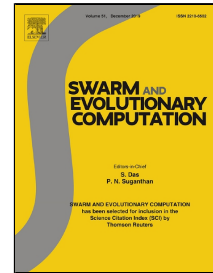


# Journal Pre-proof

Improved Differential Evolution for Noisy Optimization

Pratyusha Rakshit



PII: S2210-6502(18)30826-5  
DOI: <https://doi.org/10.1016/j.swevo.2019.100628>  
Reference: SWEVO 100628

To appear in: *Swarm and Evolutionary Computation*

Received Date: 26 September 2018  
Accepted Date: 24 November 2019

Please cite this article as: Pratyusha Rakshit, Improved Differential Evolution for Noisy Optimization, *Swarm and Evolutionary Computation* (2019), <https://doi.org/10.1016/j.swevo.2019.100628>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2019 Published by Elsevier.

# Improved Differential Evolution for Noisy Optimization

Pratyusha Rakshit

Electronics and Telecommunication Engineering Department, Jadavpur University, Kolkata- 700032, India

**Abstract**— A novel approach is proposed in this paper to improve the optimization proficiency of the differential evolution (DE) algorithm in the presence of stochastic noise in the objective surface by utilizing the composite benefit of four strategies. The first strategy is devised with an aim to employ reinforcement learning scheme of stochastic learning automata for autonomous selection of the sample size of a trial solution (for its repeated fitness evaluation) based on the characteristics of the fitness landscape in its local neighborhood. The second stratagem is proposed to estimate the effective fitness measure from multiple fitness samples of a trial solution, resulting from sampling. The novelty of the second policy lies in considering the distribution of noisy samples during effective fitness evaluation, instead of their direct averaging. The third strategy deals with amelioration of the DE/current-to-best/1 mutation scheme to judiciously direct the search in promising region, even in prevailing existence of noise in the objective surface. Finally, the greedy selection policy of the traditional DE is modified by introducing the principle of probabilistic crowding induced niching to ensure both the population quality and the population diversity. Comparative analysis performed on simulation results for diverse noisy benchmark functions reveal the statistically significant superiority of the proposed algorithm to its contenders with respect to function error value.

**Index terms**— *differential evolution; stochastic noise; learning automata; crowding based niching; fitness estimates.*

## I. INTRODUCTION

Real world problems usually attempt to determine estimator variables from a given set of measurements. Because of poor physical models of the sensors and/or sensor aging, measurement variables are often found to be contaminated with noise. The estimator variables being connected with measurement variables through a process model, the estimators too get contaminated with noise. In a single objective optimization problem, the objective function represents the system model and trial solutions are instantiations of the system variables. The output variables of the objective estimate represent the estimators. Determining the optimum in the objective surface in the presence of noisy measurements is an interesting problem, usually referred to as noisy single objective optimization problem (NSOOP) [2], [3], [25], [50], [56].

Traditional evolutionary algorithms (EAs) generate trial solutions by a population-based evolutionary procedure and employ a selection strategy to promote trial solutions having better objective estimate (often-called fitness measure) to the next generation population. The rest of the population is abandoned to maintain a uniform population size over generations of the algorithms. In an NSOOP, the fitness estimate being noisy, the selection operator may decline a quality trial solution from being promoted to the next generation population due to its illusive (noisy) fitness estimate, while allowing accommodation of a deceptive solution in the population with misleading good fitness [11], [24], [37], [42], [64].

Recently researchers are taking keen interest to design new EAs to address NSOOP. Traditional EAs developed for solving NSOOPs can be categorized primarily into three classes [24], including sampling (or explicit averaging), population sizing (or implicit averaging) and modified selection. Sampling strategy is concerned with the measurements of the fitness of a trial solution quite a number of times (referred to as ‘fitness samples’) [11], [23], [37], [42], [44], [64], [75] to check the possibility of infiltration of noise in the trial solution. The drawback of sampling lies in the context of increased computational complexity. The problem can be alleviated by considering a large population size [4], [6], [18], [19], [58]. It is based on the observation that the detrimental effect of noise on a population member can be compensated by the average behavior of similar members in a large population due to inherent

explorative capability of EAs. The deterministic selection strategy of traditional EAs has been modified in [7], [10], [14], [33], [59] to carefully identify the quality solutions over evolutionary generations. Among the other NSOOP techniques, uncertainty handling in covariance matrix adaptation-evolutionary strategy [20], local search [38], optimal computing budget allocation [30], [40], opposition based learning [47], logistic map-induced chaotic jump [35] and Kalman formulation of genetic algorithm [66] need special mentioning.

The present work addresses the issues of decisive selection of quality solutions over generation in a NSOOP by introducing four policies. Differential evolution (DE) [5], [13], [43], [45], [51], [65], [70], [71] is selected as the basic framework of our NSOOP because of its wide popularity as a real-parameter optimizer [15]. One or more of the proposed four strategies can be integrated with the DE to enhance its performance on complicated noisy fitness landscape.

First, it proposes an adaptive system to judiciously select sample size of a trial solution based on the noise characteristics in its local neighborhood. The reinforcement learning dynamics of stochastic learning automata (SLA) [27], [39] is utilized here to allocate a small (and a large) sample size to a solution located in a less (and more) noisy region in the objective space. It effectively balances the trade-off between computational accuracy and run-time complexity.

Second, the paper considers a weighted median approach of the fitness samples of a trial solution to estimate its effective fitness. The weight of a sample determines its contribution towards the effective fitness estimate. The weight assignment policy is devised here with an aim to reduce the influence of a noisy fitness sample measure (differing significantly from the aggregated measure of the remaining samples) towards estimation of effective fitness of the given solution.

Third, the paper proposes an amendment of DE/current-to-best/1 mutation scheme [13], [15], [45] to astutely select the noise-robust participants of the mutation policy. Furthermore, the scale factors are determined to surmount the misleading search direction of noise.

Finally, a probabilistic crowding based niching method is employed to allow “competition among similar (geographically near) members in an ecosystem” [28], [32], [36]. The strategy overcomes deterministic dismissal of quality solutions and simultaneously preserves population diversity by maintaining several “niches” (local optima) in the presence of noise.

The benefits of the four noise-handling strategies adopted in the paper over the existing noisy optimization algorithms are enlisted below.

The first stratagem, adaptive sampling using SLA, addresses the issue of dynamic sampling. The existing dynamic sampling policies [3], [11], [41], [44], [63], [67] vary the sample size based on hypothesis tests or with progress in evolutionary generation without capturing the jeopardizing effect of noise in the local neighborhoods of trial solutions. This problem has been circumvented in our previous works [48], [49], [52–55] by setting sample size proportional to the fitness variance in the local neighborhood of the trial solutions. However, only the fitness variance sensitive proportional selection of sample size does not work well at different sub-space of the fitness landscape, probably because of varying convexity of the landscape. In addition, the correctness of identifying the sample size of a trial solution is dependent on the appropriate selection of its mathematical relationship (linear or non-linear) with the fitness variance of the subpopulation around the given solution.

This paper gives an alternative approach to address the present problem by incorporating SLA [27], [39] to determine the sample size for periodic fitness evaluation of a given trial solution. The inherent learning dynamics of the SLA naturally selects the sample size by capturing non-uniform local distribution of noise using both the fitness estimate and the inter-quartile range (IQR) of fitness estimates of sub-population in the local neighborhood of a trial solution.

The second policy proposed here handles the intriguing issue of NSOOP to estimate the fitness of a trial solution based on the measure of its fitness samples [61], [62]. The average estimate of noisy fitness samples [23], [30], [40], [44], [64] being contingent upon their uniform probability of occurrence, also seems to be noisy. The alternative approaches proposed in [52], [54] divide the sample space (in objective space) into a number of predefined intervals. The density of samples in each interval describes the noise

induced uncertainty in their measurements. It is worth mentioning that the performances of the proposed approaches greatly rely on the proper selection of the number of intervals in the sample space.

To overcome the shortcomings, we here adopt a new policy for the measurement of effective fitness of a trial solution based its noisy fitness samples. The proposed method considers weighted median of the fitness samples in the neighborhood, where the weight of each fitness sample is determined by an aggregation of the fitness measure of other samples. An aggregation function has been devised to assign larger value to a weight  $\mu_l$  corresponding to the fitness sample  $J_l$ , if the median fitness of all samples, excluding the  $l$ -th sample  $J_l$ , differs slightly with the fitness measure of the  $l$ -th sample.

Third, the proposed algorithm introduces a modification in the traditional mutation strategy. The difference vector term in the traditional DE/current-to-best/1 mutation scheme [13], [15], [45] selects two members randomly from the entire population. However, in noisy optimization, random selection of population members from a relatively high noisy zone may result in poor donor vectors. This is avoided in the paper by adopting a strategy that involves two main steps. In the first step, we rank the members of the population based on their respective effective fitness estimate and weighted inter-quartile range (IQR) separately. Next, we evaluate the rank of a member by the product of their ranks in the above two sorting. The population members are then sorted in ascending order of their ranks. The two members, to be picked up for the mutation scheme, are then selected randomly from the candidates of the first half of the sorted list (relatively more robust than the rest of the population). The paper also recommends a novel strategy of adaptively tuning the mutation scale factors based on the fitness of population members.

The paper further addresses the issue of uncertainty management through introduction of ‘niching behavior’ [28], [32], [36] in the selection phase of the traditional single objective optimization algorithm. The principle of “niching” is introduced here to form and maintain multiple, diverse sub-populations with identical or varying fitness in the noisy objective surface. In the traditional single objective optimization, a trial solution participates in a competitive selection with its respective parent for being promoted to the next generation population. This parental selection strategy [14], [47] may fail to preserve representatives in multiple local optima in the noisy objective surface. The problem here is circumvented by following the principle of crowding which allows a trial solution to contend with its “geographically near member in the ecosystem [28]”. Crowding thus helps in preserving the population diversity by maintaining several “niches” (local optima) of the “ecosystem” (population of trial solutions) in the presence of noise in the objective surface.

Experiments have been undertaken to test the efficiency of the proposed approach for noisy optimization by contaminating the objective surface with noise samples taken from seven different types of noise distribution: i) zero mean Gaussian, ii) Poisson, iii) Rayleigh, iv) exponential, v) Gamma, vi) Cauchy and vii) random. The performance of the proposed noisy single objective optimization algorithm realized with DE (hereafter referred to as noisy differential evolution-NDE) is compared with ten state-of-art techniques [14], [21], [23], [35], [38], [40], [47], [62], [73], [75] to handle two well-known sets of benchmark functions [17], [29]. Experiments reveal that the proposed realization outperforms other algorithms by function error value.

The paper is divided into five sections. Section II overviews the DE algorithm and the basic principle of SLA. Section III provides the noise handling mechanisms in NDE. Experimental settings for the benchmarks and simulation strategies are explained in section IV. Conclusions are given in section V.

## II. PRELIMINARIES

### A. Differential Evolution Algorithm

This section overviews the main steps of differential evolution (DE) [5], [13], [15], [43], [45], [51], [65], [70], [71] a well-known meta-heuristic algorithm.

**1. Initialization:** DE randomly initializes a population  $\mathbf{\Omega}(t) = \{\vec{X}_1(t), \vec{X}_2(t), \dots, \vec{X}_S(t)\}$  of  $S$  real-valued target vectors, each of dimension  $D$ , denoted by  $\vec{X}_i(t) = \{x_{i,1}(t), \dots, x_{i,j}(t), \dots, x_{i,D}(t)\}$  for  $i = 1, 2, \dots, S$ , within the prescribed search space boundary  $[\vec{X}^l, \vec{X}^h]$ , where  $\vec{X}^l = \{x_1^l, x_2^l, \dots, x_D^l\}$  and  $\vec{X}^h = \{x_1^h, x_2^h, \dots, x_D^h\}$  at generation  $t = 0$ . This is realized by setting

$$x_{i,j}(0) = x_j^l + \text{rand} \times (x_j^h - x_j^l) \quad (1)$$

for  $j = 1, 2, \dots, D$  and  $i = 1, 2, \dots, S$ . Here  $\text{rand}$  represents a number selected from uniform distribution of random numbers in  $(0, 1)$ . The crossover ratio  $CR$  is initialized in  $[0, 1]$ . The objective function value  $J(\vec{X}_i(0))$  is evaluated for the target vector  $\vec{X}_i(0)$  with  $i = 1, 2, \dots, S$ .

**2. Mutation:** For each target vector  $\vec{X}_i(t)$ , a donor vector  $\vec{V}_i(t)$  is generated by randomly picking up two different members  $\vec{X}_{r_1}(t)$  and  $\vec{X}_{r_2}(t)$  from  $\mathbf{\Omega}(t)$  apart from  $\vec{X}_i(t)$  for  $i = 1, 2, \dots, S$ . The policy is given by

$$\vec{V}_i(t) = \vec{X}_i(t) + \eta_1(\vec{X}^{best}(t) - \vec{X}_i(t)) + \eta_2(\vec{X}_{r_1}(t) - \vec{X}_{r_2}(t)) \quad (2)$$

where  $\eta_1$  and  $\eta_2$  are scale factors in  $[0, 2]$  and  $i = 1, 2, \dots, S$ . Here  $\vec{X}^{best}(t)$  symbolizes the best candidate of current population. For a minimization problem,  $J(\vec{X}^{best}(t))$  is the minimum objective function value of the entire  $\mathbf{\Omega}(t)$ . The mutation operation described by (2) is called DE/current-to-best/1. Details of alternative mutation policies are available in [65].

