1.05e+01	4.45e+02† 5.85e+00	4.27e+02† 1.35e+01	4.12e+02† 1.25e+01	2.36e+00† 1.24e+00	1.46e+02† 1.29e+01	1.02e+02† 1.56e+00	1.10e+02† 1.09e+01	1.00e+02† 1.51e+00	**1.30e+00** **1.25e+01**
F19	3.25e+01† 2.32e+00	3.12e+00† 2.32e-01	6.56e+00† 2.32e+00	3.25e+02† 9.69e-01	2.65e+01† 6.12e-02	2.82e+02† 1.36e+01	1.23e+02† 1.25e+01	2.36e+01† 1.25e+02	3.56e+00† 2.69e-01	1.25e+00† 5.69e-02	3.22e+00† 2.10e-01	1.21e+00† 5.62e-02	**4.01e-01** **1.56e-01**
F20	3.21e+01† 2.32e+00	3.24e+01† 4.60e+00	2.65e+01† 1.65e-01	3.21e+02† 5.32e-01	2.65e+01† 6.15e-02	2.36e+02† 2.69e+01	2.05e+02† 1.25e+01	3.21e+01† 1.25e+02	1.99e+01† 7.69e-01	1.00e+01† 2.36e-01	1.10e+01† 7.10e-01	1.10e+01† 2.06e-01	**3.20e+00** **1.56e+01**
F21	8.36e+02† 5.15e+02	6.74e+02† 4.35e-02	9.25e+02† 3.25e+02	7.12e+02† 4.25e+03	8.14e+02† 2.36e+02	7.12e+02† 1.36e+01	7.11e+02† 1.25e+01	1.56e+02† 1.24e+02	8.59e+02† 3.96e+02	7.56e+02† 3.25e-02	8.19e+02† 3.06e+02	7.26e+02† 3.29e-02	**1.36e+02** **1.23e+00**
F22	3.56e+03† 5.36e+02	2.99e+01† 1.23e+01	3.26e+03† 1.65e+02	1.25e+04† 1.69e+03	1.25e+04† 2.65e+01	1.06e+04† 1.86e+03	1.65e+04† 1.25e+03	1.56e+01† 2.36e+01	1.36e+01† 1.36e+00	1.11e+01† 3.36e-02	1.32e+01† 1.26e+00	1.09e+01† 3.06e-02	**2.65e+00** **1.25e-01**
F23	2.36e+04† 1.35e+02	1.04e+04† 2.32e+03	1.65e+04† 3.26e+02	2.15e+05† 5.21e+01	1.50e+04† 2.56e+01	1.53e+04† 5.36e+02	3.25e+04† 2.36e+02	2.65e+01† 2.35e+00	7.69e+03† 6.69e+02	7.56e+03† 3.25e+02	7.10e+03† 6.36e+02	7.06e+03† 3.05e+02	**1.51e+00** **2.45e+01**
F24	4.25e+02† 2.36e+00	3.10e+02† 1.56e+01	4.56e+02† 6.32e+00	3.89e+02† 2.32e+00	3.99e+02† 1.96e+01	3.96e+02† 2.80e+02	3.36e+04† 1.25e+02	1.56e+01† 2.32e+00	2.36e+02† 1.01e+01	2.22e+02† 1.01e-01	2.11e+02† 1.00e+01	**2.02e+02** **1.00e-01**	3.56e+02† 2.23e+02
F25	3.85e+02† 2.36e+01	3.95e+02† 2.32e+01	5.65e+02† 6.26e+01	3.82e+02† 2.78e+00	3.54e+02† 2.96e+00	3.56e+03† 1.62e+02	3.64e+03† 1.25e+02	1.56e+03† 1.65e+01	3.96e+02† 3.06e+01	3.45e+02† 3.25e-01	3.12e+02† 3.03e+01	3.15e+02† 3.21e-01	**1.96e+02** **1.23e+00**
F26	5.12e+02† 2.36e+00	3.56e+02† 9.56e+01	4.76e+02† 3.23e+01	5.21e+02† 5.79e+01	4.56e+02† 2.45e+01	4.85e+02† 1.35e+01	9.35e+02† 1.32e+05	2.56e+02† 1.58e+02	2.69e+02† 9.65e+01	1.56e+02† 3.25e+01	2.67e+02† 9.00e+01	**1.23e+02** **3.15e+01**	1.76e+02† 1.26e+01
F27	2.56e+03† 4.95e+01	2.32e+03† 3.36e+01	3.26e+03† 4.25e+01	2.56e+03† 4.64e+01	2.15e+03† 2.0e+01	2.13e+03† 1.25e+02	2.36e+03† 5.36e+01	6.65e+03† 1.25e+00	9.56e+02† 3.01e+02	9.25e+02† 1.26e+02	9.25e+02† 3.03e+02	9.15e+02† 1.06e+02	**2.12e+02** **2.25e+00**
F28	2.32e+03† 1.63e+03	5.02e+02† 4.25e-02	4.15e+02† 1.32e+00	1.43e+03† 1.78e+02	1.56e+03† 1.74e+03	1.42e+03† 1.22e+02	1.25e+03† 2.32e+01	6.66e+02† 1.25e+01	4.56e+02† 4.15e+02	4.25e+02† 1.53e+02	4.50e+02† 4.22e+02	4.16e+02† 1.13e+02	**3.96e+02** **0.00e+00**
W/T/L	0/0/28	1/0/27	0/0/28	0/0/28	0/0/28	0/0/28	0/0/28	1/0/27	0/0/28	0/0/28	0/0/28	4/0/24	**22/0/6**

remains an added advantage for MDE-DS. Also it is to be noted that the gross performance of MDE-DS is better than NRDE which however remains closes contender according to average rank.

Table 2B summarizes simulation results for the four other noise models on the 100D Bench-1 problems. MDE-DS performs best in 49 test cases, a close contender of it being MUDE, which outperforms all methods in exponential noise model for f_9 and DERSFTS, which gives the best result for random noise model in f_{10} and f_{11}. MDE-DS performs best followed by MUDE and DERSFTS, based on the average ranking, indicating its high degree of robustness across various noise PDFs.

F. SIMULATION RESULTS ON BENCH-2 PROBLEMS

Bench-2 comprises of total 8 well-known numerical functions corrupted with zero mean Gaussian noise of 5 different noise amplitudes (0.2, 0.4, 0.6, 0.8, and 1.0). Simulation results of MDE-DS on Bench-2 are compared with PSOlbest+CJ, PSOgbest+CJ, FIPS+CJ, and BBPSO+CJ [37].

30 runs for each problem corresponding to each noise level and for all algorithms are considered and mean and standard deviation of the absolute error is reported. Maximum FEs for Bench-2 problems are limited to $3e{+}04$.

Table 3 holds simulation results of Bench-2 problems. We can see that MDE-DS outperforms other algorithms in thirty-eight cases, only for two cases FIPS+CJ beats MDE-DS. According to the average ranking, MDE-DS stands first followed by PSOgbest+CJ and BBPSO+CJ.

