Fighting Noise with Noise: DE with Individuals Shaking to Tackle Noisy Problems

Ali Esmailzadeh, Shahryar Rahnamayan

Abstract— **The idea of fighting noise with noise is introduced in this paper and it has been utilized to enhance Differential Evolution algorithm to solve noisy problems efficiently. The Monte-Carlo method is employed to investigate applicability of the proposed concept on a simple real-life problem. Based on the current concept, DE with population shaking (DEPS) and with individuals shaking (DEIS) are developed. Furthermore, the parent algorithm, DE, is experimentally compared with DEPS and DEIS on a benchmark test suite with nine well-known noisy functions. Detailed experimental verifications and corresponding analysis are presented for the 2D to 500D problems, for various noise levels and shaking rates.**

I. Introduction

This paper proposes the concept of *Fighting Noise with Noise*. First, it employs Monte-Carlo method to simulate a real-world problem (i.e., shooting on a noisy target) for revealing that how intentional embedding of a second noise can be helpful to solve a noisy problem more efficiently. The main claim is that, at least, there are some interesting circumstances which the combination of the two noises can result a better performance in solution favor. Tackling noisy problems appropriately is always a valuable attempt in optimization field. As a case study in the current work, Differential Evolution, a well-known evolutionary algorithm, is chosen to be enhanced by the mentioned concept to tackle with noisy problems. In this direction, two schemes are considered, namely, DE with population shaking (DEPS) and with individuals shaking (DEIS). The shaking process is based on adding uniform noise to the current population according to a shaking rate.

Dealing with noisy fitness functions in evolutionary algorithms has been addressed by some authors in this field, such as evolutionary programming (EP) [3], genetic algorithm (GA) [4], particle swarm optimization (PSO) [5], and differential evolution (DE) [6], [16]. *Re-sampling* and *Thresholding* are well-known methods to overcome the noisy fitness evaluation [7], [8]. 1) Re-sampling suggests evaluating of the same candidate solution for N times and approximating of the true fitness value by averaging. N should be determined properly to achieve a reasonable tradeoff between accurate evaluation of fitness value and computation cost. 2) Thresholding method is applied on selection step. According to this method, a parent can only be replaced by an offspring if fitness value of offspring is larger than a threshold value τ. The main problem with this method is finding an optimal static value or modified adaptation rule for τ. Proposing a self-adaptive algorithm to work with optimal value of N or τ is challenging task because noise is generally unpredictable.

This paper is organized as follows: in section II, the idea of fighting noise with noise is introduced and explained. The classical DE is briefly reviewed in section III. The proposed case studies (DEPS and DEIS) are presented in section IV. The experimental results and corresponding analysis are given in section V. Finally, the work is concluded in section VI.

Faculty of Engineering and Applied Science, University of Ontario Institute of Technology (UOIT), 2000 Simcoe Street North, Oshawa, Ontario, L1H 7K4, Canada (phone: 1-(905)-721-8668 ext. 3843, fax: 1-(905)-721-3178, emails: (corresponding author) ali.esmailzadeh@uoit.ca; shahryar.rahnamayan@uoit.ca).

II. Fighting Noise with Noise

In this paper, we present a new concept that could be utilized to solve noisy problems efficiently. This section explains the idea of *fighting noise with noise* in details and then conducts Monte-Carlo simulations for a real-world scenario. Experimentally, it will be shown in a noisy environment, introducing an intended noise *could* result in favor of targeting the solution. It means that the combination of the noises could be beneficial depending on the problem and environment's circumstances.

The existence of the noise is not a new phenomenon. In fact, in the nature and majority of real-world problems the fingerprints of noise are visible. The reference to noise in this concept refers to the elements of interference with any real-world actions. For instance, satellite communications or radio wavelengths are transferring constantly in noisy environments, where other elements in the nature could disturb and disrupt their transmissions. Noise is not only bounded to wireless communications and networks; indeed, the effects of noise can be observed even in a simple target-and-shoot situation. In this example, the *human factors*, namely, emotional stress, eye-sight problem, shaking hands or *environmental factors* such as wind, humidity, and temperature can all contribute to introduce a *noise* in making a perfect *hit* by the shooter. Generally speaking, any interference of an activity can be affected by *noise*, and such interferences do always exist in almost all real-world problems. That is why, in all engineering design problems, noise is always taken into consideration. The objective of a design is to make the effect of noise as small as possible on the operation of the entities. Therefore, any concept and method which could possibly reduce the effects of the noise is considered valuable.

The natural instinct of human being dictates the interpretation of that, any kind of noise is attached with a negative effect. It is simply considered as the disturber

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

and a negative factor which never assists an operation in a positive way. Although, this can be preserved naturally as common-sense, it is indeed interesting to see if this is always true. Even though the human mind perseveres noise as a negative phenomenon, there cannot be a general proof for this assumption, but simply that is based on human's exuberance. In fact, since every real-world problem has its own various parameters and elements involved in its process, it is nearly impossible or complex to claim that any noise with any type is indeed disturber or ineffective for *all* operations in every nature or application oriented fields. In [1], [2], Deborah M. Gordon[1] mentioned that the Ants' communications (performed by their antennas) are noisy because no ant can do any sophisticated counting, but their food seeking and collecting processes are done without any problem. She said *"... so what's interesting about this system is that It's variable. It's noisy. And, in particular, in two ways. The first is that the experience of the ant – of each ant – can't be very predictable. Because the rate at which ants come back depends on all the little things that happen to an ant as it goes out and does its task outside. And the second thing is that an ant's ability to assess this pattern must be very crude, because no ant can do any sophisticated counting. So, we do a lot of simulation and modeling, and also experimental work, to try to figure out how those two kinds of noise combine to, in the aggregate, produce the predictable behavior of ant colonies. Again, I don't want to say that this kind of haphazard pattern of interactions produces a factory that works with the precision and efficiency of clockwork."* We say maybe combination of two noises (noisy return time and noisy communication [1]) makes the search doable for ants. However, one can only wonder if this is really correct. It raises following fundamental question: *Are there any cases or situations in which having an extra noise could actually benefit or at the minimum, give an alternative to minimize the effect of the original noise?* To further examine this question, we first mention following example.

Let's suppose we consider the *target-shooter* example mentioned previously. In our example, we consider *shaking* of target and shooter's hands as a *noise*. For both the target and the shooter, there are two scenarios: *shaking* (i.e., noisy) and *non-shaking* (i.e., precise). The objective of this example is helping to a better understanding of fighting noise with noise concept. Let us consider the target object being shaky at all times, we can assume that it is an enemy fighter jet flying in front and it constantly has some shaking to the sides. Would we increase our chances to make a precise hit if, 1) we have some shakes in our jet, or 2) would the chance be greater if we make a shake-free shoot?