**3. Crossover:** Among two different crossover strategies available in literature [65], binomial crossover is employed here based on its established performance irrespective of problem dimension [15]. The strategy is concerned with generating a trial vector  $\vec{Z}_i(t)$  for each pair of  $\vec{X}_i(t)$  and  $\vec{V}_i(t)$  for  $i = 1, 2, \dots, S$ . This is realized by

$$z_{i,j}(t) = \begin{cases} v_{i,j}(t), & \text{if } \text{rand} \leq CR \text{ or } j = j_r \\ x_{i,j}(t), & \text{otherwise} \end{cases} \quad (3)$$

for  $j = 1, 2, \dots, D$  and  $i = 1, 2, \dots, S$ . Here  $j_r$  represents an integer randomly selected from  $\{1, 2, \dots, D\}$ .

**4. Selection:** To ensure the quality of the population while keeping its size fixed at  $S$ , a greedy selection process is eventually employed to keep either the target vector  $\vec{X}_i(t)$  or its respective trial competitor  $\vec{Z}_i(t)$  in the next generation population based on their objective function values  $J(\vec{X}_i(t))$  and  $J(\vec{Z}_i(t))$  for  $i = 1, 2, \dots, S$ . For the minimization problem, the selection strategy is given by

$$\begin{aligned} \vec{X}_i(t+1) &= \vec{Z}_i(t), & \text{if } J(\vec{Z}_i(t)) \leq J(\vec{X}_i(t)) \\ &= \vec{X}_i(t), & \text{otherwise} \end{aligned} \quad (4)$$

**5. Convergence:** After each evolution step, steps 2 to 4 are repeated to achieve the terminating criterion. The condition includes restricting the number of generations or maintaining error limits or the both, whichever is achieved earlier.

## B. Stochastic Learning Automata

Stochastic learning automata (SLA) is a well-known member of the class of reinforcement learning [27], [39]. In SLA, an agent at a given state  $\tau(i) \in \{\tau(1), \tau(2), \dots, \tau(M)\}$  selects and executes an action  $\alpha(j)$  from a set of possible  $N$  actions  $\{\alpha(1), \alpha(2), \dots, \alpha(N)\}$  and eventually receives a reward/penalty from its environment. It helps the agent to learn a control policy to reach a definite goal by increasing the probability of its correct response. Let,  $p_{i,j}$  be the probability of selecting the action  $\alpha(j)$  at state  $\tau(i)$  for  $i = 1, 2, \dots, M$  and  $j = 1, 2, \dots, N$ .

To ensure unbiased selection of actions in the initial phase, equal selection probability of  $1/N$  is assigned to each action for each state. At a particular instant of SLA, the agent occupies a state  $\tau(i) \in \{\tau(1), \tau(2), \dots, \tau(M)\}$ , consults the action probabilities  $p_{i,l}$  for  $l = 1, 2, \dots, N$  to select and execute an action  $\alpha(j) \in \{\alpha(1), \alpha(2), \dots, \alpha(N)\}$  and consequently receives a response from the environment. Based on the observed response from the environment, the action probabilities  $p_{i,l}$  for  $l = 1, 2, \dots, N$  at state  $\tau(i)$  are modified keeping  $\sum_{l=1}^N p_{i,l} = 1$  with  $p_{i,l} \geq 0$ . The currently updated values of the action probabilities are used to guide the agent to judiciously select its actions at state  $\tau(i)$  in subsequent instants. Evidently, the reinforcement learning policy used to update the action probabilities plays the vital role to ensure the quality performance of SLA. The linear reinforcement learning scheme  $L_{R-P}$  [27], [39] is employed here to update the action probabilities as described below.

On receiving a reward from the environment at state  $\tau(i)$  by executing action  $\alpha(j)$

$$\begin{aligned} p_{i,j} &\leftarrow p_{i,j} + \beta_1 \times (1 - p_{i,j}) \quad \text{and} \\ p_{i,l} &\leftarrow (1 - \beta_1) p_{i,l}, \quad \text{for } l = 1, 2, \dots, N \text{ but } l \neq j. \end{aligned} \quad (5)$$

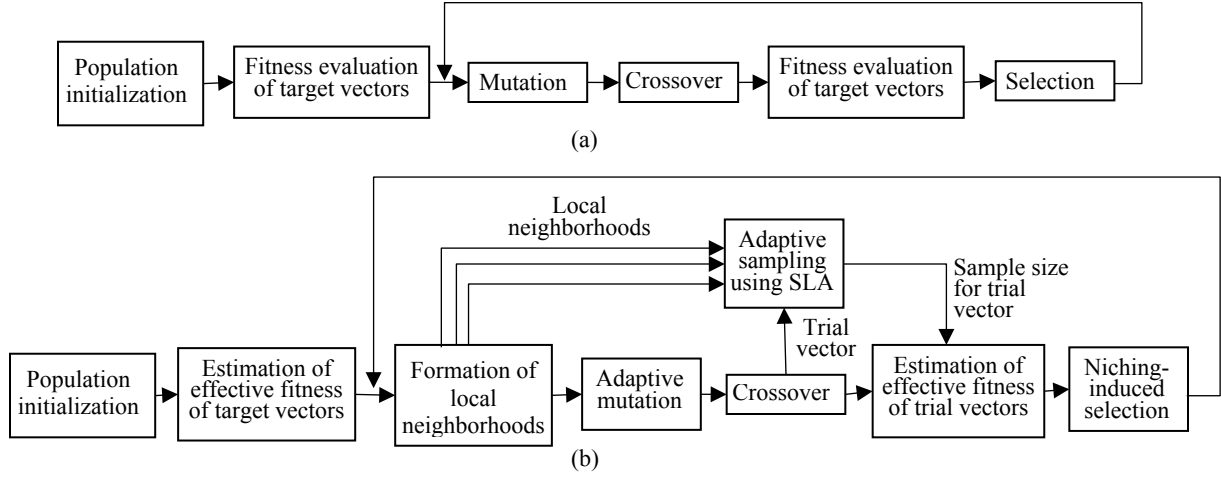
Contrarily, if a penalty response is obtained by the environment by execution of action  $\alpha(j)$  at state  $\tau(i)$ , then

$$\begin{aligned} p_{i,j} &\leftarrow (1 - \beta_2) \times p_{i,j} \quad \text{and} \\ p_{i,l} &\leftarrow \frac{\beta_2}{N-1} + (1 - \beta_2) \times p_{i,l}, \quad \text{for } l = 1, 2, \dots, N \text{ but } l \neq j. \end{aligned} \quad (6)$$

The parameters  $\beta_1 \in [0, 1]$  and  $\beta_2 \in [0, 1]$  are respectively associated with the reward and penalty response. In this work, we set  $\beta_1 = \beta_2$  following the  $L_{R-P}$  model [27], [39].

### III. OPTIMIZATION IN PRESENCE OF NOISE

This section extends the traditional DE algorithm by four strategies to improve its optimization capability when the objective function values of the population members are contaminated with stochastic noise. The extensional strategies include i) estimation of effective fitness (EEF), ii) adaptive mutation (AM), iii) adaptive sampling using SLA (ASSLA) and iv) niching-induced selection (NS). The schematic diagrams of the traditional DE and the proposed NDE (DE integrated with the proposed noise handling policies) algorithms are given in Fig. 1. The four extensions incorporated in NDE are elaborately discussed next.



**Fig. 1** Schematic architecture of (a) the traditional DE and (b) the proposed NDE

#### A. Estimation of Effective Fitness (EEF)

The primary concern of a NSOOP is to reduce the possible risk of promoting inferior solution to the next generation based on the fitness measures of contending solutions in the presence of noise in the fitness landscape. The certainty in assessing fitness measure of a population member  $\vec{X}_i$  is enhanced by sampling its objective function  $J(\vec{X}_i)$  for a number of times, called sample size  $n(\vec{X}_i)$ . The first strategy proposed here is concerned with estimation of effective fitness  $\bar{J}(\vec{X}_i)$  of  $\vec{X}_i$  from the measurement values of  $n(\vec{X}_i)$  samples of  $J(\vec{X}_i)$ .

The existing literature on NSOOP usually takes an average of  $n(\vec{X}_i)$  samples of  $J(\vec{X}_i)$  to estimate  $\bar{J}(\vec{X}_i)$  disregarding the degree of contamination of noise on individual samples. A large value of a severely noise-affected sample among  $n(\vec{X}_i)$  samples of  $J(\vec{X}_i)$  in turn makes the average estimate to be an erroneous one. This impasse has been overcome here by considering weighted average of  $n(\vec{X}_i)$  samples of  $J(\vec{X}_i)$  to estimate  $\bar{J}(\vec{X}_i)$ . The weights are assigned to individual samples based on their location in the fitness landscape with respect to others. The strategy assigns larger (and smaller) weight to a sample in the crowded (and sparse) region in the objective surface. The weight here signifies the degree of creeping of noise in the specific fitness sample.

Let the fitness samples of  $J(\vec{X}_i)$ , sorted in the ascending order, be symbolized by  $J_l(\vec{X}_i)$  for  $l = 1, 2, \dots, n(\vec{X}_i)$  with

$$J_1(\vec{X}_i) \leq J_2(\vec{X}_i) \leq \dots \leq J_{n(\vec{X}_i)}(\vec{X}_i). \quad (7)$$

Moreover, let  $J_l'(\vec{X}_i)$  be the median of  $J(\vec{X}_i)$  considering all its samples excluding  $J_l(\vec{X}_i)$ . In the proposed approach, the weight  $\mu_l$  of  $J_l(\vec{X}_i)$  is modeled as a decreasing function of the absolute difference between  $J_l(\vec{X}_i)$  and  $J_l'(\vec{X}_i)$ , given by

$$\mu_l = \exp\left(-\left|J_l(\vec{X}_i) - J_l'(\vec{X}_i)\right|\right) \quad (8)$$

for  $l = 1, 2, \dots, n(\vec{X}_i)$ . It is evident from (8) that less (and more) is the discrepancy between  $J_l(\vec{X}_i)$  and  $J'_l(\vec{X}_i)$ , larger (and smaller) the  $h_l$ , which indicates that  $J_l(\vec{X}_i)$  conforms to the remaining samples (captured by their median) to the large (and small) extent affirming a high (and a low) level of correctness in the measurement of  $J_l(\vec{X}_i)$  in the presence of noise. After evaluating the weights  $0 < \mu_l < 1$  for all fitness samples  $J_l(\vec{X}_i)$  for  $l = 1, 2, \dots, n(\vec{X}_i)$ , the weighted median of  $n(\vec{X}_i)$  samples is used as the effective fitness estimate  $\bar{J}(\vec{X}_i)$ . For  $n(\vec{X}_i)$  ordered sequence of fitness samples, given by (7), with respective positive weights  $\mu_l$  for  $l = 1, 2, \dots, n(\vec{X}_i)$ , the weighted median is the  $k$ -th sample (with  $k \in \{1, 2, \dots, n(\vec{X}_i)\}$ ) where

$$\sum_{l=1}^{k-1} \mu_l \leq \frac{1}{2} \sum_{l=1}^{n(\vec{X}_i)} \mu_l \quad \text{and} \quad \sum_{l=k+1}^{n(\vec{X}_i)} \mu_l \leq \frac{1}{2} \sum_{l=1}^{n(\vec{X}_i)} \mu_l. \quad (9)$$

The weighted median of  $n(\vec{X}_i)$  samples used effective fitness estimate in (9) thus obtained provides a robust measure of  $J(\vec{X}_i)$  against outlier.

The accuracy in the multiple measurements of  $J(\vec{X}_i)$  can be determined by its weighted inter-quartile range (IQR)  $R(\vec{X}_i)$  revealing the spread of sample values of  $J(\vec{X}_i)$ . Smaller the  $R(\vec{X}_i)$ , less is the contamination level of noise.  $R(\vec{X}_i)$  is defined as

$$R(\vec{X}_i) = Q_{0.75} - Q_{0.25} \quad (10.a)$$

where  $Q_{0.25}$  and  $Q_{0.75}$  respectively denote the first and third weighted quartiles of the  $n(\vec{X}_i)$  fitness samples, sorted in ascending order. To evaluate  $Q_p$  where  $p \in \{0.25, 0.75\}$ , we evaluate  $k$  such that

$$\frac{\chi_k}{\chi_{n(\vec{X}_i)}} \leq p \leq \frac{\chi_{k+1}}{\chi_{n(\vec{X}_i)}} \quad (10.b)$$

and then we set  $Q_p$  as

$$Q_p = J_k(\vec{X}_i) + (J_{k+1}(\vec{X}_i) - J_k(\vec{X}_i)) \frac{p\chi_{n(\vec{X}_i)} - \chi_k}{\chi_{k+1} - \chi_k} \quad (10.c)$$

where  $\chi_k = (k-1)\mu_k + (n(\vec{X}_i)-1) \sum_{l=1}^{k-1} \mu_l$  with  $\chi_1 = 0$  and  $\chi_{n(\vec{X}_i)} = (n(\vec{X}_i)-1) \sum_{l=1}^{n(\vec{X}_i)} \mu_l$ . (10.d)

### B. Adaptive Mutation (AM)

The mutation policy of DE is primarily responsible for controlling its population diversity [15], which eventually controls its exploitative and explorative capabilities. An accurate balance between exploration and exploitation governs the efficacy of DE in identifying the true global optimum in noisy multi-modal fitness landscape. It is evident from (2) that the second term  $\vec{X}^{best} - \vec{X}_i$  helps the target vector  $\vec{X}_i$  to perform



local exploitation around the population best position  $\vec{X}^{best}$  in the search space. The attraction of all target vectors towards  $\vec{X}^{best}$  ensures faster convergence. Again, the perturbation of  $\vec{X}_i$  by the differential mutation part  $\vec{X}_{r_1} - \vec{X}_{r_2}$  of (2) helps  $\vec{X}_i$  to globally explore the search space. The scale factors  $\eta_1$  and  $\eta_2$  respectively control the exploitative and explorative proficiency, the two cornerstones of DE.