G. SIMULATION RESULTS ON IEEE CEC 2005 FUNCTION f_{17}

IEEE CEC 2005 is a well-accepted benchmark suite for testing effectiveness of any single objective global optimization algorithm. Among other functions in this suite f_{17} is a hybrid function which is corrupted by noise. We compare

TABLE 6. Cost ± standard deviation for 50D CEC 2017 problems for Gaussian noise with 0.2 noise strength.

Func	Algorithms												
	DE/rand/1/bin	jDE	GADS	DERSFTS	OBDE	NADE	MUDE	NRDE	jSO	L-SHADE-cnEpSin	jSO-DS	L-SHADE-cnEpSin-DS	MDE-DS
F1	5.06E+09† 6.32E+08	2.98E+03† 2.92E+03	2.69E+03† 3.26E+03	2.13E+03† 2.99E+03	2.59E+03† 3.23E+03	2.09E+09† 3.35E+08	2.65E+03† 3.74E+03	2.02E+02† 1.20E+02	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**
F2	3.96E+63† 5.68E+63	1.16E+58† 3.63E+58	4.49E+55† 1.13E+56	1.18E+58† 3.63E+58	4.56E+55† 1.32E+56	3.89E+63† 6.63E+63	4.95E+55† 1.26E+56	3.38E+02† 1.23E+56	**0.00e+00≈ 0.00e+00**	1.59e+00† 1.95e+00	**0.00e+00≈ 0.00e+00**	1.15e+00† 1.44e+00	**0.00e+00≈ 0.00e+00**
F3	1.91E+05† 1.55E+04	1.22E+05† 1.34E+04	1.23E+05† 1.22E+04	1.29E+05† 1.33E+04	1.58E+05† 1.36E+04	1.59E+05† 1.65E+04	1.36E+05† 1.95E+04	1.12E+02† 1.23E+04	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**
F4	9.94E+02† 9.27E+02	1.39E+02† 4.28E+01	1.69E+02† 4.45E+01	1.36E+02† 4.22E+01	1.36E+02† 4.23E+01	9.75E+02† 9.65E+02	1.74E+02† 4.36E+01	1.22E+01† 4.32E+01	5.63e+01† 4.82e+01	5.12e+01† 4.46e+01	3.25e+01† 1.03e-01	3.12e+01† 2.10e+01	**3.32e+00 2.56e-05**
F5	5.67E+02† 1.71E+01	4.04E+02† 1.42E+01	3.74E+02† 1.92E+01	4.07E+02† 1.46E+01	3.98E+02† 1.96E+01	4.95E+02† 1.15E+01	3.96E+02† 1.45E+01	3.85E+01† 1.92E+01	1.66e+01† 3.49e+00	2.59e+01† 6.48e+00	1.23e+02† 1.02e+00	1.65e+01† 1.31e+00	**9.38e+00 2.35e-01**
F6	1.88E+01† 1.36E+00	3.79E-04† 1.74E-04	1.14E-04† 3.63E-05	3.79E-04† 1.72E-04	1.33E-04† 3.28E-05	1.36E+01† 1.45E+00	1.69E-04† 3.58E-05	1.32E-04† 3.23E-05	1.99e-06† 2.99e-06	9.14e-07† 1.06e-06	1.01e-07† 2.12e-01	4.68e-08† 1.12e-04	**0.00e+00 0.00e+00**
F7	6.02E+02† 1.83E+01	4.51E+02† 1.54E+01	4.18E+02† 1.69E+02	4.59E+02† 1.54E+01	4.56E+02† 1.46E+02	6.25E+02† 1.36E+01	4.65E+02† 1.69E+02	4.51E+02† 1.24E+02	6.56e+01† 3.55e+00	7.69e+01† 6.08e+00	2.12e+01† 1.20e+00	**1.21e+01 1.02e+01**	1.85e+01† 2.62e-05
F8	5.65E+02† 1.49E+01	4.09E+02† 1.54E+01	4.56E+02† 1.33E+01	4.02E+02† 1.56E+01	3.98E+02† 1.33E+01	4.65E+02† 1.45E+01	3.86E+02† 1.96E+01	3.15E+02† 1.21E+01	1.99e+01† 3.89e+00	2.69e+01† 6.58e+00	1.62e+01† 3.22e+00	**1.03e+01 2.25e+00**	5.43e+01† 1.34e-02
F9	7.02E+03† 9.09E+02	2.19E-02† 9.01E-02	1.79E-04† 1.24E-03	2.19E-02† 9.02E-02	1.79E-04† 1.22E-03	7.02E+03† 9.06E+02	1.45E-04† 1.36E-03	1.65E-02† 1.21E-03	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**	**0.00e+00≈ 0.00e+00**
F10	1.39E+04† 3.03E+02	1.36E+04† 2.99E+02	1.36E+04† 3.68E+02	1.59E+04† 2.98E+02	1.22E+04† 3.66E+02	1.45E+04† 3.75E+02	1.63E+04† 3.66E+02	**1.26E+03 3.62E+01**	3.95e+03† 3.66e+02	3.29e+03† 3.38e+02	2.31e+03† 1.22e+01	3.23e+03† 3.12e+02	5.87e+03† 2.25e-02
F11	8.99E+02† 8.95E+01	2.29E+02† 1.99E+01	2.20E+02† 1.86E+01	2.26E+02† 1.99E+01	2.08E+02† 1.86E+01	8.14E+02† 8.65E+01	2.96E+02† 1.45E+01	2.00E+01† 1.86E+00	2.72e+01† 3.21e+00	2.19e+01† 2.29e+00	2.06e+01† 1.81e+00	2.22e+01† 2.01e+00	**7.90e+00 3.36e-04**
F12	5.98E+08† 8.63E+07	5.49E+07† 1.49E+06	4.96E+07† 1.16E+06	5.48E+07† 1.43E+06	4.59E+07† 1.12E+06	5.86E+08† 8.96E+07	4.56E+07† 1.36E+06	4.51E+03† 1.21E+01	1.69e+03 5.22e+02	1.49e+03† 3.69e+02	1.89e+03† 1.22e+02	1.35e+03† 2.13e+02	**9.81e+02 2.32e-01**
F13	9.98E+05† 4.45E+05	4.19E+04† 2.99E+04	5.67E+04† 2.59E+04	4.19E+04† 2.96E+04	5.66E+04† 2.09E+04	9.64E+05† 4.42E+05	5.69E+04† 2.45E+04	5.24E+02† 2.06E+01	3.03e+01† 2.19e+01	6.99e+01† 3.49e+01	2.61e+01† 1.62e+01	5.02e+01† 3.74e+01	**2.23e+01 2.34e-04**
F14	4.98E+05† 1.45E+05	9.79E+04† 4.29E+04	8.99E+04† 3.48E+04	9.79E+04† 4.26E+04	8.89E+04† 3.49E+04	4.32E+05† 1.56E+05	8.36E+04† 3.45E+04	8.25E+01† 3.40E+01	2.59e+01† 1.88e+00	2.69e+01† 2.49e+00	2.10e+01† 1.32e+00	2.02e+01† 1.49e+00	**5.26e+00 3.38e-04**
F15	7.48E+04† 2.68E+04	1.19E+04† 5.29E+03	1.36E+04† 4.85E+03	1.19E+04† 5.25E+03	1.29E+04† 4.59E+03	7.96E+04† 2.64E+04	1.96E+04† 4.68E+03	1.21E+02† 4.55E+03	2.40e+01† 2.10e+00	2.59e+01† 4.08e+00	2.45e+01≈ 1.61e+00	2.48e+01† 2.99e+00	2.94e+01† 5.64e-02