As another example, we can simply think of a shaky target and competition between two shooters: a soldier and a machine gun operated by a computer in order to control the amount of shaking. We want to consider that if the soldier cannot make a precise shot, due to other factors (environmental or human based factors), then would the chance of hitting the target be greater if we let the computer-controlled gun to shoot with an applied shake, or would the soldier have a better chance to hit the shaking target?

In order to investigate this concept, we have used Monte-Carlo method to simulate a comparison between shake-free and shaky shooters, both on a shaky target. It has been conducted by defining a uniform random point inside of a circle (r=1) which indicates our noisy target, and we defined two kinds of shooting, 1) shake-free shooting towards the center of the circle, and 2) shaking hands which shots uniform randomly toward the same circle. The diameter of the target circle was divided to 100 intervals; therefore, the test was done for 100 shake-free shoots and also shaky (random) shoots on that range. Moreover, the simulation was repeated 10^5 times per point (interval) and the average has been measured and reported. According to obtained simulation results, we conclude that in noisy problems (the *shaky target*, in our example), there is certain range that an shake-free shooter with a small tolerance can hit and have the higher chance of making the hit (closer to the noisy target). On the other hand, if the shooter has a higher shooting tolerance, then the chance of hitting the target is greater if we use shaking shooter to hit the target. The Fig. 1 demonstrates the boundary where the shake-free shooter has a higher chance of hitting the target. Moreover, it shows the rest of the area of the target where the shaky-shooter would have a higher chance of hitting the target than the free shaking shooter.

Fig. 1 illustrates the probabilities of hitting target by shake-free hands (p1) and shaking hands (p2). As seen, none of the shooters are dominant over the whole range; for the inner part, p1 shows a higher probability and for the outer range, p2 illustrate a higher chance. However, as the shooter gets closer to the center of the range, the probability of hitting for shake-free shooter is increased. The current results demonstrate that, at least, there are some situations which having noise (shaking) can help to achieve a better result, in term of accuracy.

To further our investigation of the concept, we want to examine how results differ if the amount of the noise for shaky-shooter is changed (the same level of noise for target but smaller noise (smaller circles) for shaky shooter). In this case, we change the amount of the noise for shaky shooter by decreasing its range within the target (step size is set to 0.1). Fig. 2 presents the probabilities of closeness to noisy target for both shooters with different amount of the uniform noise for shaky shooter (r shows the range of the noise, which is the radius of the circle which shaky shooting are limited in that circle). The significant data for us in this experiment are the *range* values that are acquired for each r value. We would like to see what the range or area is for the shake-free and shaky shooter, within the region of the target. The Table

[1]She is an Ant biologist. Contrary to the popular notion that colonies have evolved into efficient, organized systems, she has instead discovered that the long evolution of the ant colony has resulted in a system driven by accident, adaptation and the chaos and "noise" of unconscious communication [TED.COM].

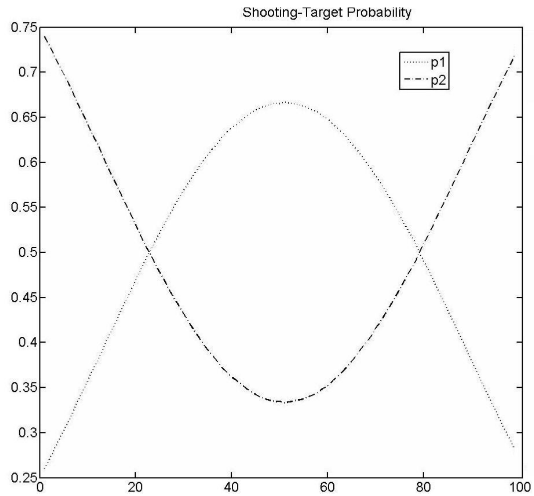


Fig. 1. Probability of closeness of a *hit* to the noisy Target (y-axis) vs. the range on the target (x-axis), in the target-shooting example. The curve p1 indicates the probability for shake-free shooter; p2 represents the probability for shaky-shooter.

I presents those ranges by indicating the cross points of the probabilities (Fig. 2) for indicated radii (i.e., r=0.1, 0.2, 0.3, 0.4, and 0.5).

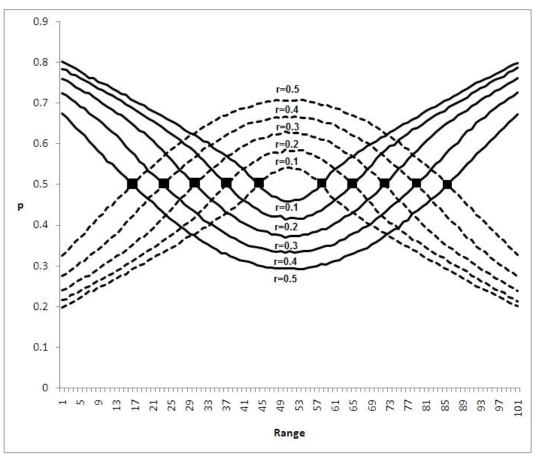


Fig. 2. Probability of the closeness to noisy target for variable shaking rates for shaky-shooter. The solid lines represent the shaky shooter's probabilities; the dashed lines represent the shake-free shooter's probabilities.

Similarly, in Fig. 3, it shows the average distance of hits from noisy targets for shaking-free shooting and shaky-shooting.

By examining the cross-points in Fig. 3, we can see that the cross-points in this graph have the same range values as in Fig. 2 (given in Table I).

Now, if we consider a specific r from the Fig. 2, we can draw a shoot-target illustration based on r=0.4 (the same amount of the noise for the target and shaky-shooter). As seen in Fig. 4, the inner circle of the target (A_1) represents the region for higher probabilities for shake-free shooter; similarly, the area between two circles (A_2) represents the region for higher probabilities for shaky shooter. Therefore, if the shooter's tolerance allows him/her to hit the inner circle, then it is better to shoot towards the center of the circle; otherwise, it is better to shoot with shaky hands.