However, in the present context of noisy optimization, it is expected that the best performance of DE cannot be achieved by equal setting of scale factors for all population members. For example, the solution with the minimum objective function value  $\bar{J}(\vec{X}^{best})$  is declared as the population best  $\vec{X}^{best}$ , which attracts the target vectors for their exploitation. It is noteworthy that  $\bar{J}(\vec{X}^{best})$  is calculated using the noisy samples of  $J(\vec{X}^{best})$  following (9). Intuitively, the scale factor  $\eta_1$  controlling the degree of attraction of a target vector  $\vec{X}_i$  towards  $\vec{X}^{best}$  should be modulated with the level of noise-contamination of  $J(\vec{X}^{best})$ . As indicated by (10), the weighted IQR  $R(\vec{X}^{best})$  quantitatively provides an estimate of the degree of the jeopardizing effect of noise on  $J(\vec{X}^{best})$ . More the  $R(\vec{X}^{best})$ , more is  $J(\vec{X}^{best})$  affected by the noise and hence less is the desired degree of orientation of  $\vec{X}_i$  towards  $\vec{X}^{best}$ . This is realized by setting  $\eta_1$  as

$$\eta_1 = \eta_1^l + (1 - \eta_1^l) \times \exp(-R(\vec{X}^{best})) \quad (11)$$

where  $\eta_1^l$  denotes the minimum permissible value of  $\eta_1$ .

Similarly, the remaining scale factor  $\eta_2$  controlling the degree of perturbation of  $\vec{X}_i$  by  $\vec{X}_{r_1} - \vec{X}_{r_2}$  is also set as a decreasing function of the maximum of  $IQR(\vec{X}_{r_1})$  and  $IQR(\vec{X}_{r_2})$ , as given by (12).

$$\eta_2 = \eta_2^l + (1 - \eta_2^l) \times \exp(-\max(R(\vec{X}_{r_1}), R(\vec{X}_{r_2}))) \quad (12)$$

It is also observed that complete random selection of  $\vec{X}_{r_1}$  and  $\vec{X}_{r_2}$  from the population may mislead the direction of search by  $\vec{X}_i$  in deceptive zones if  $\vec{X}_{r_1}$  and/or  $\vec{X}_{r_2}$  is largely affected by noise. The paper proposes a novel approach to judicious selection of  $\vec{X}_{r_1}$  and  $\vec{X}_{r_2}$  from the population by avoiding misleading effect of noise.

To achieve this, for each population member, we compute the following composite measure

$$\phi(\vec{X}_i) = \frac{\bar{J}(\vec{X}_i)}{\sum_{j=1}^S \bar{J}(\vec{X}_j)} \times \frac{R(\vec{X}_i)}{\sum_{j=1}^S R(\vec{X}_j)} \quad (13)$$

for  $i = 1, 2, \dots, S$ . Then, the population members are sorted in ascending order of their respective composite measures. For each target vector  $\vec{X}_i$ ,  $\vec{X}_{r_1}$  and  $\vec{X}_{r_2}$  are then randomly selected from the top  $S/2$  target vectors of the sorted population to generate a donor vector  $\vec{V}_i$  from  $\vec{X}_i$  following the DE/current-to-best/1 mutation provided  $r_1 \neq r_2 \neq i$ . Expression (13) reveals that the composite measure  $\phi(\vec{X}_i)$  is influenced by both fitness estimate  $\bar{J}(\vec{X}_i)$  and weighted IQR  $R(\vec{X}_i)$ . A solution possessing small measure for both  $\bar{J}(\cdot)$  and  $R(\cdot)$  appears in the forefront of the population sorted in the ascending order of  $\phi(\cdot)$ , representing its robustness against noise. Hence the controlled random selection of  $\vec{X}_{r_1}$  and  $\vec{X}_{r_2}$  from the

noise-resilient set of top  $S/2$  sorted population members ensures the exploration of  $\bar{X}_i$  in the promising regions of the noisy fitness landscape.

### C. Adaptive Sampling Using Stochastic Learning Automata (ASSLA)

After generating a trial vector  $\bar{Z}_i$  from its respective target vector  $\bar{X}_i$  through the proposed AM strategy and crossover policy, we need to identify the sample size  $n(\bar{Z}_i)$  of  $\bar{Z}_i$  for periodic evaluation of its objective function  $J(\bar{Z}_i)$ . More the sample size  $n(\bar{Z}_i)$ , better is the accuracy in the assessment of  $\bar{J}(\bar{Z}_i)$ , given by (9). But a large sample size increases computational complexity. This in turn calls for an adaptive strategy for sample size selection. In the present work,  $n(\bar{Z}_i)$  is adapted based on the fitness profile in the local neighborhood (LN) of  $\bar{Z}_i$ . The LN of the trial vector  $\bar{Z}_i$  is formed by the target vectors of the current population which are in the close proximity of  $\bar{Z}_i$  in the parameter space.

The weighted IQR of the fitness estimates of the sub-population surrounding  $\bar{Z}_i$  thus formed represents the intensity of noise contamination in the LN of  $\bar{Z}_i$ . Evidently, a large IQR in the LN of  $\bar{Z}_i$  indicates the need of a large sample size  $n(\bar{Z}_i)$ . Moreover, for a minimization problem, if  $\bar{Z}_i$  falls in an LN with small fitness estimate, a large  $n(\bar{Z}_i)$  is required to guarantee the true quality of  $\bar{Z}_i$ . Otherwise, a small value of  $n(\bar{Z}_i)$  is used to reduce the run-time. This adaptive selection of sample size is realized here using SLA. The following steps elaborately discuss the implementation of the proposed sample size selection strategy.

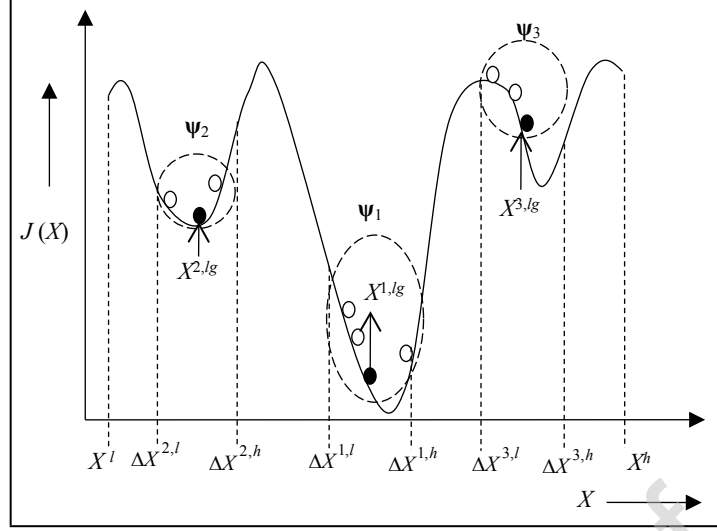
## 1. Local Neighborhood Formation

The LNs are identified from the population already sorted in ascending order of the composite measure  $\varphi(\cdot)$  given by (13). The target vector at the top most position of the sorted population  $\Omega$  is selected as the first local guide  $\bar{X}^{1,lg}$  of the first LN  $\psi_1$ . The members of LN  $\psi_1$  in the locality of  $\bar{X}^{1,lg}$  is identified from the population  $\Omega$  lying within the hyperspace bounded by  $[\Delta \bar{X}^{1,l}, \Delta \bar{X}^{1,h}]$ , where

$$\Delta \bar{X}^{1,l} = \{x_1^{1,lg} - \Delta x_1, x_2^{1,lg} - \Delta x_2, \dots, x_D^{1,lg} - \Delta x_D\} \text{ and } \Delta \bar{X}^{1,h} = \{x_1^{1,lg} + \Delta x_1, x_2^{1,lg} + \Delta x_2, \dots, x_D^{1,lg} + \Delta x_D\} \quad (14.1)$$

$$\text{and } \Delta x_j = (x_j^h - x_j^l) / S \quad \text{for } j = 1, 2, \dots, D. \quad (14.2)$$

Next, the solution at the top most position of the sorted list  $\{\Omega - \psi_1\}$  is recorded as the second local guide  $\bar{X}^{2,lg}$  with its subordinates within  $[\Delta \bar{X}^{2,l}, \Delta \bar{X}^{2,h}]$  (following (14)) forming the second LN  $\psi_2$ . Continuation of this procedure assigns each member of  $\Omega$  to an LN and therefore segments the  $D$ -dimensional search space into a number of LNs. Let the number of LNs thus identified in the current generation be denoted by  $C$ . The LNs formed in a one-dimensional search space with their respective local guides are pictorially presented in Fig. 2.



**Fig. 2** Identification of local neighborhoods in one-dimensional search space with their respective local guides

## 2. Fitness Estimate and Weighted IQR in Local Neighborhood

Composite measures of effective fitness estimates and weighted IQR of the constituent members are used to characterize the corresponding LN. For LN  $\psi_j$ , its fitness estimate  $\rho_j$  and the weighted IQR  $\delta_j$  are computed following the principles of evaluating weighted median and weighted IQR as discussed in section III.A for  $j = 1, 2, \dots, C$ . The procedure for determining fitness estimate  $\rho_j$  and the fitness variance  $\delta_j$  of the LN  $\psi_j$  based on its constituent members is summarized below.

Let  $\bar{X}_{j,l}$  be the  $l$ -th member of the  $j$ -th LN  $\psi_j$  with its effective fitness estimate  $\bar{J}(\bar{X}_{j,l})$  for  $l = 1, 2, \dots, |\psi_j|$ . Here  $|\psi_j|$  denotes the size of  $\psi_j$ . First the effective fitness estimates  $\bar{J}(\bar{X}_{j,l})$  for  $l = 1, 2, \dots, |\psi_j|$  of the members of the LN  $\psi_j$  are sorted in the ascending order with

$$\bar{J}(\bar{X}_{j,1}) \leq \bar{J}(\bar{X}_{j,2}) \leq \dots \leq \bar{J}(\bar{X}_{j,|\psi_j|}). \quad (15)$$

Moreover, let  $\bar{J}'(\bar{X}_{j,l})$  be the median of the sorted list of effective fitness estimates of all members of  $\psi_j$  excluding  $\bar{J}(\bar{X}_{j,l})$ . The contribution of  $\bar{J}(\bar{X}_{j,l})$  towards evaluation of fitness estimate  $\rho_j$  of  $\psi_j$  is captured by its weight  $w_l$ , given by

$$w_l = \exp\left(-\left|\bar{J}(\bar{X}_{j,l}) - \bar{J}'(\bar{X}_{j,l})\right|\right). \quad (16)$$

for  $l = 1, 2, \dots, |\psi_j|$ . After evaluating the weights  $0 < w_l < 1$  for  $l = 1, 2, \dots, |\psi_j|$ , the weighted median of the sorted list, described by (15), is used as the fitness estimate  $\rho_j$  of LN  $\psi_j$ . For  $|\psi_j|$  ordered sequence of effective fitness estimates of members  $\bar{X}_{j,l}$  of  $\psi_j$  with respective positive weights  $w_l$  for  $l = 1, 2, \dots, |\psi_j|$ , the weighted median is the  $k$ -th sample (with  $k \in \{1, 2, \dots, |\psi_j|\}$ ) where

$$\sum_{l=1}^{k-1} w_l \leq \frac{1}{2} \sum_{l=1}^{|\psi_j|} w_l \quad \text{and} \quad \sum_{l=k+1}^{|\psi_j|} w_l \leq \frac{1}{2} \sum_{l=1}^{|\psi_j|} w_l. \quad (17)$$

The weighted IQR  $\delta_j$  of  $\Psi_j$  is defined as

$$\delta_j = Q_{j,0.75} - Q_{j,0.25} \quad (18.a)$$

where  $Q_{j,0.25}$  and  $Q_{j,0.75}$  respectively denote the first and third weighted quartiles of the sorted list of effective fitness estimates of the members of  $|\Psi_j|$ . To evaluate  $Q_{j,p}$  where  $p \in \{0.25, 0.75\}$ , we evaluate  $k$  such that

$$\frac{\chi_k}{\chi_{|\Psi_j|}} \leq p \leq \frac{\chi_{k+1}}{\chi_{|\Psi_j|}} \quad (18.b)$$

and then we set  $Q_{j,p}$  as

$$Q_{j,p} = \bar{J}(\bar{X}_{j,k}) + (\bar{J}(\bar{X}_{j,k+1}) - \bar{J}(\bar{X}_{j,k})) \frac{p\chi_{|\Psi_j|} - \chi_k}{\chi_{|\Psi_j|} - \chi_k} \quad (18.c)$$

where  $\chi_k = (k-1)w_k + (|\Psi_j|-1) \sum_{l=1}^{k-1} w_l$  with  $\chi_1 = 0$  and  $\chi_{|\Psi_j|} = (|\Psi_j|-1) \sum_{l=1}^{|\Psi_j|} w_l$ . (18.d)

Once we obtain  $\rho_j$  and  $\delta_j$  for all neighborhoods  $j = 1, 2, \dots, C$ , the normalized fitness estimate  $\hat{\rho}_j \in [0,1]$  and the normalized IQR  $\hat{\delta}_j \in [0,1]$  of  $\Psi_j$  are computed using (19) for  $j = 1, 2, \dots, C$ .

$$\hat{\rho}_j = \rho_j \Big/ \sum_{l=1}^C \rho_l \quad \text{and} \quad \hat{\delta}_j = \delta_j \Big/ \sum_{l=1}^C \delta_l \quad (19)$$

Evidently,  $\hat{\rho}_j$  and  $\hat{\delta}_j$  for  $j = 1, 2, \dots, C$  belong to  $(0, 1)$ .