F16	3.95E+03† 1.19E+02	3.18E+03† 2.49E+02	3.95E+03† 2.36E+02	3.19E+03† 2.06E+02	3.09E+03† 2.09E+02	3.69E+03† 1.12E+02	3.12E+03† 2.36E+02	3.02E+01† 2.02E+01	4.65e+02† 1.65e+02	2.79e+02† 9.99e+01	4.10e+02† 1.96e+02	4.88e+02† 1.69e+02	**1.96e+00 2.38e-01**
F17	1.83E+03† 1.69E+02	1.24E+03† 1.45E+02	1.80E+03† 1.96E+02	1.78E+03† 1.56E+02	1.45E+03† 1.65E+02	1.36E+03† 1.65E+02	1.96E+03† 1.42E+02	1.72E+01† 1.20E+01	2.63e+01† 2.52e+00	2.09e+02† 7.39e+01	2.42e+01† 1.62e+00	2.08e+02† 9.16e+01	**7.76e+00 2.62e-01**
F18	9.99E+06† 2.67E+06	4.98E+06† 1.55E+06	3.46E+06† 1.95E+05	4.75E+06† 1.59E+06	3.36E+06† 1.65E+05	9.36E+06† 2.20E+06	3.86E+06† 1.12E+05	3.20E+01† 1.10E+01	2.33e+01† 2.32e+00	2.49e+01† 2.18e+00	1.63e+01† 1.61e+00	2.88e+01† 4.83e+00	**6.05e+00 3.64e-04**
F19	9.59E+04† 2.37E+04	1.86E+04† 6.99E+03	9.84E+03† 4.36E+03	1.02E+04† 6.46E+03	9.66E+03† 4.12E+03	9.64E+04† 2.45E+04	9.93E+03† 4.14E+03	9.66E+01† 4.13E+01	1.91e+01† 2.92e+00	1.78e+01† 2.45e+00	1.05e+01† 6.56e+00	1.88e+01† 4.90e+00	**5.53e+00 6.24e-03**
F20	1.79E+03† 1.45E+02	1.65E+03† 1.15E+02	1.32E+03† 1.78E+02	1.53E+03† 1.38E+02	1.49E+03† 1.73E+02	1.71E+03† 1.93E+02	1.36E+03† 1.86E+02	1.30E+01† 1.25E+00	1.95e+02† 7.96e+01	1.18e+02† 3.59e+01	1.29e+02† 1.29e+01	1.89e+02† 1.59e+01	**1.86e+01 2.62e-05**
F21	6.84E+02† 1.56E+01	6.69E+02† 1.45E+01	5.54E+02† 1.59E+01	6.06E+02† 1.45E+01	5.96E+02† 1.59E+01	6.73E+02† 1.91E+01	5.63E+02† 1.25E+01	5.92E+02† 1.68E+01	2.52e+02† 3.25e+00	2.29e+02† 7.08e+00	1.68e+02† 2.38e+00	2.69e+02† 8.89e+00	**1.00e+02 2.64e-04**
F22	1.46E+04† 1.36E+03	1.36E+04† 4.95E+03	1.80E+04† 4.93E+03	1.19E+04† 4.06E+03	1.09E+04† 4.92E+03	1.83E+04† 1.29E+03	1.56E+04† 4.96E+03	1.22E+04† 4.26E+03	1.63e+03† 1.65e+03	1.88e+03† 1.68e+03	1.45e+03† 1.56e+02	1.25e+03† 1.72e+03	**1.00e+02 2.74e-04**
F23	9.98E+02† 2.13E+01	8.46E+02† 1.54E+01	8.12E+02† 1.66E+01	8.25E+02† 1.69E+01	8.15E+02† 1.63E+01	9.67E+02† 2.76E+01	8.36E+02† 1.66E+01	8.67E+02† 1.38E+01	4.45e+02† 6.69e+00	4.39e+02† 6.98e+00	4.12e+02† 6.13e+00	4.31e+02† 1.02e+00	**3.02e+02 1.84e-05**
F24	9.76E+02† 1.42E+01	8.78E+02† 1.35E+01	8.89E+02† 1.42E+01	8.94E+02† 1.32E+01	8.85E+02† 1.43E+01	9.15E+02† 1.35E+01	8.96E+02† 1.75E+01	8.62E+02† 1.81E+01	5.02e+02 4.16e+00	5.19e+02† 5.58e+00	5.12e+02† 1.36e+01	5.10e+02† 4.12e+00	**1.00e+02 0.00e+00**
F25	1.13E+03† 7.34E+01	5.56E+02† 3.34E+01	4.96E+02† 2.58E+01	5.08E+02† 3.51E+01	4.99E+02† 2.52E+01	1.42E+03† 7.26E+01	4.95E+02† 2.56E+01	4.63E+02† 2.99E+01	4.82e+02† 2.76e+00	4.82e+02† 1.06e+00	4.51e+02† 1.65e+00	4.78e+02† 2.01e+00	**3.99+02 1.64e-02**
F26	6.09E+03† 1.76E+02	4.94E+03† 1.86E+02	4.89E+03† 1.56E+02	4.99E+03† 1.82E+02	4.86E+03† 1.52E+02	6.36E+03† 1.25E+02	4.82E+03† 1.59E+02	4.95E+03† 1.75E+02	1.15e+03† 5.66e+01	1.28e+03† 1.19e+02	1.08e+03† 1.69e+01	1.01e+03† 9.32e+01	**3.00e+02 0.00e+00**
F27	1.29E+03† 5.03E+02	6.79E+02† 4.92E+01	6.59E+02† 3.91E+01	6.73E+02† 4.92E+01	6.56E+02† 3.93E+01	1.14E+03† 5.02E+02	6.54E+02† 3.96E+01	6.75E+02† 3.80E+01	5.18e+02 1.16e+01	5.27e+02† 9.28e+00	5.09e+02† 1.28e+01	5.13e+02† 2.12e+01	**3.98e+02 1.66e-02**
F28	1.08E+03† 7.23E+01	4.63E+02† 1.38E+01	4.56E+02† 6.89E+00	4.66E+02† 1.33E+01	4.60E+02† 6.25E+00	1.65E+03† 7.56E+01	4.59E+02† 6.82E+00	4.99E+02† 6.84E+00	4.66e+02 6.81e+00	4.59e+02† 1.18e+01	4.66e+02† 1.32e+00	4.51e+02† 2.62e+01	**3.01e+02 1.59e-02**
F29	2.79E+03† 1.51E+02	2.06E+03† 2.74E+02	2.65E+03† 2.65E+02	2.89E+03† 2.71E+02	2.03E+03† 2.30E+02	2.48E+03† 1.95E+02	1.96E+03† 2.09E+02	1.59E+03† 2.65E+02	3.33e+02† 1.32e+01	3.51e+02† 9.72e+00	3.35e+02† 1.86e+01	9.10e+02† 1.72e+02	**2.98e+02 1.34e-05**
F30	6.68E+07† 1.61E+07	1.53E+07† 4.87E+06	1.95E+07† 3.32E+06	1.25E+07† 4.65E+06	1.69E+07† 3.32E+06	6.63E+07† 1.36E+07	1.62E+07† 3.33E+06	1.50E+07† 3.41E+06	6.09e+05† 2.96e+04	6.56e+05† 7.25e+04	2.98e+05† 1.55e+04	2.13e+03† 2.20e+02	**1.26e+03 2.64e-02**
W/T/L	0/0/30	0/0/30	0/0/30	0/0/30	0/0/30	0/0/30	0/0/30	1/0/29	0/4/26	0/3/27	0/4/26	2/3/25	22/4/4

performance of MDE-DS on f_{17} against IPOP-CMA-ES, which was the winner of the competition on real parameter optimization under IEEE CEC 2005. Table 4 summarizes the results of comparison on 30D and 50D instances of f_{17}.