TABLE I

CROSS POINTS FOR PROBABILITIES (P1 AND P2) FOR FIG. 2. WHICH INDICATES WINNER RANGE FOR THE SHAKE-FREE SHOOTER.

r	Approximate Cross-points	Winning range for shake-free shooter
0.1	0.44, 0.56	12%
0.2	0.35, 0.65	30%
0.3	0.30, 0.70	40%
0.4	0.24, 0.78	56%
0.5	0.16, 0.84	68%

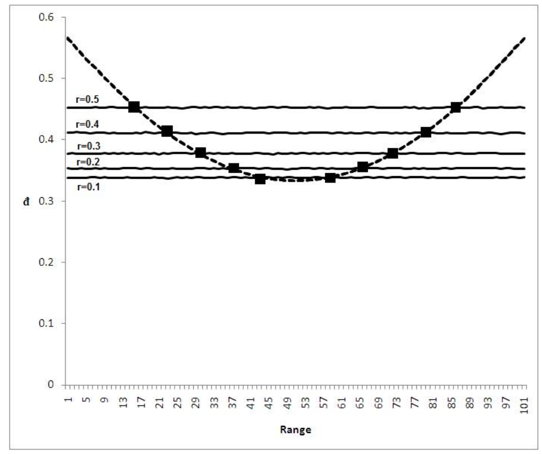


Fig. 3. Average Distance of hits from noisy targets for shaking-free shooting and shaky shooting.

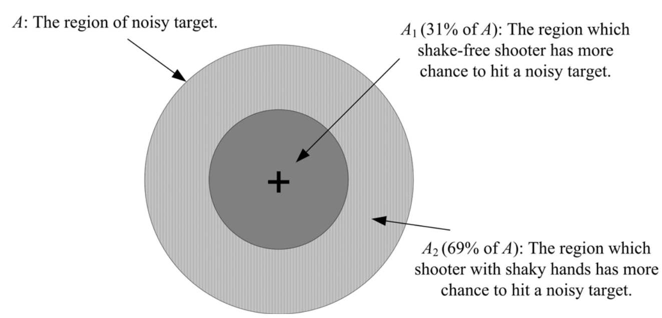


Fig. 4. Illustration of region for noisy target (A) and the regions for a better chance of hit for shake-free shooter (A_1) and shaky shooter (A_2).

Fig. 4 indicates that if tolerance of shake-free shooter is so high which for the majority of shootings it hits area A_2, then it would be better to shoot with shaky hands, which will increase the chance of *hitting* of noisy target.

Based on the results shown above, the conclusion is that there *are* in-fact cases which adding additive noise could assist in solving a noisy problem more efficiently. The current results motivate us to put the fighting noise with noise concept on track by taking a real-world case study from global optimization field and conducting an experimental verification.

Before explaining the proposed approach, which utilizes noise in Differential Evolution (DE) to solve noisy problems effectively, we need to briefly explain the parent algorithm, DE.

III. The Classical DE

Differential Evolution (DE) is a population-based and directed search method [9], [10]. Like other evolutionary algorithms, it starts with an initial population vector, which is randomly generated when no preliminary knowledge about the solution space is available.

Let us assume that $X_{i,G}(i = 1, 2, ..., N_p)$ are solution vectors in generation G (N_p : population size). Successive populations are generated by adding the weighted difference of two randomly selected vectors to a third randomly selected vector.

For classical DE (indicated by $DE/rand/1/bin$), the mutation, crossover, and selection operators are straightforwardly defined as follows:

Mutation - For each vector $X_{i,G}$ in generation G a mutant vector $V_{i,G}$ is defined by

$$V_{i,G} = X_{a,G} + F(X_{b,G} - X_{c,G}), \quad (1)$$

where $i = \{1, 2, ..., N_p\}$ and a, b, and c are mutually different random integer indices selected from $\{1, 2, ..., N_p\}$. Further, i, a, b, and c are different so that $N_p \geq 4$ is required. $F \in (0, 2]$ is a real constant which determines the amplification of the added differential variation of $(X_{b,G} - X_{c,G})$. Larger values for F result in higher diversity in the generated population and lower values cause faster convergence.

Crossover - DE utilizes the crossover operation to generate new solutions by shuffling competing vectors and also to increase the diversity of the population. For the classical version of the DE ($DE/rand/1/bin$), the binary crossover (shown by 'bin' in the notation) is utilized. It defines the following trial vector:

$$U_{i,G} = (U_{1i,G}, U_{2i,G}, ..., U_{Di,G}), \quad (2)$$

where $j = 1, 2, ..., D$ (D : problem dimension) and

$$U_{ji,G} = \begin{cases} V_{ji,G} & \text{if } rand_j(0,1) \leq C_r \vee j = k, \\ X_{ji,G} & \text{otherwise.} \end{cases} \quad (3)$$

$C_r \in [0, 1]$ is the predefined crossover rate constant, and $rand_j(0, 1)$ is the j^{th} evaluation of a uniform random number generator. $k \in \{1, 2, ..., D\}$ is a random parameter index, chosen once for each i to make sure that at least one parameter is always selected from the mutated vector, $V_{ji,G}$. Most popular values for C_r are in the range of $(0.4, 1)$ [6].

Selection - The approach that must decide which vector ($U_{i,G}$ or $X_{i,G}$) should be a member of the next (new) generation, $G + 1$. For a maximization problem, the vector with the higher fitness value is chosen. There are other variants based on different mutation and crossover strategies [11].

IV. A Case Study: Proposing DE with Additive Noise to Solve Noisy Problems effectively

The work performed in this paper can be considered as a case study in order to investigate the practicality of the mentioned idea (Fighting noise with noise). So, the conducted experiments in the following sections are on limited cases and - needless to say - do not represent a general proof.

As the title of the paper indicates, the idea is adding noise to the population in order to solve noisy problems by DE. Therefore, we propose DE with shaky population. In fact, we will add noise to DE's population based on a predefined shaking rate.

A. Population vs. Individual Shaking

In order to implement the idea presented in the previous section, we take two scenarios. In one scenario, we give the same additive noise, σ^2, to the entire DE population as a fix group shaking, we call that *DE with Shaking Population* (DEPS), which means all individuals in the population experience the same amount of the noise. In the second scenario, we apply independent various additive noise, σ^2, to each individual of the population, that is why we call that *DE with Individuals Shaking* (DEIS). The objective of these two scenarios is to investigate that which kind of shaking (population vs. individual) can present superior results. The shaking will be applied according to a predefined Shaking Rate (S_r). The population shaking process is exactly similar to generation jumping in Opposition-Based Differential Evolution (ODE) [12]. Accordingly, the formula for the Shaking Population of both DEPS and DEIS scenarios are as follows:

- DEPS

$$SAF = \sigma^2 \times rand(0, 1),$$
$$SP = SAF + Population,$$

where SP indicates Shaking Population

- DEIS

$$SAF = \sigma^2 \times rand_j(0, 1),$$
$$SP_j = SAF + Population_j,$$

for $j = 0$ to Np (population size)

In Figures 5 and 6, newly added or changed blocks in the DE are emphasized by shaded blocks and are explained in details as follows:

As a population initialization, we start with a random population, $P(n)$. In both algorithms, as seen, the entire population for DEPS and the individuals separately for DEIS are shaken based on a predefined shaking rate. As illustrated in Fig. 7, we apply a uniform noise (shake) to the entire current population and save that as a new *Shaken Population*. Similarly, in Fig. 7, we apply a shake to the current population; however, in this case, the shake is a unified-random value per individual. After giving a shake to current population (different for DEPS and DEIS), the N_p fittest individuals are selected from union of the current population and shaken one; then we follow the original DE steps, namely, mutation, crossover and selection.