### 3. Design of State-Action Table

A three-dimensional state-action table is designed as shown in Fig. 3 to help ASSLA to select a sample size for a trial vector from the pool  $\{2, 3, \dots, N\}$  where  $N$  denotes the maximum sample size. The sample size thus selected is used for estimation of effective fitness and weighted IQR of the given trial vector. A trial vector occupies a state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  with  $\tau_1(m) \in \{\tau_1(1), \tau_1(2), \dots, \tau_1(M)\}$  and  $\tau_2(l) \in \{\tau_2(1), \tau_2(2), \dots, \tau_2(L)\}$  based on the normalized fitness estimate and IQR of its LN respectively. Evidently, here  $\{\tau_1(1), \tau_1(2), \dots, \tau_1(M)\}$  and  $\{\tau_2(1), \tau_2(2), \dots, \tau_2(L)\}$  of the state-action table respectively symbolize set of possible  $M$  and  $L$  states of the population  $\Omega$  based on the normalized fitness estimate  $\hat{\rho}$  and the normalized IQR  $\hat{\delta}$  of their LNs. The states  $\tau_1(m)$  and  $\tau_2(l)$  correspond to the population members with normalized LN fitness estimate  $\hat{\rho}$  and IQR  $\hat{\delta}$  in  $[(m-1)/M, m/M)$  and  $[(l-1)/L, l/L)$  for  $m = 1, 2, \dots, M$  and  $l = 1, 2, \dots, L$  respectively. The indices of the third dimension of the table correspond to uniformly quantized values of the sample size  $\{2, 3, \dots, N\}$ , to be used for the periodic fitness evaluation of a trial vector. The component  $p_{m,l,\alpha}$  of the state-action table denotes the probability of selecting a sample size  $\alpha \in \{2, 3, \dots, N\}$  at a specific state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  for  $m \in \{1, 2, \dots, M\}$  and  $l \in \{1, 2, \dots, L\}$ .

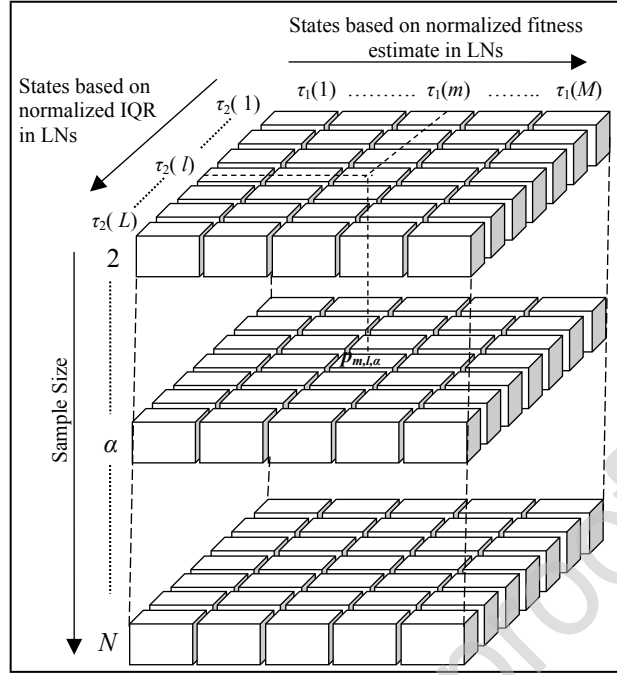


Fig. 3 State-action table for SLAAS

#### 4. Identification of Local Neighborhood of a Trial Vector

Next, a trial vector  $\bar{Z}_i$  is generated from its respective parent  $\bar{X}_i$  using the proposed AM strategy and the traditional crossover phase. Then we find out the local guide  $\bar{X}^{k,lg}$  (for  $k \in \{1, 2, \dots, C\}$ ) which is the nearest local guides to  $\bar{Z}_i$  among  $C$  current local guides. Therefore,  $\bar{Z}_i$  is accommodated in the LN  $\Psi_k$  formed around  $\bar{X}^{k,lg}$ . Hence  $\hat{\rho}_k$  and  $\hat{\delta}_k$  are respectively used as the normalized fitness estimate and IQR of  $\bar{Z}_i$  for assigning its state-pair in the state-action table.

#### 5. State-pair Assignment of a Trial Vector

$\bar{Z}_i$  is assigned to the state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  if  $\hat{\rho}_k \in [(m-1)/M, m/M)$  and  $\hat{\delta}_k \in [(l-1)/L, l/L)$  for  $m \in \{1, 2, \dots, M\}$  and  $l \in \{1, 2, \dots, L\}$ .

#### 6. Adaptive Selection of Sample Size using Stochastic Learning Automata

SLA helps the trial vectors to judiciously select their sample sizes based on the action probabilities  $p_{m,l,\alpha}$  which are adapted based on the reward and penalty received during selection of sample size  $\alpha$  at state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  in the previous generations for  $m = 1, 2, \dots, M$ ,  $l = 1, 2, \dots, L$  and  $\alpha = 1, 2, \dots, N$ . For example, at a specific state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$ ,  $p_{m,l,\alpha} \geq p_{m,l,\tilde{\alpha}}$  for  $\tilde{\alpha} = 2, 3, \dots, N$  indicates that the selection of particular sample size  $\alpha$  was rewarded many times before in the evolution process. Naturally, the learning experience directs a trial vector  $\bar{Z}_i$  at state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  to select the sample size  $n(\bar{Z}_i) = \alpha$  with the highest selection probability.

However, selection of sample size only based on the highest action probability may result in premature convergence. To overcome the impasse, the Roulette-choice strategy [51] is employed for selection of

sample size. The strategy selects a sample size  $\alpha$  from the pool  $\{2, 3, \dots, N\}$  for a target vector  $\vec{Z}_i$  at state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  if the following condition holds.

$$\sum_{\tilde{\alpha}=1}^{\alpha-1} p_{m,l,\tilde{\alpha}} \leq r \leq \sum_{\tilde{\alpha}=1}^{\alpha} p_{m,l,\tilde{\alpha}} \quad (20)$$

## 7. Reward/Penalty-based Update of Action Probabilities

The sample size  $n(\vec{Z}_i) = \alpha$  selected for  $\vec{Z}_i$  in the last step is used for its periodic fitness evaluation, effective fitness estimate  $\bar{J}(\vec{Z}_i)$  and weighted IQR  $R(\vec{Z}_i)$ , using (9) and (10) respectively. The weighted IQR  $R(\vec{Z}_i)$  captures the spread of the  $n(\vec{Z}_i)$  fitness samples of  $J(\vec{Z}_i)$  in the noisy objective surface. Intuitively, the sample size resulting in small (and large) weighted IQR of a solution is rewarded (and penalized). The reinforcement learning scheme of SLA is used to update the action probabilities based on this reward/penalty. To determine whether the selection  $n(\vec{Z}_i) = \alpha$  by  $\vec{Z}_i$  at state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  is to be rewarded or penalized, we first identify the nearest neighbor  $\vec{X}_j$  (from the pool of target vectors) of the given target vector  $\vec{Z}_i$  in its LN  $\Psi_k$ . Symbolically,

$$\|\vec{Z}_i - \vec{X}_j\| = \arg \left( \min_{\vec{X}_l \in \Psi_k} \|\vec{Z}_i - \vec{X}_l\| \right) \quad (21)$$

where  $\|\cdot\|$  denotes the Euclidean norm. It is worth mentioning that the nearest neighbors  $\vec{Z}_i$  and  $\vec{X}_j$  in the same LN  $\Psi_k$  have the identical normalized LN fitness estimate  $\hat{\rho}_k$  and IQR  $\hat{\delta}_k$ . Hence, they occupy the identical state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$ . However, the sample size  $n(\vec{X}_j)$  selected by the target vector  $\vec{X}_j$  for its periodic fitness evaluations in the last generation may be different from the sample size  $n(\vec{Z}_i) = \alpha$  selected by the current trial vector  $\vec{Z}_i$ , resulting in unequal weighted IQRs  $R(\vec{Z}_i) \neq R(\vec{X}_j)$ , even at the same state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$ . Smaller the weighted IQR, more robust is a solution against noise. The sample size assigned to the robust contender among  $\vec{Z}_i$  and  $\vec{X}_j$  is thus expected to be more appropriate for solutions falling in the LN  $\Psi_k$  in the future generations.

In other words, if the selection of  $n(\vec{Z}_i) = \alpha$  (by the Roulette-choice strategy of SLA) enhances the precision in measuring the fitness samples of  $\vec{Z}_i$  in the presence of noise as compared to its nearest neighbor  $\vec{X}_j$ , captured by  $R(\vec{Z}_i) \leq R(\vec{X}_j)$ , a positive reward is given to the selection of  $n(\vec{Z}_i) = \alpha$  by  $\vec{Z}_i$  at state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$ . The corresponding action probability  $p_{m,l,\alpha}$  is then increased following the SLA policy given in (5). If the selected action results in no improvement in fitness precision, a penalty is given to the selected action  $n(\vec{Z}_i) = \alpha$  at state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  by a reduction in  $p_{m,l,\alpha}$  following (6). Mathematically,

if  $R(\vec{Z}_i) \leq R(\vec{X}_j)$ , then

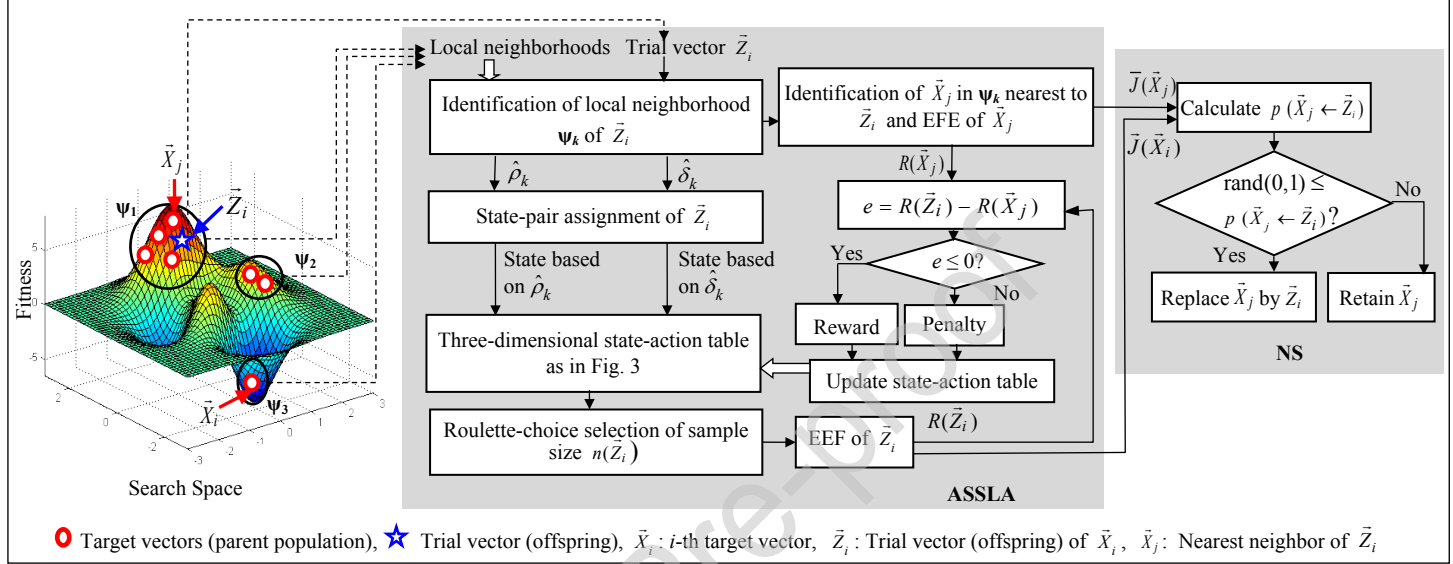
$$\begin{aligned} p_{m,l,\alpha} &\leftarrow p_{m,l,\alpha} + \beta_1 \times (1 - p_{m,l,\alpha}) \\ p_{m,l,\tilde{\alpha}} &\leftarrow (1 - \beta_1) p_{m,l,\tilde{\alpha}}, \quad \text{for } \tilde{\alpha} = \{2, 3, \dots, N\}, \quad \tilde{\alpha} \neq \alpha, \end{aligned} \quad (22)$$

otherwise,

$$p_{m,l,\alpha} \leftarrow (1-\beta_2) \times p_{m,l,\alpha}$$

$$p_{m,l,\tilde{\alpha}} \leftarrow \frac{\beta_2}{N-1} + (1-\beta_2) \times p_{m,l,\tilde{\alpha}}, \quad \text{for } \tilde{\alpha} = \{2,3,\dots,N\}, \quad \tilde{\alpha} \neq \alpha. \quad (23)$$

Steps (4) to (7) are repeated for all trial vectors. The adaptation of sample size using the ASSLA strategy is represented in Fig. 4.