We can see that for both 30D and 50D cases, MDE-DS outperforms IPOP-CMA-ES in a statistically significant way. Maximum number of FEs are limited to $3e{+}05$ for 30D instance and $5e{+}05$ for 50D following the competition rules of CEC 2005.

H. SIMULATION RESULTS ON IEEE CEC 2013 FUNCTIONS

TABLE 5 shows the comparative performance of MDE-DS with other previously discussed algorithms along with two improved DE variants, namely, SHADE [41] and L-SHADE [42] as well as their counterparts SHADE-DS and LSHADE-DS, equipped with the proposed distance based selection applied on the noisy versions of the 28 test functions in 50D from the CEC 2013 test-suite [38]. Noise simulation (by adding Gaussian noise of strength 0.2) for these benchmark functions are done in same fashion like previous Bench-1 and Bench-2 problems. A close scrutiny of Table 5 reveals that MDE-DS significantly outperforms all the peer algorithms on 22 out of 28 test cases. The performance of SHADE-DS and L-SHADE-DS remained consistently superior to that of SHADE and L-SHADE respectively on majority of the test cases, thus, showcasing the efficiency of the distance-based threshold scheme proposed as a component of the MDE-DS algorithm. For functions F4, F16, F24,

TABLE 7. Cost ± standard deviation for 50D CEC 2017 composite problems for Gaussian noise.

	Algorithms												
Func	DE/rand/1/bin	jDE	GADS	DERSFTS	OBDE	NADE	MUDE	NRDE	jSO	L-SHADE -cnEpSin	jSO-DS	L-SHADE-cnEpSin -DS	MDE-DS
F25	4.56E+03† 9.32E+01	6.23E+02† 3.35E+01	5.63E+02† 3.25E+01	5.65E+02† 2.55E+01	5.81E+02† 1.96E+01	2.41E+03† 6.72E+01	5.89E+02† 2.12E+01	4.63E+02† 3.99E+01	5.12e+02† 3.16e+00	5.96e+02† 2.05e+00	4.96e+02† 1.64e+00	5.00e+02† 2.05e+00	**4.00+02 1.22e-02**
F26	7.56E+03† 1.96E+02	5.85E+03† 2.95E+02	5.99E+03† 2.99E+02	5.18E+03† 1.11E+02	5.76E+03† 1.41E+02	7.13E+03† 1.35E+02	5.86E+03† 1.22E+02	5.01E+03† 1.65E+02	2.25e+03† 6.61e+01	2.18e+03† 1.29e+02	1.66e+03† 1.62e+01	1.02e+03† 1.32e+01	**3.15e+02 1.25e+00**
F27	6.62E+03† 3.53E+02	7.96E+02† 5.85E+01	7.59E+02† 2.21E+01	7.96E+02† 6.92E+01	7.22E+02† 2.91E+01	2.53E+03† 2.63E+02	7.55E+02† 3.25E+01	7.20E+02† 2.60E+01	6.20e+02 1.82e+01	6.12e+02† 1.05e+01	5.12e+02† 1.25e+01	5.11e+02† 2.16e+01	**3.99e+02 1.41e-02**
F28	2.56E+03† 2.19E+01	5.21E+02† 2.22E+01	5.12E+02† 5.89E+00	5.70E+02† 2.63E+01	5.79E+02† 5.21E+00	2.96E+03† 8.56E+01	5.91E+02† 5.32E+00	5.12E+02† 3.64E+00	5.26e+02 5.81e+00	5.15e+02† 3.28e+01	4.96e+02† 1.41e+00	4.52e+02† 2.63e+01	**3.52e+02 1.55e-02**
F29	3.25E+03† 1.23E+02	2.69E+03† 2.65E+02	3.96E+03† 2.36E+02	3.86E+03† 3.72E+02	2.93E+03† 2.12E+02	3.02E+03† 2.95E+02	2.53E+03† 1.56E+02	2.11E+03† 2.62E+02	4.56e+02† 2.42e+01	4.96e+02† 2.96e+00	3.96e+02† 1.14e+01	9.65e+02† 1.74e+02	**3.10e+02 1.41e-05**
F30	7.96E+08† 3.61E+07	2.98E+07† 1.99E+06	2.06E+07† 2.56E+06	2.96E+07† 2.65E+06	2.88E+07† 1.32E+06	7.86E+07† 1.02E+07	2.96E+07† 1.24E+06	1.59E+07† 2.45E+06	6.99e+05† 3.96e+04	5.12e+05† 2.35e+04	3.02e+05† 1.56e+04	2.46e+03† 2.28e+02	**1.56e+03 6.74e-02**
W/T/L	0/0/6	0/0/6	0/0/6	0/0/6	0/0/6	0/0/6	0/0/6	0/0/6	0/0/6	0/0/6	0/0/6	0/0/6	6/0/0

TABLE 8. Algorithm Complexity results.

		Algorithms								
			DE/rand1/bin		jSO		LSHADE-cnEpSin		MDE-LS	
D	T0	T1	T2	(T2-T1)/T0	T2	(T2-T1)/T0	T2	(T2-T1)/T0	T2	(T2-T1)/T0
10	0.0968	0.7971	2.0152	12.5867	3.5621	15.9803	3.1295	24.0950	2.4516	17.0919
30		1.7976	3.6589	19.2283	6.9856	53.5950	6.7094	50.7417	5.6985	40.2985
50		3.4125	5.6987	23.6177	9.1239	59.0020	8.4912	52.4659	7.9874	47.2613

and F26, L-SHDAE-DS is able to beat MDE-DS. NRDE attains the best result only for function F15. Performance of the other seven algorithms, including six DE-variants specifically meant for noisy optimization, remains quite poor as compared to MDE-DS, SHADE-DS, and L-SHADE-DS.