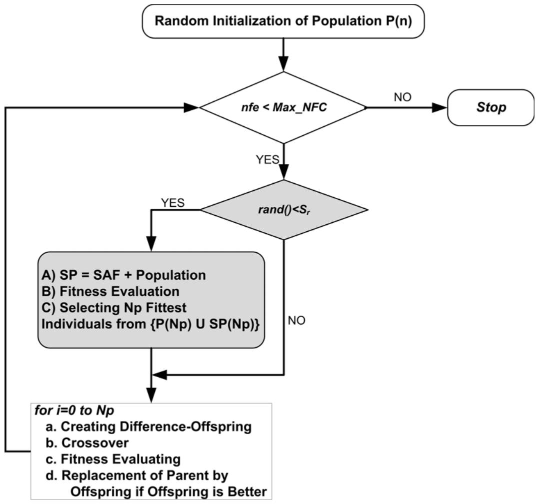


Fig. 5. DE with Population Shaking (DEPS). New blocks are illustrated by shading. The SAF (Shaking Amplification Factor) value is set to $\sigma^2 \times$rand().

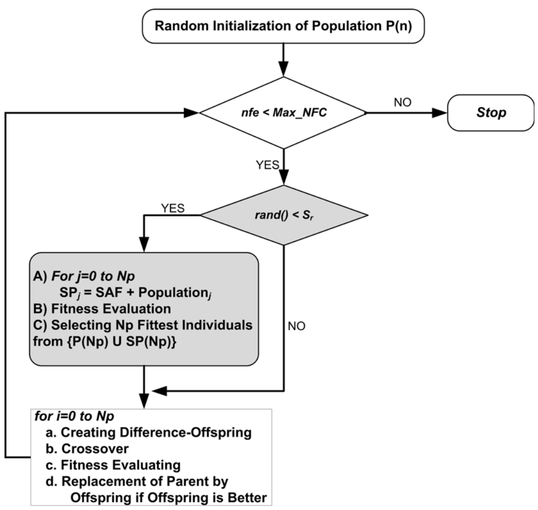


Fig. 6. DE with Individuals Shaking (DEIS). New blocks are illustrated by shading. The SAF (Shaking Amplification Factor) value is set to $\sigma^2 \times$rand().

V. Experimental Verifications

The same experiment strategy, benchmark functions, parameter settings, and comparison criteria have been chosen from ODE paper to solve noisy problems [16].

A. Benchmark Functions

Following functions are well-known benchmark functions for minimization [6], [8], [13], [16]. The noisy version of each benchmark function, is defined as:

$$f_n(\vec{x}) = f(\vec{x}) + N(0, \sigma^2), \tag{4}$$

where $f(\vec{x})$ is the noise-free function; $f_n(\vec{x})$ is the corresponding noisy function; and $N(0, \sigma^2)$ is normal, zero maen distribution with and deviation σ. For all benchmark functions the minima are at the origin or very close to the

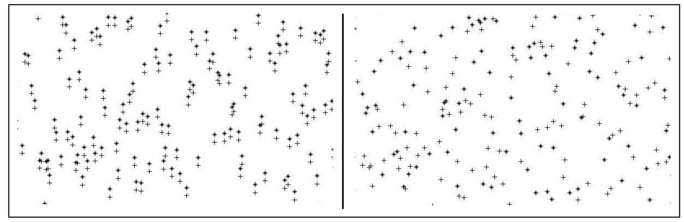


Fig. 7. A sample current population (indicated by $+$) and its corresponding shaken population (indicated by $*$) are illustrated for DEPS (left) and DEIS (right) in 2D.

origin. Except for f_5 (Levy No. 5 function), its minima is at $\vec{x} = [-1.3068, 1.4248]$ with $f(\vec{x}) = -176.1375$.

- Sphere (50D)

$$f_1(x) = \sum_{i=1}^{D} {x_i}^2, \text{with} -100 \leq x_i \leq 100$$

- Rosenbrock (50D)

$$f_2(x) = \sum_{i=1}^{D} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2], \quad \text{with} -50 \leq x_i \leq 50$$

- Rastrigin (50D)

$$f_3(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10], \quad \text{with} -5.12 \leq x_i \leq 5.12$$

- Griewangk (50D)

$$f_4(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1, \quad \text{with} -600 \leq x_i \leq 600$$

- Levy No. 5 (2D)

$$f_5(x) = \sum_{i=1}^{5} i\cos[(i+1)x_1 + i] \times \sum_{j=1}^{5} j\cos[(j+1)x_2 + j] + (x_1 + 1.42513)^2 + (x_1 + 0.80032)^2, \quad \text{with} -10 \leq x_i \leq 10$$

- Beale (2D)

$$f_6(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2, \quad \text{with} -10 \leq x_i \leq 10$$

- Ackley (50D)

$$f_7(x) = -20e^{-0.2\sqrt{\frac{\sum_{i=1}^{D} x_i^2}{D}}} - e^{\frac{\sum_{i=1}^{D} \cos(2\pi x_i)}{D}} + 20 + e, \quad \text{with} -32 \leq x_i \leq 32$$

- Schaffer's f_6 (2D)

$$f_8(x) = 0.5 - \frac{(\sin\sqrt{x^2+y^2})^2 - 0.5}{(1.0+0.001(x^2+y^2))^2}, \quad \text{with} - 100 \leq x_i \leq 100$$

- De Jong's f_4 with noise (50D)

$$f_9(x) = \sum_{i=1}^{D} i{x_i}^4 + rand(0,1), \quad \text{with} - 1.28 \leq x_i \leq 1.28$$

B. Simulation Strategy

Similar to other studies in the evolutionary optimization [6], [8], [15], [16], for all conducted experiments, trials are repeated 30 times per function per noise deviation. Each run is continued up to 10^5 function calls and then mean and standard deviation of the best fitness values are reported. Re-sampling and thresholding techniques [7] are not applied in this paper.

For each algorithm (DE, DEPS, DEIS), we will conduct three different experiments series; including tests related to the dimension of the considered problems, Shaking Rates (S_r) analysis, and amount of noise level in benchmark problems.

C. General Control Parameter Settings

The settings mentioned below would be the same for all the experiments. All the common parameters of the DE, DEPS, and DEIS simulations are set to the same values, in order to have a fair comparison. The parameter settings are listed as follows [16].