**Fig. 4** ASSLA strategy for selection of  $n(\bar{Z}_i)$  based on the normalized fitness estimate  $\hat{\rho}_k$  and weighted IQR  $\hat{\delta}_k$  in its local neighborhood  $\psi_k$

#### D. Niching induced Selection (NS)

The deterministic greedy selection process of the traditional DE, as described by (4), keeps either the generated trial vector  $\bar{Z}_i$  or the respective target vector  $\bar{X}_i$  in the population of next generation based on their objective function value only, however, ignoring the location of the vectors in the search space. The promotion of either  $\bar{Z}_i$  or  $\bar{X}_i$  only based on their fitness measures in the presence of noise, may reduce the population diversity if the discarded solution is located in the sparse zone of the search space. It in turn may miss to explore promising regions (with few population members) in the search space. To overcome this problem faced by deterministic selection strategy, niching method is employed to simultaneously preserve the population quality and diversity. The probabilistic crowding policy of niching mechanism is utilized here to allow “competition among similar (geographically near) members in an ecosystem” [28], [32], [36].

There are two primarily two phases in the conventional probabilistic crowding based niching including pairing and replacement. In the pairing phase, for each trial vector  $\bar{Z}_i$ ,  $1 < w \leq S$  members are randomly selected from the population. Among these randomly selected  $w$  individuals, the candidate most similar to  $\bar{Z}_i$  (in the parameter space) participates in the competition with  $\bar{Z}_i$  for promotion to next generation [28], [32], [36]. The aim of pairing phase is to allow similar members of the population to participate in the local competition. The parameter  $w$  is called the window size. However, the efficacy of local exploitation by this pairing policy is determined by an appropriate selection of the window size  $w$ . It in turn requires *a priori* knowledge about the location and spacing of the local optima. An accurate replacement can be ensured by

choosing  $w = S$  (population size), but at the cost of additional computational overhead of distance calculation of each population member from  $\bar{Z}_i$ .

The pairing policy proposed in the present work is devised with an aim to attain pairing efficiency equivalent to  $w = S$ , but by avoiding distance measurements with all members of the population. The proposed policy allows the trial vector  $\bar{Z}_i$  with its nearest neighbor  $\bar{X}_j$  in its LN  $\Psi_k$  (which is already identified by the principle described in section III.C.4). To realize this, first we identify the local guide  $\bar{X}^{k,lg}$ , for  $k \in \{1, 2, \dots, C\}$ , which is nearest to  $\bar{Z}_i$  among all possible  $C$  local guides discovered in step-1 under section III.C. Thus  $\bar{Z}_i$  is placed in the LN  $\Psi_k$  around  $\bar{X}^{k,lg}$  for  $k \in \{1, 2, \dots, C\}$ . Next, we select the nearest neighbor  $\bar{X}_j$  of  $\bar{Z}_i$  in the LN  $\Psi_k$  for the subsequent competitive selection. Evidently, the solution  $\bar{X}_j$  providing minimum distance from the trial vector  $\bar{Z}_i$  is not necessarily identical with  $\bar{Z}_i$ 's immediate parent  $\bar{X}_i$ . Hence even after discarding one solution after competition among  $\bar{Z}_i$  and  $\bar{X}_j$ , the remaining one is still kept as a member of the LN  $\Psi_k$ , thus preserving the population diversity. Moreover, the proposed pairing policy computes distance for  $C+|\Psi_k|$  times ( $C$  times for LN identification and  $|\Psi_k|$  times for identifying the nearest neighbor  $\bar{X}_j$  of  $\bar{Z}_i$  in the LN  $\Psi_k$ ) to discover  $\bar{X}_j$ , the nearest neighbor of  $\bar{Z}_i$ .

In the replacement phase of crowding policy, a probabilistic estimate is used to decide the substitution of the nearest neighbor  $\bar{X}_j$  by  $\bar{Z}_i$  for the next generation. Smaller the  $\bar{J}(\bar{Z}_i)$  (and  $R(\bar{Z}_i)$ ) with respect to  $\bar{J}(\bar{X}_j)$  (and  $R(\bar{X}_j)$ ), more is the probability of  $\bar{Z}_i$  to replace  $\bar{X}_j$ . This is captured by the following replacement probability.

$$p(\bar{X}_j \leftarrow \bar{Z}_i) = \frac{1}{1 + \exp(-(\bar{J}(\bar{X}_j) \times R(\bar{X}_j) - \bar{J}(\bar{Z}_i) \times R(\bar{Z}_i)))} \quad (24)$$

It is evident from (24) that  $p(\bar{X}_j \leftarrow \bar{Z}_i) \rightarrow 1$  if  $\bar{J}(\bar{X}_j) \gg \bar{J}(\bar{Z}_i)$  and  $R(\bar{X}_j) \gg R(\bar{Z}_i)$ . The strategy is pictorially represented in Fig. 4.

The pseudo-code for the proposed NDE algorithm is given below.

*Pseudo code of the Proposed NDE*

---



---

**Procedure DEN**

**Input:** Noisy optimization problem, stopping criterion, maximum ( $N$ ) sample size, number of states  $M$  and  $L$  of the state-action table, maximum value of scale factor  $\eta_1^h$  and  $\eta_2^h$ , stochastic learning parameters  $\beta_1$  and  $\beta_2$ .

**Output:** Optimal solution  $\bar{X}^{best}$ .

---

**Begin**

**1. Initialization:**

- 1.(i). Randomly initialize a population of  $S$ ,  $D$ -dimensional individuals  $\Omega(t) = \{\bar{X}_1(t), \bar{X}_2(t), \dots, \bar{X}_S(t)\}$  following (1) at  $t=0$ .
- 1.(ii). Set the sample size  $n(\bar{X}_i(t))$  of the objective function  $\bar{J}(\bar{X}_i(t))$  to the minimum sample size 2 for  $i = 1, 2, \dots, S$ .
- 1.(iii). Initialize  $[p_{m,l,\alpha}] = 0.1$  where  $m = 1, 2, \dots, M$  and  $l = 1, 2, \dots, L$  and  $\alpha = \{2, 3, \dots, N\}$ .
- 1.(iv). **Effective Estimation of Fitness:** Evaluate  $\bar{J}(\bar{X}_i(t))$  and  $R(\bar{X}_i(t))$  using  $n(\bar{X}_i(t))$  fitness samples for  $i = 1, 2, \dots, S$  using (9) and (10) respectively.

- 1.(v). Set  $\bar{X}^{best}(t) = \arg \left( \min_{\forall \bar{X}_i(t)} (\bar{J}(\bar{X}_i(t))) \right)$  for minimization problem.

**2. While stopping criterion is not reached do begin**

- 2.(i) **Local Neighborhood Formation:** Identify  $C(t)$  local neighborhoods in the  $D$ -dimensional parameter space using the principle stated in section III.C.1. Compute  $\hat{\rho}_j$  and  $\hat{\delta}_j$  using (19) for  $j = 1, 2, \dots, C(t)$ .

- 2.(ii) **For  $i=1$  to  $S$  do begin**



a) **Adaptive Mutation:** Generate  $\vec{V}_i(t)$  corresponding to  $\vec{X}_i(t)$  using (2) and following the AM strategy proposed in section III.B.

b) **Binomial Crossover:** Generate  $\vec{Z}_i(t)$  for the pair  $\vec{X}_i(t)$  and  $\vec{V}_i(t)$  using (3).

c) **Local Neighborhood Identification:** Discover the local neighborhood  $\Psi_k$  of  $\vec{Z}_i(t)$  following the principle of section III.C.4 for  $k \in \{1, 2, \dots, C(t)\}$ .

d) **Nearest Neighbor Identification:** Identify  $\vec{X}_j(t) \in \Psi_k$  satisfying (21).

e) **State Assignment in State-Action Table:** Assign  $\vec{Z}_i(t)$  to the state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  if  $\hat{\rho}_k \in [(m-1)/M, m/M]$  and  $\hat{\delta}_k \in [(l-1)/L, l/L]$  for  $m \in \{1, 2, \dots, M\}$  and  $l \in \{1, 2, \dots, L\}$  respectively.

f) **Roulette-Wheel Selection:** Select  $n(\vec{Z}_i(t)) = \alpha$  from  $\{2, 3, \dots, N\}$  following the principle given in section III.C.6.

g) **Effective Estimation of Fitness:** Evaluate  $\bar{J}(\vec{Z}_i(t))$  and  $R(\vec{Z}_i(t))$  using (9) and (10) respectively.

h) **Reward/Penalty-based Update of Action Probabilities** (of  $\vec{Z}_i(t)$  at state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  and action  $n(\vec{Z}_i(t)) = \alpha$ ):

**If**  $R(\vec{Z}_i(t)) \leq R(\vec{X}_j(t))$  **then do**

Update the action probabilities at state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  by (22).

**Else do**

Update the action probabilities at state-pair  $\langle \tau_1(m), \tau_2(l) \rangle$  by (23).

**End If.**

i) **Modified Probabilistic Crowding based Niching:** Select either  $\vec{Z}_i(t)$  or its nearest neighbor  $\vec{X}_j(t)$  as follows.

$$\vec{X}_j(t+1) = \vec{Z}_i(t), \quad \text{if } \text{rand}(0,1) \leq p(\vec{X}_j(t) \leftarrow \vec{Z}_i(t))$$

$$= \vec{X}_j(t), \quad \text{otherwise}$$

j) Set  $\vec{X}^{best}(t+1) = \vec{X}_j(t+1)$ , if  $\bar{J}(\vec{X}_j(t+1)) \leq \bar{J}(\vec{X}^{best}(t))$ .

**End For.**

2.(iii) Set  $t \leftarrow t+1$ .

**End While;**

**End.**

---

#### IV. EXPERIMENTS AND RESULTS

This section abridges the experiments undertaken to analyze the relative performance of the proposed algorithm with its competitors and summarizes the results when the efficacy of the contender algorithms is tested on noisy benchmark functions.

##### A. Benchmark Functions and Evaluation Metrics

**1. Benchmark Functions:** The performance of the proposed NDE algorithm is analyzed here with respect to following two sets of well-known benchmark functions, which effectively represent such real world fitness landscapes.

**BF1:** It comprises 28 CEC'2013 recommended benchmark functions [29]. Among these benchmarks, five (f01–f05) are unimodal, fifteen (f06–f20) are multimodal and the remaining eight (f21–f28) are composition test functions. These functions here are made noisy by additive noise, discussed in the next sub-section.

The noisy version of each function of BF1 is realized by contaminating each function with additive noise samples drawn from certain distribution, including Gaussian [9], Poisson [26], Rayleigh [22], exponential [34], Gamma [1], Cauchy [22] and random [8].

**BF2:** It consists of 30 noisy benchmark functions, recommended in GECCO'2010 [17]. Among these benchmarks, six (101–106) are affected with moderate noise, fifteen (107–121) with severe noise and the rest (122–130) are highly multimodal functions with severe noise.

**2. Comparative Framework and Parameter Settings:** The comparative framework is formed by selecting widely popular algorithms in NSOOP domain. The list includes increased population-uncertainty handling-covariance matrix adaptation-evolutionary strategy (IPOP-UH-CMA-ES) [21], memetic for uncertainties DE (MUDE) [38], subset based LA incorporated particle swarm optimization (LAPSO) [73], noise tolerant genetic algorithm (NTGA) [23], immune algorithm with adaptive sampling (IAAS) [75], PSO with OCBA and hypothesis test (PSOHT) [40], bare bones PSO with chaotic jump (BBPSO-CJ) [35], opposition-based DE (ODE) [47], DE with random scale factor and stochastic selection (DE-RSF-SS) [14], and genetic algorithm with memory-based fitness evaluation (MFEGA) [62]. For all competitors, a population of 50 solutions is initialized with same random seed except IPOP-UH-CMA-ES to ensure fair comparison. Apart from this, the maximum function evaluations (FEs) for all contenders is fixed at  $10^4 \times D$  for  $D$ -dimensional search space problem. The parametric set-up adopted for all the competitor algorithms is given in Table-I.