I. SIMULATION RESULTS ON IEEE CEC 2017 FUNCTIONS

TABLE 6 records the comparative performance of MDE-DS with respect to other previously discussed methods as well as two more algorithms, jSO [44] and LSHADE-cnEpSin [45] and their counterparts jSO-DS and LSHADE-cnEpSin-DS equipped with the distance-based threshold for handling additive noise on the 50D IEEE CEC 2017 benchmark suite [39] with simulated Gaussian noise of noise strength 0.2. Table 6 indicates that MDE-DS is able to outperform all the peer algorithms including the competition winner jSO and the runner up LSHADE-cnEpSin on 22 out of 30 functions in a statistically significant manner. On relatively simpler unimodal functions F1 - F3 and multimodal F9, the performance of MDE-DS remains statistically equivalent to jSO, LSHADE-cnEpSin and their DS counterparts (on F1, F3 and F9) and superior to all others. NRDE attains the best rank only on function F10 (shifted and rotated Schwefel's function). It can be also seen that addition of the proposed distance based selection scheme to jSO and LSHADE-cnEpSin algorithms considerably improves their performance on most of the noise added functions.

Table 7 shows a special scenario when the Gaussian noise strength is made proportional to the original objective function value. In this simulation we took the proportionality constant as 0.5 since if this constant exceeds 0.5, the objective function gets too much distorted and no algorithms considered here, is able to estimate the optimal basin due to severe random fluctuations of the fitness landscape. Gaussian noise with the calculated strength is added to the original objective function values to simulate the noisy behavior. In this scenario of non-uniform Gaussian noise with cost-proportionate strength, MDE-DS significantly outperforms all other competitor algorithms. For the sake of space economy, we show results for the five composite functions (F25 - F30) as they contain mixture of properties of several basic functions and are quite hard to optimize. Also, the results on all other functions follow the same trend, as our experiments indicate.

The algorithm complexity of MDE-DS is estimated following the instructions prepared for the CEC 2017 competition on real parameter optimization [39] and the same is recorded in Table 8. The computational complexity is calculated as described in [39]. In Table 8, *T0* is the time needed to run following test problem:

for $i = 1 : 1000000$

$x = x + x; x = x/2; x = x * x;$

$x = sqrt(x); x = log(x);$

$x = exp(x); x = x\ /(x + 2);$

end

The computed complexities for 10, 30 and 50 dimensions of the are shown. T1 is the time required to execute 200,000 evaluations of function F18 (as instructed in [39]) from the CEC 2017 benchmark suite and T2 is the time to execute corresponding algorithm with 200,000 evaluations of F18 in dimension *D*. (T2-T1)/T0 shows the time complexity for each corresponding algorithm, lower value for this indicator reflects less complexity. As can be seen, for all the 3 dimensions, the estimated complexity of MDE-DS is lower

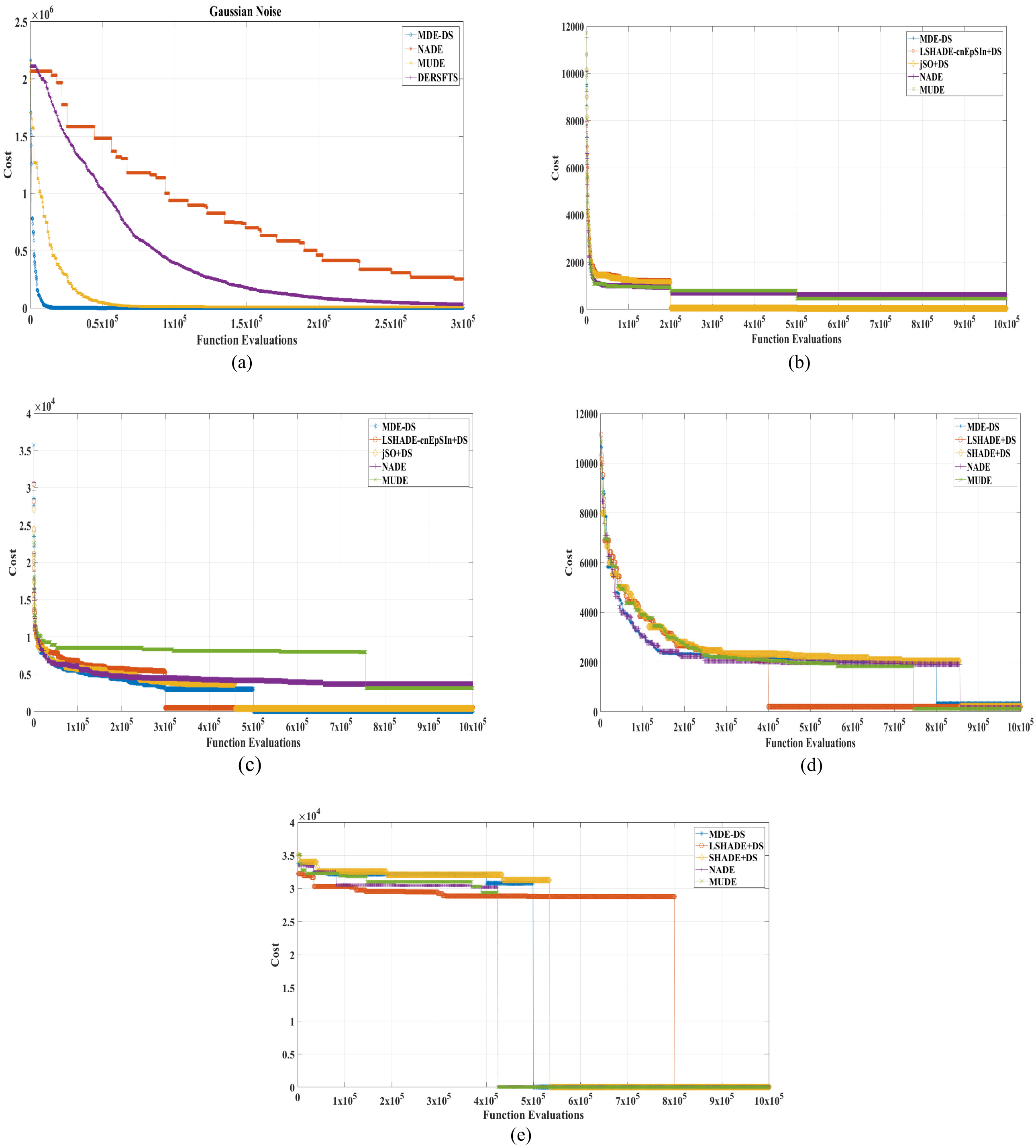


FIGURE 5. **Sample convergence characteristics of MDE-DS and other competitors for Gaussian noise strength of 0.2 coupled with (a) function f_3 from Bench - I (b) CEC'17 function F16 (c) CEC'17 function F7 (d) CEC'13 function F10 (e) CEC'13 function F24.**

compared to jSO and LSHADE-cnEpSin, due to the use of simpler search operators.

However, the complexity of MDE-DS is higher than DE/rand/1/bin, as a price of achieving significantly better accuracy across a wide spectrum of objective functions.

Fig. 5 shows sample and representative convergence characteristics of MDE-DS and some of its prominent competitors on five functions chosen from various test-suites used. The characteristics are plotted for the median run of each algorithm when the runs are ordered by the best-of-the-run error values achieved. The plots clearly indicate the

TABLE 9. Cost ± standard deviation for selected 100D bench-1 problems with Gaussian noise.