- Population size, $N_p = 100$
- Differential amplification factor, F=0.5
- Crossover probability constant, Cr=0.9
- Strategy, $DE/rand/1/bin$
- Maximum function calls (which determines termination criteria), $\text{MAX}_{\text{NFC}}=10^5$

Experiment Series 1: DE vs. DEPS and DEIS

In this experiment we want to compare DE, DEPS, and DEIS in term of solution accuracy.

For the current experiment, we have

- Noise factor, σ^2=0.5 (same for all functions)
- Shaking rate constant, S_r=0.3 (for DEPS and DEIS)

Results Analysis - The results are presented in Table II. It is apparent that DE outperforms DEPS and DEIS on 4 functions (namely, f_4(50D), f_5(2D), f_6(2D), and f_9(50)), while DEPS performs better than DE on five functions (out of nine), by closely checking the results we can see that for the cases which DEPS performs better, the accuracy is of the solution much higher than DE's; DEIS outperforms DE just on two functions (the same on one function).

Performance comparison graphs for DE, DEPS and DEIS are illustrated in Fig. 8 for the current experiments.

Experiment Series 2: Testing on High Dimensional Problems

TABLE II

MEAN $\pm$ (STANDARD DEVIATION) OF THE BEST FITNESS VALUE, BY HAVING CONSTANT MEDIUM NOISE LEVEL, σ^2=0.5, S_r=0.3. THE BEST RESULT FOR EACH CASE IS HIGHLIGHTED IN **BOLDFACE**.

Function	DE	DEPS	DEIS
f_1(50D)	$0.96 \pm (0.217)$	**0.85 ± (0.20)**	$0.91 \pm (0.21)$
f_2(50D)	$59.06 \pm (22.478)$	**42.49 ± (67.89)**	$112.86 \pm (184.02)$
f_3(50D)	$376.00 \pm (15.355)$	$90.341 \pm (20.93)$	**73.43 ± (20.97)**
f_4(50D)	**1.88 ± (0.21)**	$1.98 \pm (0.28)$	$1.96 \pm (0.27)$
f_5(2D)	**-176.09 ± (0.04)**	$-176.08 \pm (0.06)$	$-176.08 \pm (0.06)$
f_6(2D)	**0.06 ± (0.07)**	$0.065 \pm (0.08)$	$0.07 \pm (0.07)$
f_7(50D)	$20.039 \pm (4.07)$	**3.81 ± (0.43)**	$3.90 \pm (0.46)$
f_8(2D)	$0.500 \pm (0.00)$	**0.50 ± (0.00)**	$0.50 \pm (0.00)$
f_9(50D)	**0.36 ± (0.15)**	$0.49 \pm (0.19)$	$0.41 \pm (0.18)$

Similar to the previous experiment, the medium Noise value of σ^2=0.5 and Shaking Rate of S_r=0.3 are applied. In the current case, the only difference is the dimension of the problems which is set to higher values of 2D (100) and 10D (500). It will give us a better understanding about DE, DEPS and DEIS performances to solve high dimensional problems. Since functions f_5, f_6, and f_8 are non-scalable functions they have not been considered for the current test.

Results Analysis - The results are summarized in Table III. For D=100, DE and DEPS just present best results on one function, at the same time, DEIS outperforms both DE and DEPS on four functions (out of six).

For D=500, DE on none of the functions performs better than others, DEPS just on two and DEIS again on four functions is superior.

It seems, on large-scale problems DEIS performs better than DEPS, and much better than DE.

Experiment Series 3: Testing with Variable Noise Values

Now, that is the time to analyze the affect value of noise (σ^2) in the problems on the performance of DE, DEPS, and DEIS. The experiments are conducted for noise values 0, 0.25, 0.5, 0.75.

Result Analysis - The results for DE, DEPS and DEIS are analyzed separately as follows:

For σ^2=0.0 (noise-free) results in Table IV show that there is no big difference among DE, DEPS, and DEIS's results. This indicates that in noise-free problems, in general it does not make a big difference to use DE, DEPS or DEIS.

For low noise, σ^2=0.25, again, DE is clearly beaten by DEPS and DEIS. Moreover, DEIS outperforms DEPS on seven functions.

In the medium noise, σ^2=0.5, DE, DEPS, and DEIS perform better than two others on 2, 3, 4 functions, respectively.

For higher noise level, σ^2=0.75, DE, DEPS, and DEIS perform better than two others on 2, 3, 3 functions, respectively (same on one function).

In general, DEPS and DEIS have shown to have an improvement to the original DE where there is any level of noise (σ^2). For the overall comparison of the three algorithms, in most cases the results of DEPS are very close and similar to the results of DEIS. Which means these two methods have a tight competition most of the time. In comparison to original DE, both of DEPS and DEIS methods introduce a major improvement to DE. In other words, DEPS and DEIS beat DE on the majority of functions on the utilized test suite.

Experiment Series 4: Variable Shaking Rates

In this section, the effects of different Shaking Rates (S_r) with a constant noise value setting is examined in order to determine the impact of the Shaking.

TABLE VI

SUMMARY OF THE BEST RESULTS FROM TABLE V.

Function	Winner Algorithm	Winner S_r
f_1(50D)	DEPS	0.3
f_2(50D)	DEIS	0.1
f_3(50D)	DEIS	0.7
f_4(50D)	DEIS	0.1
f_5(2D)	DEPS	0.7
f_6(2D)	DEPS	0.7
f_7(50D)	DEPS	0.3
f_8(2D)	DEIS	0.5
f_9(50D)	DEPS	0.7

Result Analysis - In this scenario of simulations, only the Shaking Rate (S_r) is changed, while the noise level is unchanged (σ^2=0.5). Since S_r only applies to the *Shaking* part of the code, therefore, the DE results should be the same for all S_r settings. In the table V, we only pick one *best* result for each function, among all the methods with different shaking rates (the bolded ones). As seen, all the best results have always been from either DEPS or DEIS method.

Moreover, according to the summary Table VI, as the shaking rate increases, the best results are either with DEPS or DEIS; in other words, DE is always beaten by DEPS or DEIS. This shows that in this case study, DEPS and DEIS perform better than original DE in solving noisy problem and even by varying shaking rate, this situation is not changed in favor of DE. In addition, it is apparent that the optimal value of S_r for DEPS and DEIS is problem oriented.

VI. CONCLUDING REMARKS

The DE method is well-known for solving global optimization problems. However, for noisy problems DE could take longer to converge. We aimed to know if noise in a problem has always a negative effect or not. More specifically, we wanted to find out that wether are there any cases in which having noise could be beneficial in finding the answer? This concept was investigated by Monte-Carlo simulation.