TABLE I-A: PARAMETER SETTINGS OF COMPARATIVE FRAMEWORKS IN D-DIMENSIONAL SEARCH SPACE

NDE		IPOP-UH-CMA-ES		MUDE		LAPSO		NTGA		
Parameters	Values	Parameters	Values	Parameters	Values	Parameters		Parameters	Values	
Sample size	Adaptive	Number of newly sampled candidate solutions in each generation	$4 + \lfloor 3 \ln D \rfloor$	Minimum scale factor	0.1	Network topology		Star	Size of mating pool	2
Size of state-transition table $M \times L \times N$	10×10×20	Learning rate for mean	1	Maximum scale factor	0.9	Acceleration parameters	$c_1$	1.49618	Size of buffer pool	50
Fitness estimation	Weighted averaging	$\alpha_{cov}$	0.5	Probability to update scale factor	0.1		$c_2$	1.49618	Initial sample size	3
Mutation	AM	$\lambda_{mintarget}$	0.66	Probability of generating offspring using scale factor golden selection search (SFGSS)	0.04	Inertia		0.729844	Crossover rate	0.6
$\eta_1^l$	0.6	$\alpha_{old}$	0.5	Probability of generating offspring using scale factor hill climb (SFHC)	0.07	Maximum velocity		$0.25 z^{\max} - z^{\min} $	Mutation operator	Polynomial
$\eta_2^l$	0.6			Initial exploratory radius of SFHC	0.1	Total computing budget constraint		250	Mutation rate	1/D
Selection	Probabilistic crowding			Maximum budget for SFGSS	40 <i>FEs</i>	Initial simulation replications of each design (solution)		5	Selection	Probabilistic restricted tournament
SLA parameter	0.5 each			Maximum budget for SFHC	100 <i>FEs</i>	Additional simulation replications		50		
				Maximum number of evaluations per each individual for noise analysis based selection	30	Threshold		0.9		
						Subset capacity		3		

TABLE I-B: PARAMETER SETTINGS OF COMPARATIVE FRAMEWORKS IN D-DIMENSIONAL SEARCH SPACE

IAAS		PSOHT		BBPSO-CJ		ODE		DE-RSF-SS		MFEGA	
Parameters	Values	Parameters	Values	Parameters	Values	Parameters	Values	Parameters	Values	Parameters	Values
Initial sample size	2	Acceleration parameters	$c_1$ 3.85	Acceleration parameters	$c_1$ 0.25	Scale factor	0.5	Scale factor	[0.5, 1]	Crossover	UNDX
Control parameter for maximum sample size	100		$c_2$ 0.25		$c_2$ 0.25	Differential amplification factor	0.1	Crossover rate	0.9	Children size	5
Reproduction	Dyna mic	Constriction factor 0.729		Constriction factor 0.729		Crossover rate	0.9				
Mutation	Hyper-mutation	Total computing budget constraint 2000		Scaling parameter [0.1%, 10%] of $D$		Mutation	DE/rand/1/bin				
Control parameters for mutation probability	$\lambda$	Initial simulation replications of each design (solution) 10		Maximum stagnation interval 5	5	Jumping rate constant 0.3	0.3				
	$T_0$										
Gaussian mutation parameter	$\beta_1$	Additional simulation replications 10		Range of chaotic jump [-1, 1]	[-1, 1]						
	$\beta_2$										
Adjustable parameter for suppression radius	0.4			Chaos control parameter	4						
Control parameter for suppression prob.	0.3										

**3. Performance Metric:** Function error value (FEV) is used to investigate the comparative analysis of performance of NDE and its contenders. FEV is defined by  $|J(\bar{X}) - J(\bar{X}^*)|$  where  $J(\bar{X})$  denotes the median objective function value obtained by an algorithm over  $K$  runs (say 50) and  $J(\bar{X}^*)$  represents the true fitness of the global optimum of a particular benchmark function. Smaller the FEV better is the performance of the algorithm.

### B. Results and Performance Analysis

This section first examines the effectiveness of settings of the control parameters tabulated in Table-I. the section then provides the relative merits of the individual extensions embedded in NDE through computer simulations. A comparative analysis of the performance of NDE with its competitors is also undertaken in this section. The latter part of the experiment attempts to improve the performance of some well-known self-adaptive DE variants by integrating the proposed uncertainty management strategies in the algorithms for quality performance in the noisy fitness landscape. It is worth mentioning that all the experiments are undertaken with respect to both noisy versions of BF1 and BF2 benchmark functions for different dimensions of the search space. The BF1 benchmark functions are contaminated with all possible settings of different noise models to validate the proficiency of the proposed noise handling mechanisms. However, a few results are reported in the paper for space limitation.

### 1. Sensitivity Analysis of Control Parameters of NDE

A relative analysis of the performance of the proposed NDE for different settings of its control parameters, including the number of states of the state-action table  $M$  and  $L$ , the maximum sample size  $N$ , the minimum value of sale factors  $\eta_1^l$  and  $\eta_2^l$ , the ASSLA control parameters  $\beta_1$  and  $\beta_2$  is undertaken in this section. The sensitivity analysis approach proposed in [57], [60] is used here to validate the effect of individual NDE control parameter  $p \in \{M, L, N, \eta_1^l, \eta_2^l, \beta_1, \beta_2\}$  on its optimization performance.

To examine the influence of a specific parameter  $p$  from  $\{M, L, N, \eta_1^l, \eta_2^l, \beta_1, \beta_2\}$ , keeping the rest of the parameters (excluding  $p$ ) fixed at the values given in Table-I, only  $p$  is varied from its permissible

minimum value (with Label  $-1$ ) to its maximum value (with Label  $+1$ ) as indicated in Table-II. For example, during examining the effect of the maximum sample size  $N$ , it is varied from 15 to 25 with other NDE control parameters set as  $M = 20$ ,  $L = 20$ ,  $\eta_1^l = 0.6$ ,  $\eta_2^l = 0.6$ ,  $\beta_1 = 0.5$  and  $\beta_2 = 0.5$  following Table-I.

TABLE II: LOW, MEDIUM AND HIGH VALUES OF CONTROL PARAMETERS OF NDE USED FOR SENSITIVITY ANALYSIS

Control Parameters		Low (Label $-1$ )	Medium (Label 0)	High (Label $+1$ )
Number of states in state-action table	$M$	5	10	15
	$L$	5	10	15
Maximum sample size $N$		10	20	30
Minimum scale factors	$\eta_1^l$	0.1	0.6	1
	$\eta_2^l$	0.1	0.6	1
ASSLA control parameter	$\alpha$	0.2	0.5	1
	$\beta$	0.2	0.5	1

To assess the sensitivity of a specific control parameter  $p$  set at a specific label  $l \in \{-1, 0, +1\}$ , NDE is run 50 times on a particular benchmark function  $J$  in  $D$ -dimensional search space and the corresponding FEV values and run-times (RTs) are recorded. The effect of  $p$  with label  $l$  on the performance of NDE is then captured by its sensitivity factor, given by

$$SF_{J,D,l}(p) = \left( \frac{FEV_{\text{med}}}{FEV_{\text{max}}} \cdot \frac{RT_{\text{med}}}{RT_{\text{max}}} \right)_{J,D,l} \quad (25)$$

Here  $FEV_{\text{med}}$  (and  $FEV_{\text{max}}$ ) and  $RT_{\text{med}}$  (and  $RT_{\text{max}}$ ) respectively denote the median (and maximum) value of FEV and RT obtained after 50 independent runs of NDE. Here  $J \in \{1, 2, \dots, 58\}$  (28 BF1 and 30 BF2 benchmark functions) indicates the benchmark function index and  $D \in \{10, 30, 50, 100\}$  represents the search space dimension. Evidently, smaller the value of  $SF_{J,D,l}(p)$ , better is the performance of NDE on the function  $J$  in the  $D$ -dimensional search space with the parameter  $p$  fixed at its  $l$ -th label. The sensitivity factor of  $p$  at its  $l$ -th label,  $SF_l(p)$  is now defined by (26).

$$SF_l(p) = \frac{1}{4} \left[ \sum_{D \in \{10, 30, 50, 100\}} \left[ \frac{1}{58} \sum_{J=1}^{58} SF_{J,D,l}(p) \right] \right] \quad (26)$$

Small value of  $SF_l(p)$  demonstrates the quality performance of NDE to achieve the optimal trade-off between FEV and RT metrics. Fig. 5 portrays the measures of  $SF_l(p)$  for all possible combinations of  $p$  and  $l$  with  $p = \{M, L, N, \eta_1^l, \eta_2^l, \beta_1, \beta_2\}$  and  $l = \{-1, 0, +1\}$  (i.e., low, medium, high). Fig. 5 reveals that the setting of each NDE control parameter  $p$  as given in Table-I yield the minimum SF value among all possible labels of  $p$ .

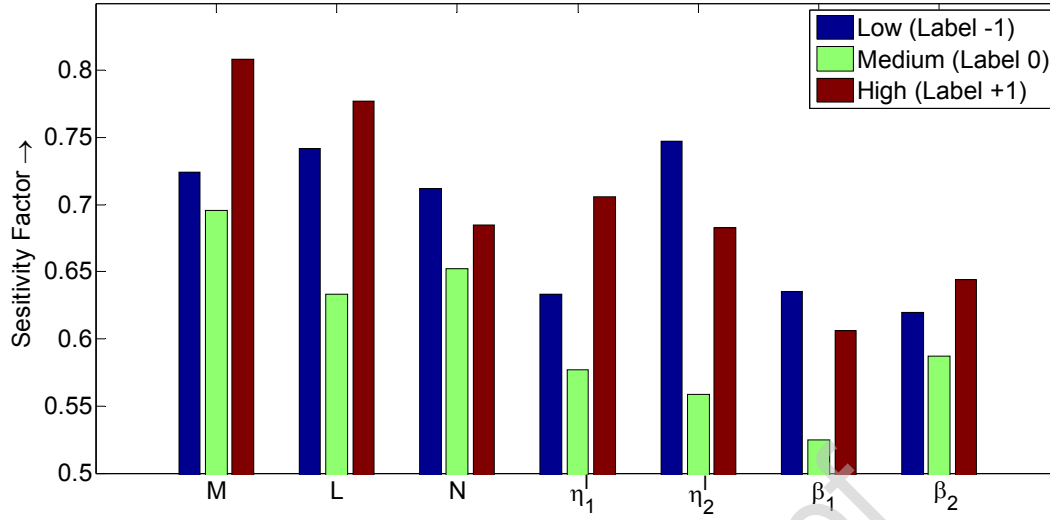


Fig. 5 Sensitivity analysis of control parameters of NDE

Similar approach is considered to validate the parameter settings of the rest of the algorithms in the comparative framework. Tables III-A to III-J present the SFs of the control parameters for their different label of settings (within parenthesis). The minimum SF value and the corresponding parameter setting are set in bold. The results reported in Table-III also support the optimal settings of the parameters as given in Table-I.

TABLE III-A: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF IPOP-UH-CMA-ES

Control Parameters	SF		
	(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Learning rate for mean	0.692 (0.2)	0.691 (0.5)	<b>0.489 (1)</b>
$\alpha_{cov}$	0.676 (0.1)	<b>0.481 (0.5)</b>	0.653 (1)
$\lambda_{mintarget}$	0.678 (0.2)	<b>0.474 (0.66)</b>	0.608 (1)
$\alpha_{old}$	0.586 (0.1)	<b>0.519 (0.5)</b>	0.689 (1)

TABLE III-B: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF MUDE

Control Parameters		SF		
		(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Minimum scale factor		0.854 (0.05)	<b>0.679 (0.1)</b>	0.796 (0.15)
Maximum scale factor		0.838 (0.85)	<b>0.552 (0.9)</b>	0.656 (0.95)
Probability to update scale factor		0.793 (0.05)	<b>0.742 (0.1)</b>	0.855 (0.15)
Probability of generating offspring using	SFGSS	0.845 (0.02)	<b>0.513 (0.04)</b>	0.814 (0.06)
	SFHC	0.774 (0.05)	<b>0.751 (0.07)</b>	0.780 (0.09)
Initial exploratory radius of SFHC		0.645 (0.01)	<b>0.563 (0.1)</b>	0.742 (0.2)
Maximum budget for SFGSS		0.761 (20 FEs)	<b>0.511 (40 FEs)</b>	0.602 (60 FEs)
Maximum budget for SFHC		0.604 (80 FEs)	<b>0.517 (100 FEs)</b>	0.535 (120 FEs)
Maximum number of evaluations per each individual		0.757 (10)	<b>0.617 (30)</b>	0.851 (50)

TABLE III-C: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF LAPSO

Control Parameters		<i>SF</i>		
		(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Acceleration Parameters	$c_1$	0.662 (0.5)	<b>0.512 (1.496)</b>	0.641 (2)
	$c_2$	0.794 (0.5)	<b>0.569 (1.496)</b>	0.783 (2)
Total computing budget constraint		0.681 (100)	<b>0.664 (250)</b>	0.739 (400)
Initial simulation replications of each design		0.779 (2)	<b>0.602 (5)</b>	0.762 (10)
Additional simulation replications		0.742 (40)	<b>0.560 (50)</b>	0.751 (60)
Threshold		0.855 (0.85)	<b>0.544 (0.9)</b>	0.684 (0.95)
Subset capacity		0.716 (2)	<b>0.582 (3)</b>	0.625 (5)
Inertia		0.687 (0.528)	<b>0.594 (0.729)</b>	0.777 (0.918)

TABLE III-D: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF NTGA

Control Parameters	<i>SF</i>		
	(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Size of mating pool	<b>0.788 (2)</b>	0.840 (4)	0.859 (6)
Size of buffer pool	0.747 (40)	<b>0.637 (50)</b>	0.640 (60)
Initial sample size	<b>0.668 (3)</b>	0.669 (5)	0.826 (10)
Crossover rate	0.819 (0.5)	<b>0.665 (0.6)</b>	0.850 (0.7)