Function	MDE-DS with $b = 0.1$	MDE-DS with $b = 0.5$	MDE-DS with $b = 0.9$	MDE-DS with $b = \text{rand}[0,1]$	MDE-DS
f_1	1.46e-03† 2.69e-04	2.66e-05† 3.86e-08	1.96e-05† 2.26e-06	2.24e-03† 1.24e-08	**0.00e+00 0.00e+00**
f_5	5.56e-01† 2.77e-02	3.55e-02† 1.29e-03	1.55e-02† 2.68e-06	4.45e-02† 2.87e-08	**4.82e-04 2.45e+00**
f_{10}	9.85e-01† 2.54e-06	6.45e-01† 5.05e-01	2.49e-01† 1.59e-01	2.56e-01† 2.96e-04	**1.54e-02 4.75e-03**
f_{12}	2.82e-03† 1.69e-05	2.55e-05† 8.46e-06	1.26e-05† 2.46e-06	1.24e-03† 4.22e-05	**0.00e+00 0.00e+00**

TABLE 10. Cost ± standard deviation for selected 100D bench-1 problems with Gaussian noise.

Function	MDE-DS with *PC_MS*	MDE-DS with *DMB_MS*	MDE-DS
f_1	5.75e-02† 7.68e-01	7.68e-01† 4.56e--01	**0.00e+00 0.00e+00**
f_5	2.75e-02† 4.85e-03	7.84e-01† 2.88e-01	**4.82e-04 2.45e+00**
f_{10}	1.22e-01† 2.55e-01	5.45e+00† 2.05e+00	**1.54e-02 4.75e-03**
f_{12}	8.84e-02† 2.29e-02	3.45e+00† 2.46e+00	**0.00e+00 0.00e+00**

competitive convergence speed of MDE-DS on different functional landscapes with respect to other DE-variants.

J. EFFECT OF DIFFERENT PARAMETER CONFIGURATIONS

MDE-DS has only one parameter b which is used in the recombination step. Value for b is selected from among 3 distinct representative values and these are 0.1, 0.5 and 0.9. In our proposal, if b is too small then the target vector gets preference in offspring generation, and on the other hand, if b is too high or close to 1, then the donor vector gets preference. Thus, we proposed a scheme where value of b is randomly selected from the set $\{0.1, 0.5, 0.9\}$ corresponding to high target preference, neutral and high donor preference scenarios in offspring generation. In order to demonstrate the effectiveness of proposed scheme over a fixed b value, we compare performances of the MDE-DS variants with 4 different b value settings and contrast them against the performance of the original MDE-DS. Table 9 summarizes results on four different functions from Bench-1 with Gaussian noise having 0.2 as noise amplitude.

K. EFFECT OF DIFFERENT MUTATION CONFIGURATIONS

MDE-DS uses two mutation strategies which we call Population Centrality based Mutation strategy (PC_MS) and Difference Mean based Mutation Strategy (DM_MS). These two different strategies are coupled in a probabilistic switchable manner. PC_MS tends to a greedy search and DM_MS makes room for a diversified search along the fitness landscape. To demonstrate the effect of invoking these two schemes with equal probabilities (as is done in MDE_DS) against using any one scheme for all individuals, we compare original MDE_DS against two of its algorithmic variants: one with PC_MS and the other with DM_DS only. Sample results are provided for four 100D Bench-1 problems in Table 10, which indicates the best performance of MDE-DS and this trend is seen experimentally for all other test functions as well. This indicates that when we do not have specific feedback information about the fitness landscape, it is always beneficial to uniformly mix up opposite natured schemes so as to gain an overall better performance on a wide variety of problems.

TABLE 11.A. Cost ± standard deviation for selected 100D bench-1 problems with Gaussian noise.

Function	MDE-DS + Binomial crossover	MDE-DS with *Blending crossover*
f_1	3.58e-03† 3.25e-01	**0.00e+00 0.00e+00**
f_5	4.98e-01† 2.45e-03	**4.82e-04 2.45e+00**
f_{10}	1.96e-01† 1.74e-01	**1.54e-02 4.75e-03**
f_{12}	9.65e-02† 2.52e-02	**0.00e+00 0.00e+00**

TABLE 11.B. Cost ± standard deviation for selected 50D CEC 2017 problems with Gaussian noise with strength 0.2.

Function	MDE-DS + Binomial crossover	MDE-DS with *Blending crossover*
f_{25}	4.58e+03† 6.21e+01	**3.99+02 1.64e-02**
f_{26}	4.41e+02† 6.29e+01	**3.00e+02 0.00e+00**
f_{27}	5.74e+02† 1.88e+02	**3.98e+02 1.66e-02**
f_{28}	5.84e+02† 1.52e+01	**3.01e+02 1.59e-02**
f_{29}	6.12e-02† 3.82e-02	**2.98e+02 1.34e-05**
f_{30}	6.12e+03† 1.58e-01	**1.26e+03 2.64e-02**

L. EFFECT OF DIFFERENT CROSSOVER OPERATIONS

In Table 11, two comparisons are presented between MDE-DS with binomial crossover and with blending crossover. In Table 11A four representative functions are shown from Bench-1 and in Table 11B six composite functions are selected from the IEEE CEC 2017 test-suite for this comparison purpose. In both of the cases, Gaussian noise with noise strength of 0.2 is considered. From Tables 11A and B it can be seen that MDE-DS with blending crossover every time outperforms its counterpart equipped with traditional binomial crossover. This empirically shows that blending crossover gets an edge over traditional binomial crossover for all the problems tested here.

V. CONCLUSION

To tackle single-objective, noisy and continuous optimization problems, we present a simple but very efficient DE variant namely MDE-DS, which is equipped with simple switchable mutation strategies based on population central tendency and the difference mean based perturbation, a blending crossover, and has a unique distance-based stochastic selection

TABLE 12. Detailed description of the benchmark suites.