We used a case study for this purpose, by testing for variable dimensions, variable noise level and variable shaking rate, in three different experiment series. Based on the proposed concept, two shaking population scheme for DE were introduced. First one by shaking entire population (DEPS), and second one by shaking individuals independently (DEIS). On majority of functions, DEPS and DEIS presented better results than DE. In other words, applying noise to fight with noisy problems showed some preliminary promising results. Furthermore, the results confirm that to solve noisy high dimensional problems; DE with individuals shaking (DEIS) performs much better than population shaking scheme (DEPS).

Obviously, there are much more work that still has to be done in this direction and more detailed experiments involving other control parameters should be conducted. As directions for our future work, the Shaking Rate (S_r) parameter has to be analyzed in more depth. Moreover, that is better to test proposed algorithms on more comprehensive test suites with much complex and large-scale search spaces.

REFERENCES

[1] D.M. Gordon, S. Holmes, S. Nacu, *The short-term regulation of foraging in harvester ants*, Behavioral Ecology Vol. 19, pp. 217-222, 2008.

[2] D. Gordon, *How Do Ants Know What to Do?*, http://blog.ted.com/2008/01/deborah_gordon.php, Jan. 2008.

[3] L.J. Fooel, A.J. Owens, M.J. Walsh, *Artificial Intelligence through Simulnted Evolution*, John Wiley & Sons, New York, 1966.

[4] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, New York: Addison-Wesley, 1989.

[5] J. Kennedyj and R.C. Eberhartr, *Particle Swarm Optimization*, Proceedings of the I995 IEEE International Conference on Neural Network (Perth, Australia, IEEE Service Center, Piscataway, NI), Vol. 4, pp. 1942-1948, 1995.

[6] S. Das, A. Konar, Uday K. Chakraborty, *Improved Differential Evolution Algorithms for Handling Noisy Optimization Problems*, Proceedings of IEEE Congress on Evolutionary Computation, CEC2005, Vol. 2, pp. 1691-1698, 2005.

[7] Yaochu Jin and Jürgen Branke, *Evolutionary Optimization in Uncertain Environments- A Survey*, IEEE Transactions on Evolutionary Computatioon, Vol. 9, No. 3, pp. 303-317, June 2005.

[8] T. Krink, B. Filipič, Gary B. Fogel, *Noisy optimization problems - A Particular Challenge for Differential Evolution?*, Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004, Vol. 1, pp. 332-339, 2004.

[9] K. Price, *An Introduction to Differential Evolution*, In: D. Corne, M. Dorigo, F. Glover (eds) New Ideas in Optimization, NcGraw-Hill, London (UK), pp. 79-108, 1999, ISBN:007-709506-5.

[10] Godfrey C. Onwubolu and B.V. Babu, *New Optimization Techniques in Engineering*, Berlin ; New York : Springer, 2004.

[11] R. Storn and K. Price, *Differential Evolution- A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces*, Journal of Global OPtimization 11, pp. 341-359, 1997.

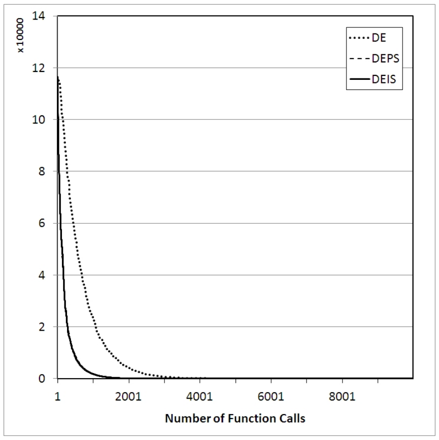
[12] S. Rahnamayan, H.R. Tizhoosh, M.M.A Salama, *Opposition-Based Differential Evolution*, IEEE Transactions on Evolutionary Computation, Volume 12, Issue 1, Feb. 2008, pp. 64-79.

[13] X. Yao, Y. Liu, G. Lin, *Evolutionary Programming Made Faster*, IEEE Transactions on Evolutionary Computatioon, Vol. 3, No. 2, pp. 82-102, 1999.

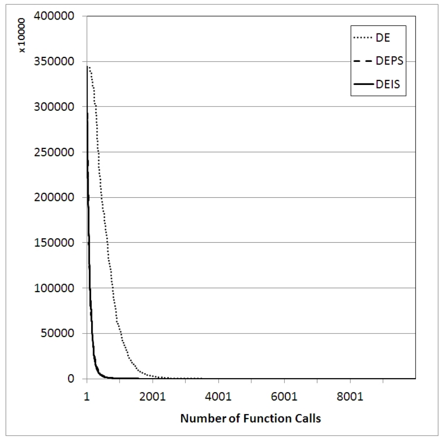
[14] K. Price, R.M. Storn, J.A. Lampinen, *Differential Evolution : A Practical Approach to Global Optimization (Natural Computing Series)* Springer; 1st Edition, 2005, ISBN: 3540209506.

[15] J. Vesterstrøm and R. Thomsen, *A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems.* Proceedings of the Congress on Evolutionary Computation (CEC'04), IEEE Publications, Vol. 2, pp. 1980-1987, 2004.

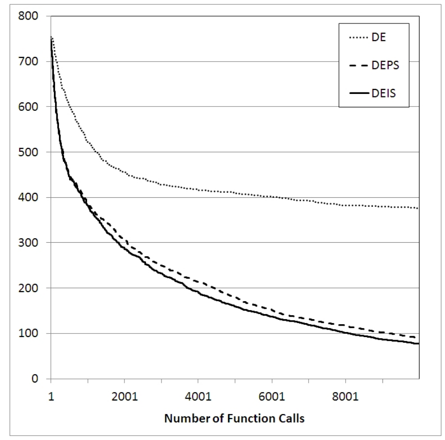
[16] S. Rahnamayan, H. R. Tizhoosh, M. M.A. Salama, *Opposition-Based Differential Evolution for Optimization of Noisy Problems* Proceedings of the Congress on Evolutionary Computation (CEC'06), IEEE Publications, pp. 6756-6763, 2006.