TABLE III-E: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF IAAS

Control Parameters		<i>SF</i>		
		(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Initial sample size		<b>0.552 (2)</b>	0.575 (4)	0.616 (6)
Control parameter for maximum sample size		0.728 (50)	<b>0.617 (100)</b>	0.668 (150)
Control parameters for mutation probability	$\lambda$	0.818 (0.85)	<b>0.700 (0.9)</b>	0.715 (0.95)
	$T_0$	0.788 (50)	<b>0.590 (100)</b>	0.855 (150)
Gaussian mutation parameter	$\beta_1$	<b>0.629 (0.01)</b>	0.708 (0.2)	0.786 (0.4)
	$\beta_2$	0.501 (0.1)	<b>0.492 (0.3)</b>	0.692 (0.5)
Adjustable parameter for suppression radius		0.862 (0.2)	<b>0.524 (0.4)</b>	0.797 (0.6)
Control parameter for suppression probability		0.708 (0.1)	<b>0.474 (0.3)</b>	0.667 (0.5)

TABLE III-F: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF PSOHT

Control Parameters		<i>SF</i>		
		(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Acceleration Parameters	$c_1$	0.867 (1.5)	<b>0.585 (3.85)</b>	0.703 (5.5)
	$c_2$	0.639 (0.1)	<b>0.595 (0.25)</b>	0.807 (0.5)
Constriction factor		0.836 (0.523)	<b>0.737 (0.729)</b>	0.806 (0.864)
Total computing budget constraint		0.840 (1000)	<b>0.683 (2000)</b>	0.771 (3000)
Initial simulation replications of each design		0.646 (5)	<b>0.644 (10)</b>	0.820 (15)
Additional simulation replications		0.853 (5)	<b>0.708 (10)</b>	0.798 (15)

TABLE III-G: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF BBPSO-CJ

Control Parameters		<i>SF</i>		
		(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Acceleration Parameters	$c_1$	0.715 (0.1)	<b>0.71 (0.25)</b>	0.906 (0.5)
	$c_2$	0.665 (0.1)	<b>0.651 (0.25)</b>	0.888 (0.5)
Constriction factor		0.673 (0.523)	<b>0.556 (0.729)</b>	0.837 (0.864)
Maximum stagnation interval		0.611 (2)	<b>0.550 (5)</b>	0.679 (10)
Chaos control parameter		0.905 (2)	<b>0.565 (4)</b>	0.911 (6)

TABLE III-H: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF ODE

Control Parameters		<i>SF</i>		
		(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Scale factor		0.786 (0.3)	<b>0.765 (0.5)</b>	0.776 (0.7)
Differential amplification factor		<b>0.793 (0.1)</b>	0.838 (0.3)	0.845 (0.5)
Crossover rate		0.881 (0.8)	<b>0.705 (0.9)</b>	0.862 (1)
Jumping rate constant		0.759 (0.1)	<b>0.713 (0.3)</b>	0.843 (0.5)

TABLE III-I: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF DE-RSF-SS

Control Parameters		<i>SF</i>		
		(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Minimum scale factor		0.734 (0.3)	<b>0.730 (0.5)</b>	0.746 (0.7)
Crossover rate		0.753 (0.8)	<b>0.691 (0.9)</b>	0.752 (1)

TABLE III-J: SENSITIVITY ANALYSIS OF CONTROL PARAMETERS OF MFEGA

Control Parameters		<i>SF</i>		
		(Low Value with Label -1)	(Medium Value with Label 0)	(High Value with Label +1)
Children Size		0.832 (2)	<b>0.624 (5)</b>	0.878 (7)

## 2. Effectiveness of Different Extensions of NDE

In section III, NDE is developed by integrating four strategies in the traditional DE, including EEF, AM, ASSLA, and NS to solve NSOOPs. The proposed strategies treat four different issues to overcome deceptive effect of noise in the fitness measure and hence they are mutually independent of each other. To examine the efficacy of individual four policies implanted in NDE, as well as their all possible combinations, a list of fifteen noisy DE variants are considered. The list consists of DE-EEF, DE-AM, DE-ASSLA, DE-EEF-AM, DE-EEF-ASSLA, DE-AM-ASSLA, DE-EEF-AM-ASSLA, DE-NS, DE-EEF-NS, DE-AM-NS, DE-ASSLA-NS, DE-EEF-AM-NS, DE-EEF-ASSLA-NS, DE-AM-ASSLA-NS, and NDE.

The effectiveness of the above-mentioned fifteen groups is tested with respect to the FEV metric values while minimizing noisy version of 28 functions of BF1. The noisy version is realized by contaminating each function of BF1 with additive random noise samples of amplitude within  $\pm 30\%$  of the true fitness values. The results, comprising the median and the inter-quartile range IQR (within parentheses) of the FEV metric values obtained by each noisy DE variant after 50 independent runs, each with 300000 FEs in a 30-dimensional search space are reported in Table-IV. For each benchmark

function, the best metric value is set in bold. It is evident from Table-IV that NDE, comprising all four strategies, outperforms other noisy DE variants (with less than four policies) in achieving lower value of FEV metric. It substantiates the significance of all four policies in handling the presence of noise in the fitness landscape.

TABLE IV-A: COMPARISON OF INDIVIDUAL AND COMBINATION OF PROPOSED EXTENSIONS EMBEDDED IN NDE WITH RESPECT TO FEV METRIC VALUES FOR 50 INDEPENDENT RUNS IN PRESENCE OF RANDOM NOISE (OF AMPLITUDE WITHIN  $\pm 30\%$  OF TRUE FITNESS VALUES) IN FITNESS LANDSCAPE FOR BF1: f01 to f24

Func.	DE-EEF	DE-AM	DE-ASSLA	DE-EEF-AM	DE-EEF-ASSLA	DE-AM-ASSLA	DE-EEF-AM-ASSLA	DE-NS	DE-EFE-NS	DE-AM-NS	DE-ASSLA-NS	DE-EFE-AM-NS	DE-EFE-ASSLA-NS	DE-AM-ASSLA-NS	NDE
f01	1.78e-09 (1.1e-08)	1.89e-09 (1.2e-08)	1.26e-09 (1.0e-08)	7.44e-10 (7.6e-09)	1.51e-10 (6.7e-09)	2.59e-10 (7.0e-09)	<b>0.00e+00</b> ( <b>0.0e+00</b> )	1.80e-09 (1.2e-08)	5.74e-10 (7.0e-09)	9.06e-10 (7.6e-09)	2.14e-10 (7.0e-09)	8.04e-12 (5.4e-09)	<b>0.00e+00</b> ( <b>0.0e+00</b> )	7.57e-12 (2.1e-09)	<b>0.00e+00</b> ( <b>0.0e+00</b> )
f02	9.12e+02 (1.2e+02)	1.51e+03 (1.0e+02)	7.09e+02 (1.0e+02)	5.41e+02 (6.5e+01)	2.35e+02 (2.4e-01)	1.81e+02 (3.0e+01)	6.52e-06 (9.0e-05)	9.42e+02 (1.2e+02)	2.64e+02 (5.0e+01)	6.41e+02 (8.8e+01)	9.93e+01 (2.0e+01)	2.21e-02 (1.7e-01)	2.68e-06 (8.2e-05)	1.80e-02 (1.6e-01)	<b>9.31e-07</b> ( <b>6.0e-05</b> )
f03	2.33e+02 (1.5e+01)	2.48e+02 (4.0e+01)	1.89e+02 (2.6e+01)	1.37e+02 (2.1e+01)	1.24e+01 (2.6e+00)	7.18e+01 (7.3e+00)	<b>2.27e-03</b> ( <b>2.4e-02</b> )	2.42e+02 (3.0e+01)	8.14e+01 (7.8e+00)	1.67e+02 (2.5e+01)	5.35e+01 (6.4e+00)	8.99e+01 (2.5e+00)	2.22e-02 (2.0e-03)	8.23e+00 (6.5e+00)	1.70e-02 (1.4e-03)
f04	4.84e+01 (8.8e-04)	5.61e+01 (1.0e-03)	3.56e+01 (8.1e-04)	1.38e+01 (6.8e-04)	1.17e-01 (2.6e-04)	3.09e+00 (4.6e-04)	7.17e-02 (2.7e-05)	4.86e+01 (9.7e-04)	8.16e+00 (4.8e-04)	3.13e+01 (7.5e-04)	1.67e+00 (4.4e-04)	9.76e-02 (2.0e-04)	5.48e-02 (4.4e-06)	9.74e-02 (1.8e-04)	<b>3.09e-02</b> ( <b>2.0e-06</b> )
f05	1.97e-06 (3.0e-03)	2.33e-06 (4.3e-03)	2.03e-06 (4.4e-03)	1.52e-06 (4.2e-03)	9.08e-09 (1.1e-03)	7.11e-07 (1.3e-03)	<b>0.00e+00</b> ( <b>7.2e-04</b> )	2.18e-06 (3.4e-03)	8.58e-07 (2.5e-03)	1.55e-06 (4.2e-03)	3.90e-07 (1.2e-03)	5.15e-09 (1.1e-03)	<b>0.00e+00</b> ( <b>5.2e-04</b> )	1.90e-09 (1.1e-03)	<b>0.00e+00</b> ( <b>2.7e-04</b> )
f06	4.03e+01 (3.6e+00)	4.19e+01 (6.4e+00)	4.20e+01 (6.3e+00)	4.20e+01 (6.0e+00)	3.12e+01 (9.6e-02)	3.58e+01 (4.4e+00)	1.75e+01 (2.3e-05)	4.12e+01 (5.8e+00)	4.14e+01 (5.8e+00)	4.20e+01 (6.0e+00)	3.57e+01 (1.3e+00)	3.11e+01 (5.8e-02)	4.29e+00 (2.0e-05)	2.72e+01 (1.5e-02)	<b>3.01e+00</b> ( <b>1.3e-05</b> )
f07	4.47e+01 (9.1e+00)	4.73e+01 (9.1e+00)	4.88e+01 (9.0e+00)	4.20e+01 (9.0e+00)	3.08e+01 (5.7e+00)	3.56e+01 (7.5e+00)	2.87e+01 (3.5e+00)	4.60e+01 (9.1e+00)	3.56e+01 (8.7e+00)	4.87e+01 (9.0e+00)	3.56e+01 (7.2e+00)	3.04e+01 (5.6e+00)	1.65e+01 (3.2e+00)	2.92e+01 (5.5e+00)	<b>1.54e+01</b> ( <b>2.8e+00</b> )
f08	2.10e+01 (4.5e-02)	2.10e+01 (4.5e-02)	2.10e+01 (4.5e-02)	2.10e+01 (4.4e-02)	2.08e+01 (3.8e-02)	2.10e+01 (4.1e-02)	2.08e+01 (3.3e-02)	2.10e+01 (4.5e-02)	2.10e+01 (4.2e-02)	2.10e+01 (4.4e-02)	2.10e+01 (4.1e-02)	2.08e+01 (3.8e-02)	2.08e+01 (3.0e-02)	2.08e+01 (3.5e-02)	<b>2.08e+01</b> ( <b>3.0e-02</b> )
f09	4.80e+01 (2.5e+00)	4.88e+01 (3.2e+00)	4.60e+01 (2.4e+00)	3.83e+01 (2.1e+00)	2.78e+01 (2.1e+00)	3.60e+01 (2.2e+00)	2.50e+01 (1.8e+00)	4.83e+01 (2.7e+00)	3.76e+01 (2.3e+00)	3.85e+01 (2.3e+00)	3.37e+01 (2.2e+00)	2.20e+01 (2.1e+00)	2.33e+01 (1.3e+00)	1.86e+01 (1.5e+00)	<b>9.35e+00</b> ( <b>1.2e+00</b> )
f10	1.74e-02 (6.4e-03)	1.94e-02 (1.0e-02)	4.20e-03 (4.0e-03)	5.50e-04 (3.0e-03)	1.46e-03 (2.0e-04)	7.68e-04 (9.0e-04)	5.68e-10 (3.5e-05)	1.75e-02 (8.5e-03)	1.50e-03 (1.1e-03)	2.90e-03 (3.2e-03)	5.12e-04 (8.3e-04)	1.00e-03 (2.0e-04)	3.75e-10 (2.6e-05)	5.11e-07 (5.4e-05)	<b>3.18e-10</b> ( <b>2.2e-05</b> )
f11	5.41e+00 (4.5e-01)	1.21e+01 (1.1e+00)	7.47e+00 (2.7e-01)	5.65e+00 (1.5e-01)	3.58e+00 (2.2e-02)	5.93e+00 (2.5e-01)	1.38e+00 (4.0e-03)	5.71e+00 (5.5e-01)	6.85e+00 (2.8e-01)	6.70e+00 (2.6e-01)	5.26e+00 (1.4e-01)	3.53e+00 (2.0e-02)	1.32e+00 (2.8e-03)	3.30e+00 (4.1e-02)	<b>7.75e-01</b> ( <b>2.5e-01</b> )
f12	8.57e+01 (1.3e+01)	8.54e+01 (2.0e+01)	6.84e+01 (1.0e+01)	6.50e+01 (9.0e+00)	3.02e+01 (6.3e+00)	6.30e+01 (7.8e+00)	1.95e+01 (1.7e+00)	8.61e+01 (1.4e+01)	6.32e+01 (8.0e+00)	6.56e+01 (9.0e+00)	4.30e+01 (7.3e+00)	2.94e+01 (6.2e+00)	1.51e+01 (1.7e+00)	2.72e+01 (5.8e+00)	<b>1.15e+01</b> ( <b>1.3e+00</b> )
f13	1.47e+02 (2.0e+01)	1.65e+02 (2.4e+01)	1.33e+02 (1.8e+01)	1.07e+02 (1.8e+01)	8.04e+01 (8.0e+00)	1.00e+02 (1.2e+01)	6.40e+01 (3.8e+00)	1.58e+02 (2.3e+01)	1.04e+02 (1.5e+01)	1.25e+02 (1.8e+01)	9.84e+01 (1.0e+01)	7.55e+01 (5.0e+00)	5.56e+01 (3.1e+00)	7.42e+01 (5.0e+00)	<b>4.00e+01</b> ( <b>3.0e+00</b> )
f14	4.48e+02 (1.6e+02)	5.96e+02 (2.0e+02)	3.18e+02 (1.0e+02)	1.10e+02 (5.8e+01)	2.03e+01 (9.3e+00)	3.66e+01 (1.5e+01)	1.36e+01 (7.4e+00)	5.41e+02 (1.8e+02)	4.23e+01 (2.0e+01)	1.60e+02 (8.0e+01)	2.22e+02 (1.4e+01)	2.03e+01 (9.0e+00)	1.30e+01 (5.4e+00)	1.96e+01 (8.0e+00)	<b>6.88e+00</b> ( <b>3.3e+00</b> )
f15	6.30e+03 (4.6e+02)	6.35e+03 (5.0e+02)	5.30e+03 (4.5e+02)	4.35e+03 (4.0e+02)	3.26e+03 (4.0e+02)	4.31e+03 (4.1e+02)	1.20e+03 (2.7e+02)	6.30e+03 (5.0e+02)	5.18e+03 (4.3e+02)	5.10e+03 (4.4e+02)	3.04e+03 (4.0e+02)	2.37e+03 (3.4e+02)	1.14e+03 (2.7e+02)	1.65e+03 (2.8e+02)	<b>7.40e+02</b> ( <b>2.7e+02</b> )
f16	1.61e+00 (2.0e-01)	1.67e+00 (2.0e-01)	1.48e+00 (2.0e-01)	1.45e+00 (1.8e-01)	8.25e-01 (4.2e-02)	1.38e+00 (1.2e-01)	6.60e-01 (3.8e-02)	1.66e+00 (2.0e-01)	1.43e+00 (1.5e-01)	1.47e+00 (1.8e-01)	9.90e-01 (6.7e-02)	8.05e-01 (1.4e-02)	2.20e-01 (2.4e-02)	7.90e-01 (4.1e-02)	<b>9.12e-02</b> ( <b>2.3e-02</b> )
f17	5.60e+01 (1.6e+00)	5.67e+01 (1.6e+00)	5.50e+01 (1.3e+00)	5.27e+01 (7.8e-01)	5.06e+01 (3.0e-02)	5.11e+01 (1.4e-01)	4.70e+01 (7.4e-03)	5.61e+01 (1.6e+00)	5.12e+01 (2.0e-01)	5.43e+01 (9.2e-01)	5.08e+01 (5.4e-02)	5.06e+01 (3.0e-02)	4.50e+01 (6.3e-03)	5.06e+01 (2.1e-02)	<b>4.20e+01</b> ( <b>1.2e-03</b> )
f18	1.40e+02 (3.0e+01)	1.50e+02 (3.3e+01)	1.31e+02 (2.5e+01)	1.30e+02 (2.1e+01)	1.23e+02 (1.3e+01)	1.26e+02 (2.0e+01)	8.97e+01 (9.5e+00)	1.50e+02 (3.0e+01)	1.27e+02 (2.0e+01)	1.30e+02 (2.3e+01)	1.25e+02 (1.7e+01)	1.23e+02 (1.2e+01)	<b>6.26e+01</b> ( <b>8.4e+00</b> )	1.20e+02 (1.1e+02)	<b>6.26e+01</b> ( <b>8.0e+00</b> )
f19	3.62e+00 (5.6e-01)	3.86e+00 (6.3e-01)	3.47e+00 (4.8e-01)	3.41e+00 (4.6e-01)	2.96e+00 (3.3e-01)	3.35e+00 (4.3e-01)	2.34e+00 (3.0e-01)	3.85e+00 (6.2e-01)	3.38e+00 (4.5e-01)	3.46e+00 (4.7e-01)	3.32e+00 (4.0e-01)	2.60e+00 (3.0e-01)	2.24e+00 (3.0e-01)	2.21e+00 (2.7e-01)	<b>2.20e+00</b> ( <b>2.6e-01</b> )
f20	1.92e+01 (6.2e-01)	1.97e+01 (6.6e-01)	1.92e+01 (6.1e-01)	1.91e+01 (5.6e-01)	1.90e+01 (4.6e-01)	1.91e+01 (5.3e-01)	1.58e+01 (3.8e-01)	1.94e+01 (6.5e-01)	1.91e+01 (5.4e-01)	1.91e+01 (5.8e-01)	1.91e+01 (5.0e-01)	1.88e+01 (4.6e-01)	1.37e+01 (3.7e-01)	1.87e+01 (4.1e-01)	<b>1.35e+01</b> ( <b>3.7e-01</b> )
f21	3.27e+02 (2.4e+02)	3.77e+02 (2.5e+02)	2.88e+02 (2.0e+02)	2.55e+02 (2.0e+02)	2.40e+02 (1.0e+02)	2.46e+02 (1.8e+02)	2.37e+02 (1.1e+01)	3.68e+02 (2.5e+02)	2.50e+02 (1.8e+02)	2.71e+02 (2.1e+02)	2.45e+02 (1.4e+02)	2.40e+02 (6.2e+01)	2.34e+02 (8.1e+00)	2.37e+02 (5.8e+01)	<b>2.31e+02</b> ( <b>5.3e+00</b> )
f22	2.92e+02 (2.5e+02)	3.88e+02 (2.7e+02)	2.44e+02 (1.0e+02)	1.08e+02 (5.0e+01)	4.13e+02 (2.4e+01)	1.05e+02 (3.5e+01)	3.44e+01 (1.3e+01)	3.80e+02 (2.6e+02)	1.22e+02 (4.6e+01)	2.27e+02 (9.2e+01)	8.95e+01 (3.0e+01)	2.65e+01 (2.3e+01)	3.11e+01 (1.2e+01)	2.40e+01 (2.1e+01)	<b>2.97e+01</b> ( <b>9.2e+00</b> )
f23	7.43e+03 (6.2e+02)	7.93e+03 (6.6e+02)	5.82e+03 (5.8e+02)	5.01e+03 (5.3e+02)	2.37e+03 (4.5e+02)	3.07e+03 (5.1e+02)	9.55e+02 (3.2e+02)	7.64e+03 (6.2e+02)	3.15e+03 (5.2e+02)	5.37e+03 (5.5e+02)	2.88e+03 (5.0e+02)	1.42e+03 (4.2e+02)	8.41e+02 (3.1e+02)	1.07e+03 (3.8e+02)	<b>7.05e+02</b> ( <b>3.1e+02</b> )
f24	2.74e+02 (1.1e+01)	2.96e+02 (1.3e+01)	2.87e+02 (1.2e+01)	2.62e+02 (1.0e+01)	2.25e+02 (7.7e+00)	2.38e+02 (9.8e+00)	2.20e+02 (3.8e+00)	2.80e+02 (1.2e+01)	2.40e+02 (1.0e+01)	2.67e+02 (1.1e+01)	2.36e+02 (9.0e+00)	2.25e+02 (6.3e+00)	2.14e+02 (3.4e+00)	2.24e+02 (6.0e+00)	<b>2.12e+02</b> ( <b>2.7e+00</b> )