Test Problem	Function	Dimension (D)	Decision Space
Bench-1			
f_1: Ackley Function	$-20+e+20\exp\left(-\frac{0.2}{D}\sqrt{\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi.x_i)x_i\right)$	30,100	$[-1,1]^D$
f_2: Alpine Function	$\sum_{i=1}^{D}\lvert x_i\sin x_i+0.1x_i\rvert$	30,100	$[-10,10]^D$
f_3: Axis Parallel Function	$\sum_{i=1}^{D}ix^2$	30,100	$[-5.12,5.12]^D$
f_4: DeJong Function	$\lvert\lvert x\rvert\rvert^2$	30,100	$[-5.12,5.12]^D$
f_5: DropWave Function	$-\frac{1+\cos(12\sqrt{\lvert\lvert x\rvert\rvert^2})}{\frac{1}{2}\lvert\lvert x\rvert\rvert^2+2}$	30,100	$[-5.12,5.12]^D$
f_6: Griewank Function	$\frac{\lvert\lvert x\rvert\rvert^2}{4000}-\prod_{i=0}^{D}\cos\frac{x_i}{\sqrt{i}}+1$	30,100	$[-600,600]^D$
f_7: Michalewicz Function	$-\sum_{i=1}^{D}\sin x_i(\sin(\frac{i.x_i^2}{\pi}))^{20}$	30,100	$[0,\pi]^D$
f_8: Moved Axis Function	$\sum_{i=1}^{D}5ix_i^2$	30,100	$[-5.12,5.12]^D$
f_9: Pathological Function	$\sum_{i=1}^{D-1}(0.5+\frac{\sin^2(\sqrt{100x_i^2+x_{i+1}^2}-0.5)}{1+0.001*(x_i^2-2x_ix_{i+1}+x_{i+1}^2)^2})$	30,100	$[-100,100]^D$
f_{10}: Rastrigin Function	$10D+\sum_{i=0}^{D}(x_i^2-10\cos(2\pi x_i))$	30,100	$[-5.12,5.12]^D$
f_{11}: Rosenbrock Valley Function	$\sum_{i=1}^{D-1}(100(x_{i+1}-x_i^2)^2+(1-x_i)^2)$	30,100	$[-2.048,2.048]^D$
f_{12}: Schwefel Function	$\sum_{i=1}^{D}x_i\sin(\sqrt{\lvert x_i\rvert})$	30,100	$[-500,500]^D$
f_{13}: Tirronen Function	$3\exp\left(-\frac{\lvert\lvert x\rvert\rvert^2}{10D}\right)-10\exp\left(-8\lvert\lvert x\rvert\rvert^2\right)+\frac{2.5}{D}\sum_{i=1}^{D}\cos(5(x_i+(1+i\ mod\ 2)\cos(\lvert\lvert x\rvert\rvert^2)))$	30,100	$[-10,5]^D$
Bench-2			
f_1: Sphere Function	$\sum_{i=1}^{D}x_i^2$	30	$[-100,100]^D$
f_2: Schaffer's F6 Function	$0.5+\frac{\sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	2	$[-100,100]^D$
f_3: Ackley Function	$-20\exp\left\{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right\}-\exp\left\{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right\}+20+e$	30	$[-32,32]^D$
f_4: Rosenbrock Function	$\sum_{i=1}^{D}[100(x_i^2-x_{i+1})^2+(x_i-1)^2]$	30	$[-50,50]^D$
f_5: Rastrigin Function	$\sum_{i=1}^{D}\{x_i^2-10\cos(2\pi x_i)+10\}$	30	$[-5.12,5.12]^D$
f_6: Griewank Function	$\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	$[-600,600]^D$
f_7: Generalized Penalized Function	$\frac{\pi}{D}\left\{10\sin^2(\pi y_1)+\sum_{i=1}^{D-1}(y_i-1)^2\{1+10\sin^2(\pi y_{i+1})\}+(y_n-1)^2\right\}+\sum_{i=1}^{D}u(x_i,10,100,4)$ $y_i=1+\left(\frac{1}{4}\right).(x_i+1)$ $u(x,a,k,m)=\begin{cases}k.(x-a)^m, x>a\\ 0, -a\le x\le x\\ k.(-x-a)^m, x<-a\end{cases}$	30	$[-50,50]^D$
f_8: Schwefel Function	$418.9829.D-\sum_{i=1}^{D}x_i\sin(\sqrt{x_i})$	30	$[-500,500]^D$

mechanism. The proposed selection process only depends on the cost value, there is no need of prior knowledge about noise strength, noise type etc.

Exploring a multi-dimensional and multi-modal, noise-corrupted search space with a fixed-sized population of candidate solutions is challenging and it requires a great balance between the exploitative and explorative tendencies of an evolutionary search along with proper selection strategy so that the algorithm may not be deceived by noisy cost value. This requirement is nicely fulfilled by the random selection of the mutation strategy among two proposed strategies of complementary nature and by coupling with the proposed distance-based stochastic selection step. The gross performance of MDE-DS remains surprisingly consistent and statistically significantly better than majority of the state-of-the-art evolutionary methods specifically tailored for noisy optimization from existing literature.

The future works may also include a scrutinized study of the dynamics and search procedure of MDE-DS, alongside an analytical explanation of its success. Also, the mutation strategy switching technique and distance based selection may be further investigated in other optimization scenarios like for, noisy multi-objective and noisy constrained and niching optimization problems. Individual algorithmic components of MDE-DS can be integrated with the recent and improved DE frameworks proposed in works like [45]–[51] for noisy as well as static objective functions.

APPENDIX

A summary description of the set of 21 conventional benchmark functions (divided into Bench-I and Bench-II sets) collected from various literatures on single-objective noisy optimization with evolutionary computing approaches can be found in Table 12.

REFERENCES

[1] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

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

[3] Y. H. S. Tau, "Optimization of power delivery design and methodologies," in *Proc. Int. Conf. Electron. Mater. Packag.*, Kowloon, China, Dec. 2006, pp. 1–3.

[4] R. Coulom, "CLOP: Confident local optimization for noisy black-box parameter tuning," in *Advances in Computer Games*. Berlin, Germany: Springer, 2012, pp. 146–157.

[5] R. Storn and K. Price, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," Int. Comput. Sci. Inst., Berkeley, CA, USA, Tech. Rep. TR-95-012, 1995.

[6] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[7] 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.

[8] M. G. Epitropakis, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, "Enhancing differential evolution utilizing proximity-based mutation operators," *IEEE Trans. Evol. Comput.*,vol. 15, no. 1, pp. 99–119, Feb. 2011.

[9] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 482–500, Apr. 2012.

[10] 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.

[11] 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.

[12] 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.

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

[14] T. Krink, B. Filipic, and G. B. Fogel, "Noisy optimization problems—A particular challenge for differential evolution?" in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 1. Jun. 2004, pp. 332–339.

[15] S. Das and A. Konar, "An improved differential evolution scheme for noisy optimization problems," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.*, Dec. 2005, pp. 417–421.

[16] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution algorithms," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2006, pp. 2010–2017.

[17] B. Liu, X. Zhang, and H. Ma, "Hybrid differential evolution for noisy optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2008, pp. 587–592.

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

[19] G. Iacca, F. Neri, and E. Mininno, "Noise analysis compact differential evolution," *Int. J. Syst. Sci.*, vol. 43, no. 7, pp. 1248–1267, 2012.

[20] 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.

[21] H. Ma and D. Simon, "Biogeography-based optimization with blended migration for constrained optimization problems," in *Proc. ACM 12th Annu. Conf. Genet. Evol. Comput. (GECCO)*, Jul. 2010, pp. 417–418.

[22] 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.*, Mar. 2015, pp. 366–380.

[23] S.-Y. Chiu *et al.*, "Differential evolution for strongly noisy optimization: Use 1.01 n resamplings at iteration n and reach the–1/2 slope," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, May 2015, pp. 338–345.

[24] P. Rakshit and A. Konar, "Differential evolution for noisy multiobjective optimization," *Artif. Intell.*, 227, pp. 165–189, Oct. 2015.

[25] R. Kundu, R. Mukherjee, S. Das, and A. V. Vasilakos, "Adaptive differential evolution with difference mean based perturbation for dynamic economic dispatch problem," in *Proc. IEEE Symp. Differ. Evol. (SDE)*, Singapore, Apr. 2013, pp. 38–45.

[26] A. Ghosh, A. Das, B. K. Panigrahi, and A. K. Das, "A noise resilient differential evolution with improved parameter and strategy control," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, San Sebastián, Spain, Jun. 2017, pp. 2590–2597.