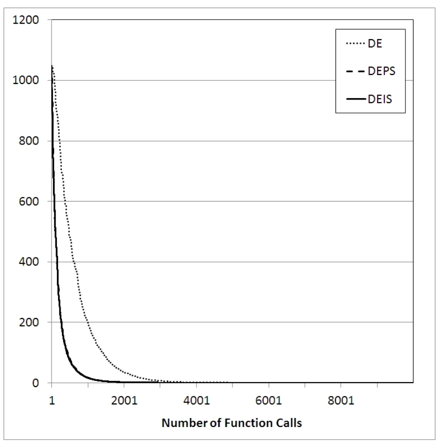
(a) f_1(50D), Sphere Function



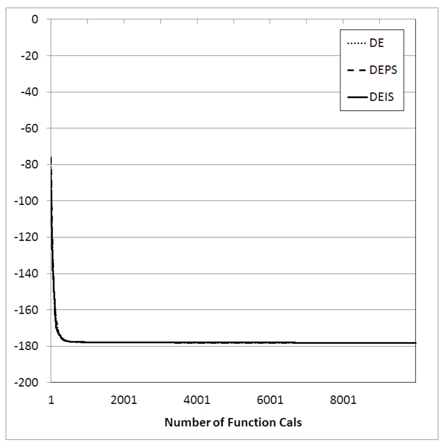
(b) f_2(50D), Rosenbrock Function



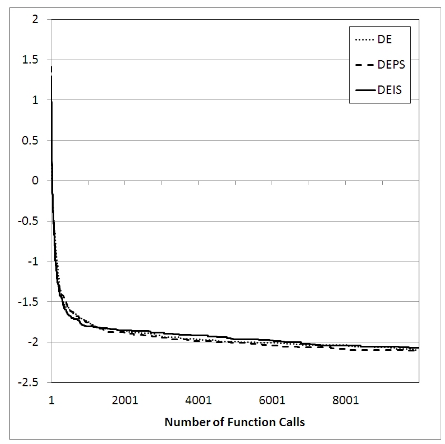
(c) f_3(50D), Rastrigin Function



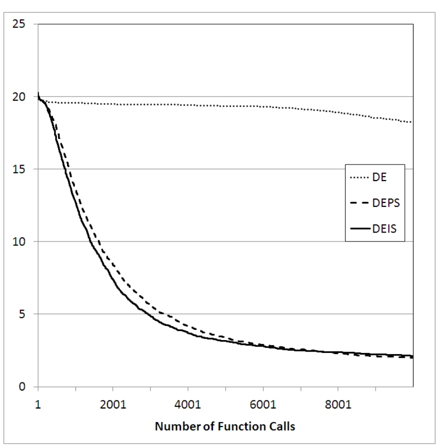
(d) f_4(50D), Griewangk Function



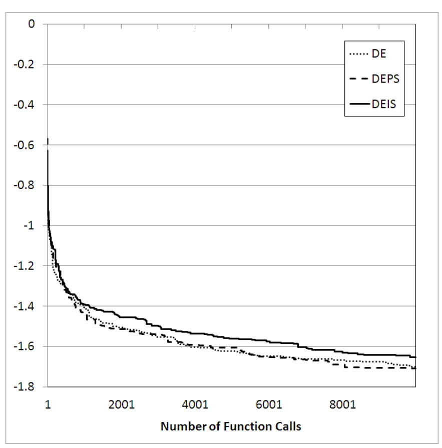
(e) f_5(2D), Levy No. 5 Function



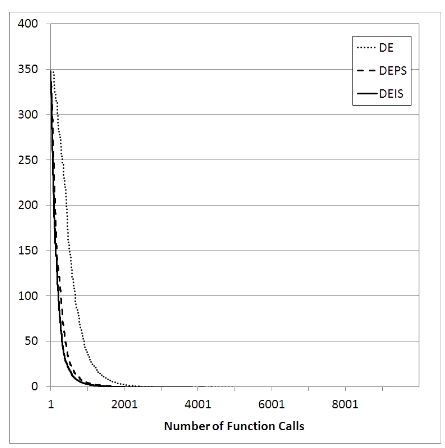
(f) f_6(2D), Beale Function



(g) f_7(50D), Ackley Function



(h) f_8(2D), Schaffer's f_6 Function



(i) f_9(50D), De Jong's f_4 Function with noise

Fig. 8. Performance graphs for DE, DEPS and DEIS. Best-fitness value vs. number-of-function-calls (NFC) is plotted (noise deviation: $\sigma^2 = 0.5$, Shaking Rate: S_r=0.3). Experiments have been repeated 30 times to plot according to the average values.

TABLE III

MEAN ± (STANDARD DEVIATION) OF THE BEST FITNESS VALUE IN HIGH DIMENSIONS OF 100 AND 500, BY HAVING CONSTANT MEDIUM NOISE LEVEL, σ^2=0.5, S_r=0.3.

	D=100			D=500		
F	DE	DEPS	DEIS	DE	DEPS	DEIS
f_1	11.21 ± (2.57)	2.26 ± (0.39)	**2.12 ± (0.37)**	$14.94E5$ ± ($29.37E3$)	$13.37E3$ ± (2347.20)	**12.34E3 ± (1727.10)**
f_2	$21.92E4$ ± ($73.30E4$)	$10.40E2$ ± 467.39)	**905.80 ± (601.31)**	$53.38E9$ ± ($14.88E8$)	**2.16E7 ± (9.48E6)**	$21.76E6$ ± ($53.97E5$)
f_3	870.38 ± (20.98)	315.79 ± (80.93)	**225.08 ± (73.87)**	8686.20 ± (76.16)	2555.50 ± (485.92)	**2217.00 ± (367.96)**
f_4	**2.87 ± (0.27)**	3.95 ± (0.73)	3.44 ± (0.61)	$13.45E3$ ± (264.36)	228.42 ± (47.93)	**201.76 ± (37.07)**
f_7	21.58 ± (0.20)	10.73 ± (3.92)	**9.69 ± (3.24)**	21.65 ± (0.14)	**17.83 ± (1.19)**	18.11 ± (1.08)
f_9	1.04 ± (0.27)	**0.97 ± (0.28)**	0.97 ± (0.22)	56177.00 ± (1652.60)	129.39 ± (47.47)	**98.50 ± (23.97)**

TABLE IV

MEAN ± (STANDARD DEVIATION) OF THE BEST FITNESS VALUE FOR EACH VARIABLE NOISE LEVEL, σ^2= 0.0, 0.25, 0.5, AND 0.75, BY HAVING S_r=0.3.