[27] A. Ghosh, S. Das, S. S. Mullick, R. Mallipeddi, and A. K. Das, "A switched parameter differential evolution with optional blending crossover for scalable numerical optimization," *Appl. Soft Comput.*, vol. 57, pp. 329–352, Aug. 2017.

[28] S. Das, A. Ghosh, and S. S. Mullick, "A switched parameter differential evolution for large scale global optimization—Simpler may be better," in *Proc. 21st Int. Conf. Soft Comput. (MENDEL)*, vol. 378. Brno, Czech Republic, Jun. 2015, pp. 103–125.

[29] R. Mukherjee, S. Debchoudhury, and S. Das, "Modified differential evolution with locality induced genetic operators for dynamic optimization," *Eur. J. Oper. Res.*, vol. 253, no. 2, pp. 337–355, 2016.

[30] J. Hu, J. Zeng, and Y. Tan, ''A diversity-guided particle swarm optimizer for dynamic environments,'' in *Bio-Inspired Computational Intelligence and Applications*. Berlin, Germany: Springer, 2007, pp. 239–247.

[31] P. N. Suganthan *et al.*, ''Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization,'' School Elect. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep. 2005005, 2005.

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

[33] 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.

[34] A. N. Aizawa and B. W. Wah, ''Scheduling of genetic algorithms in a noisy environment,'' *Evol. Comput.*, vol. 2, no. 2, pp. 97–122, Jun. 1994.

[35] S. Das, A. Konar, and U. K. Chakraborty, ''Improved differential evolution algorithms for handling noisy optimization problems,'' in *Proc. IEEE Congr. Evol. Comput.*, vol. 2. Sep. 2005, pp. 1691–1698.

[36] A. Caponio and F. Neri, ''Differential evolution with noise analyzer,'' in *Proc. Workshops Appl. Evol. Comput.*, Apr. 2009, pp. 1715–1724.

[37] 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, 2011.

[38] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, ''Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization,'' Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Singapore, Tech. Rep. 201212, 2013.

[39] 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,'' School Elect. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep., 2016.

[40] A. Auger and N. Hansen, ''A restart CMA evolution strategy with increasing population size,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, vol. 2. Sep. 2005, pp. 1769–1776.

[41] R. Tanabe and A. Fukunaga, ''Evaluating the performance of SHADE on CEC 2013 benchmark problems,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2013, pp. 1952–1959.

[42] R. Tanabe and A. S. Fukunaga, ''Improving the search performance of SHADE using linear population size reduction,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1658–1665.

[43] J. Brest, M. S. Maučec, and B. Bošković, ''iL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization,'' in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 1188–1195.

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

[45] 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. (CEC)*, Jun. 2017, pp. 372–379.

[46] S.-M. Guo and C.-C. Yang, ''Enhancing differential evolution utilizing eigenvector-based crossover operator,'' *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 31–49, Feb. 2015.

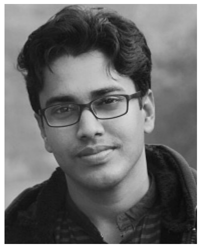
[47] L. Tang, Y. Dong, and J. Liu, ''Differential evolution with an individual-dependent mechanism,'' *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 560–574, Aug. 2015.

[48] X. Qiu, K. C. Tan, and J.-X. Xu, ''Multiple exponential recombination for differential evolution,'' *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 995–1006, Apr. 2017.

[49] 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.

[50] Y.-Z. Zhou, W.-C. Yi, L. Gao, and X.-Y. Li, ''Adaptive differential evolution with sorting crossover rate for continuous optimization problems,'' *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2742–2753, Sep. 2017.

[51] M. Z. Ali, N. H. Awad, P. N. Suganthan, and R. G. Reynolds, ''An adaptive multipopulation differential evolution with dynamic population reduction,'' *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2768–2779, Sep. 2017.



ARKA GHOSH was born in Burdwan, West Bengal, India, in 1990. He is currently pursuing Ph.D. degree with the Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, India. He has published research articles in peer-reviewed journals and international conference proceedings. His research interest includes evolutionary algorithms and data mining.

SWAGATAM DAS received the B.E. and M.E. degrees in telecommunication engineering with specilisation in control engineering and the Ph.D. degree from Jadavpur University, India, in 2003, 2005, and 2009, respectively. He is currently serving as an Associate Professor with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India. He has published one research monograph, one edited volume, and over 200 research articles in peer-reviewed journals and international conferences. His research interests include evolutionary computing, pattern recognition, multi-agent systems, and wireless communication. He is the Founding Co-Editor-in-Chief of Swarm and Evolutionary Computation, an International Journal from Elsevier. He is an Editorial Board Member of Progress in Artificial Intelligence (Springer), PeerJ Computer Science, *International Journal of Artificial Intelligence and Soft Computing*, and *International Journal of Adaptive and Autonomous Communication Systems*. He has over 12000 Google Scholar citations and an H-index of 54 till date. He has been associated with the international program committees and organizing committees of several regular international conferences including the IEEE CEC, the IEEE SSCI, the SEAL, the GECCO, and the SEMCCO. He has acted as Guest Editors for special issues in journals like the IEEE Transactions on Evolutionary Computation and the IEEE Transactions on SMC, Part C. He is the recipient of the 2012 Young Engineer Award from the Indian National Academy of Engineering. He is also the recipient of the 2015 Thomson Reuters Research Excellence India Citation Award as the highest cited researcher from India in Engineering and Computer Science category between 2010 to 2014. He has also served as or is serving as the Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics: Systems, the*IEEE Computational Intelligence Magazine*, the IEEE Access, *Pattern Recognition (Elsevier)*, *Neurocomputing (Elsevier)*, *Engineering Applications of Artificial Intelligence (Elsevier), and Information Sciences (Elsevier)*.

RAMMOHAN MALLIPEDDI received the B.Tech. degree in electrical and electronic engineering from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India, in 2005, and the master's degree in computer control and automation from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, in 2007. He is currently an Assistant Professor with the School of Electronics Engineering, Kyungpook National University, Daegu, South Korea. His research interests include evolutionary computing, artificial intelligence, image processing, digital signal processing, robotics, and control engineering.

ASIT KUMAR DAS is currently an Associate Professor with the Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah, West Bengal, India. He teaches discrete structures, database management systems, design and analysis of algorithm, and introduction to data mining. His research area of interest is data mining and pattern recognition, machine learning, bioinformatics, social networks, text and image mining, rice disease prediction, crime data analysis, audio and video data analysis.

SUBHRANSU S. DASH born in Odisha, India, in 1972. He received the degree in electrical engineering from the Institutions of Engineers, India, in 1994, the master's degree in power systems engineering from the University College of Engineering, Burla, Sambalpur University, Odisha, India, in 1996. He received the Ph.D. degree in electrical engineering from Anna University, Chennai, India in 2006. He is currently a Professor with the Department of Electrical and Electronics Engineering, SRM University, Chennai, Tamilnadu, India. He has published over 200 papers in peer-reviewed International Journals and conferences. His areas of interest are soft computing, Power system stability and smart grid.