	σ^2=0.0			σ^2=0.25		
Function	DE	DEPS	DEIS	DE	DEPS	DEIS
f_1(50D)	0.003 ± (0.002)	**0.000 ± (0.000)**	**0.000 ± (0.000)**	0.520 ± (0.139)	**0.423 ± (0.112)**	0.428 ± (0.092)
f_2(50D)	63.800 ± (26.549)	**46.052 ± (82.658)**	73.061 ± (118.568)	63.776 ± (26.324)	54.673 ± (56.387)	**49.676 ± (50.601)**
f_3(50D)	373.016 ± (13.697)	91.289 ± (24.776)	**72.647 ± (19.567)**	371.909 ± (14.540)	87.262 ± (23.310)	**75.835 ± (22.884)**
f_4(50D)	**0.004 ± (0.002)**	0.006 ± (0.013)	0.005 ± (0.008)	**1.429 ± (0.120)**	1.461 ± (0.137)	1.450 ± (0.109)
f_5(2D)	−176.138 ± (0.000)	−176.138 ± (0.000)	−176.138 ± (0.000)	−176.108 ± (0.027)	−176.112 ± (0.025)	**-176.113 ± (0.021)**
f_6(2D)	0.000 ± (0.000)	0.000 ± (0.000)	0.000 ± (0.000)	0.029 ± (0.036)	0.028 ± (0.028)	**0.024 ± (0.019)**
f_7(50D)	0.013 ± (0.003)	0.006 ± (0.002)	**0.004 ± (0.001)**	1.779 ± (0.349)	1.003 ± (0.214)	**0.926 ± (0.273)**
f_8(2D)	**0.000 ± (0.000)**	**0.000 ± (0.000)**	0.003 ± (0.015)	0.482 ± (0.076)	0.222 ± (0.186)	**0.183 ± (0.155)**
f_9(50D)	0.000 ± (0.000)	0.000 ± (0.000)	0.000 ± (0.000)	0.216 ± (0.078)	0.203 ± (0.090)	**0.200 ± (0.089)**
	σ^2=0.5			σ^2=0.75		
Function	DE	DEPS	DEIS	DE	DEPS	DEIS
f_1(50D)	1.019 ± (0.218)	0.908 ± (0.186)	**0.860 ± (0.184)**	1.425 ± (0.419)	**1.135 ± (0.241)**	1.251 ± (0.257)
f_2(50D)	70.227 ± (33.627)	60.543 ± (58.286)	**48.710 ± (53.251)**	75.720 ± (45.043)	55.966 ± (55.840)	**36.721 ± (41.272)**
f_3(50D)	373.988 ± (16.081)	93.058 ± (24.168)	**69.817 ± (18.044)**	371.231 ± (13.100)	94.230 ± (20.757)	**71.339 ± (20.057)**
f_4(50D)	**1.889 ± (0.239)**	2.026 ± (0.439)	1.914 ± (0.247)	**2.288 ± (0.3215)**	2.407 ± (0.305)	2.340 ± (0.257)
f_5(2D)	−176.051 ± (0.075)	**-176.090 ± (0.045)**	−176.085 ± (0.051)	**-176.057 ± (0.084)**	−176.039 ± (0.105)	−176.040 ± (0.093)
f_6(2D)	0.064 ± (0.061)	**0.049 ± (0.055)**	0.067 ± (0.064)	0.070 ± (0.071)	**0.067 ± (0.060)**	0.082 ± (0.097)
f_7(50D)	21.182 ± (0.633)	4.499 ± (3.250)	**3.773 ± (0.423)**	21.618 ± (0.114)	7.500 ± (4.983)	**7.023 ± (4.735)**
f_8(2D)	0.500 ± (0.000)	**0.486 ± (0.075)**	0.491 ± (0.046)	0.500 ± (0.000)	0.500 ± (0.000)	0.500 ± (0.001)
f_9(50D)	**0.368 ± (0.136)**	0.444 ± (0.201)	0.396 ± (0.138)	0.684 ± (0.241)	**0.648 ± (0.283)**	0.661 ± (0.279)

TABLE V

MEAN ± (STANDARD DEVIATION) OF THE BEST FITNESS VALUE, FOR VARIABLE SHAKING RATE, 0.1, 0.3, 0.5, AND 0.75 IN SPECIFIED MEDIUM NOISE, σ^2=0.5.

		S_r=0.1		S_r=0.3	
Function	DE	DEPS	DEIS	DEPS	DEIS
f_1(50D)	0.962 ± (0.288)	0.871 ± (0.250)	0.924 ± (0.265)	**0.837 ± (0.193)**	0.857 ± (0.199)
f_2(50D)	95.070 ± (119.647)	49.308 ± (34.823)	**35.255 ± (32.292)**	50.712 ± (46.632)	51.851 ± (64.817)
f_3(50D)	375.125 ± (11.452)	157.666 ± (32.641)	139.801 ± (18.735)	95.820 ± (24.323)	73.341 ± (20.952)
f_4(50D)	1.912 ± (0.234)	1.746 ± (0.196)	**1.738 ± (1.821)**	1.917 ± (0.225)	1.953 ± (0.254)
f_5(2D)	−176.090 ± (0.046)	−176.076 ± (0.071)	−176.090 ± (0.045)	−176.090 ± (0.070)	−176.075 ± (0.056)
f_6(2D)	0.076 ± (0.062)	0.057 ± (0.075)	0.053 ± (0.059)	0.062 ± (0.097)	0.052 ± (0.046)
f_7(50D)	20.328 ± (3.470)	6.116 ± (6.149)	5.532 ± (5.466)	**3.884 ± (0.315)**	3.925 ± (0.491)
f_8(2D)	0.500 ± (0.001)	0.493 ± (0.030)	0.500 ± (0.000)	0.490 ± (0.054)	0.489 ± (0.061)
f_9(50D)	0.417 ± (0.141)	0.414 ± (0.205)	0.438 ± (0.178)	0.447 ± (0.178)	0.359 ± (0.159)
		S_r=0.5		S_r=0.7	
Function	DE	DEPS	DEIS	DEPS	DEIS
f_1(50D)	0.962 ± (0.288)	0.866 ± (0.176)	0.892 ± (0.226)	0.927 ± (0.229)	0.917 ± (0.192)
f_2(50D)	95.070 ± (119.647)	62.379 ± (62.607)	75.388 ± (78.108)	144.886 ± (187.138)	126.293 ± (187.465)
f_3(50D)	375.125 ± (11.452)	72.244 ± (23.459)	60.486 ± (17.461)	70.708 ± (17.053)	**59.449 ± (14.009)**
f_4(50D)	1.912 ± (0.234)	2.345 ± (0.496)	2.190 ± (0.578)	2.651 ± (0.859)	2.770 ± (0.966)
f_5(2D)	−176.090 ± (0.046)	−176.074 ± (0.072)	−176.065 ± (0.052)	**-176.105 ± (0.036)**	−176.059 ± (0.071)
f_6(2D)	0.076 ± (0.062)	0.066 ± (0.069)	0.062 ± (0.069)	**0.051 ± (0.051)**	0.063 ± (0.069)
f_7(50D)	20.328 ± (3.470)	4.232 ± (0.728)	4.069 ± (0.471)	4.590 ± (0.540)	4.464 ± (0.625)
f_8(2D)	0.500 ± (0.001)	0.457 ± (0.114)	**0.446 ± (0.139)**	0.464 ± (0.117)	0.473 ± (0.096)
f_9(50D)	0.417 ± (0.141)	0.442 ± (0.142)	0.424 ± (0.174)	**0.412 ± (0.149)**	0.419 ± (0.161)