TABLE IV-B: COMPARISON OF INDIVIDUAL AND COMBINATION OF PROPOSED EXTENSIONS EMBEDDED IN NDE WITH RESPECT TO *FEV* METRIC VALUES FOR 50 INDEPENDENT RUNS IN PRESENCE OF RANDOM NOISE (OF AMPLITUDE WITHIN  $\pm 30\%$  OF TRUE FITNESS VALUES) IN FITNESS LANDSCAPE FOR *f25* TO *f28*

Func.	DE-EEF	DE-AM	DE-ASSLA	DE-EEF-AM	DE-EEF-ASSLA	DE-AM-ASSLA	DE-EEF-AM-ASSLA	DE-NS	DE-EFE-NS	DE-AM-NS	DE-ASSLA-NS	DE-EFE-AM-NS	DE-EFE-ASSLA-NS	DE-AM-ASSLA-NS	NDE
<i>f25</i>	3.30e+02 (1.2\1e+01)	3.34e+02 (1.1e+01)	3.11e+02 (1.0e+01)	2.94e+02 (9.3e+00)	3.08e+02 (9.0e+00)	3.03e+02 (9.2e+00)	2..65e+02 (4.4e+00)	3.31e+02 (1.1e+01)	3.13e+02 (1.0e+01)	3.00e+02 (9.4e+00)	2.94e+02 (9.0e+00)	3.06e+02 (7.4e+00)	2.44e+02 (4.3e+00)	3.01e+02 (6.3e+00)	<b>2.18e+02</b> <b>(3.8e+00)</b>
<i>f26</i>	2.43e+02 (5.0e+01)	2.47e+02 (6.0e+01)	2.40e+02 (4.2e+01)	2.27e+02 (3.5e+01)	2.00e+02 (2.0e+01)	2.05e+02 (2.1e+01)	1.94e+02 (5.6e+01)	2.47e+02 (5.6e+01)	2.13e+02 (2.4e+01)	2.35e+02 (3.6e+01)	2.03e+02 (2.1e+01)	2.00e+02 (1.2e+01)	1.80e+02 (1.6e+00)	2.00e+02 (9.1e+00)	<b>1.77e+02</b> <b>(1.5e+00)</b>
<i>f27</i>	1.00e+03 (1.2e+02)	1.01e+03 (1.3e+02)	9.80e+02 (1.3e+02)	9.18e+02 (1.2e+02)	7.35e+02 (1.0e+02)	9.30e+02 (1.2e+02)	3.97e+02 (3.7e+01)	1.00e+03 (1.3e+02)	9.66e+02 (1.2e+02)	9.52e+02 (1.3e+02)	8.67e+02 (1.1e+02)	6.60e+02 (7.7e+01)	3.57e+02 (3.5e+01)	6.56e+02 (7.2e+01)	<b>3.40e+02</b> <b>(3.4e+01)</b>
<i>f28</i>	4.48e+02 (3.0e+02)	4.55e+02 (4.0e+02)	4.38e+02 (2.5e+02)	4.31e+02 (3.1e+01)	4.15e+02 (3.8e-04)	4.30e+02 (4.0e-04)	3.64e+02 (3.0e-04)	4.53e+02 (3.7e+02)	4.30e+02 (4.3e-04)	4.34e+02 (1.4e+02)	4.30e+02 (3.7e-04)	4.07e+02 (3.7e-04)	3.60e+02 (2.7e-04)	4.01e+02 (3.6e-04)	<b>3.38e+02</b> <b>(1.0e-04)</b>

### 3. Comparative Analysis of NDE

In this section, first we have undertaken a comparative analysis of performance of NDE and its contenders considered in the comparative framework. Next we have examined the degree of robustness of all competitor algorithms against noise and problem dimension and their speed of convergence of NDE.

In Table-V, the relative performance of the contender algorithms is investigated with respect to the median and the IQR (within parenthesis) of the FEV metric values after 50 independent runs while minimizing BF1 functions in the presence of zero mean Gaussian noise (variance = 0.45). The best metric value in each instance has been set in bold.

Wilcoxon rank sum test [16] is undertaken with a significance level of 0.05 to judge the statistical level of significance of the difference of the 50 pairs of FEV metric sample values (obtained after each of the 50 runs) of any two competitive algorithms. The value in the third bracket in Table-V represents the *p*-value obtained through the rank sum test between the best algorithm (resulting the minimum median FEV) and each of the remaining algorithms for the noisy versions of BF1. Here NA covers the *not applicable* cases of comparing the best algorithm with itself. A *p*-value less than 0.05 (statistical level of significance) corresponding to the comparative analysis of performance of the best algorithms and one of its contenders establishes that their performances differ significantly.

Table-V reveals that almost equal performance accuracies are achieved by most of the contender algorithms for *f01*–*f07*. However, the remaining complex functions expose significant difference in the performance of the competitors. 22 cases out of 28 noisy versions of BF1 are noted in Table-V where NDE outperforms its contenders. Moreover, 21 instances of these 22 cases reveal statistically significant superiority of NDE over its contenders. For *f03*, the dominance of IPOP-UH-CMA-ES and MUDE by NDE are not statistically significant. The quality performance of IPOP-UH-CMA-ES (which attains the second best rank) over NDE is evident for *f09* and *f22*. Statistically equivalent results are yielded by NDE and IPOP-UH-CMA-ES for benchmarks *f01*, *f02*, *f05*, and *f08*. However, it is worth mentioning that out of these four cases, the lowest IQR is achieved by NDE for three instances, including *f02*, *f05*, and *f08*.

The statistical significance of performance of all contenders is investigated by undertaking the Friedman, the Iman-Davenport, the Friedman Aligned and the Quade non-parametric statistical tests [16] on the median values of FEV metric obtained by the contender algorithms (over 50 independent runs) in the presence of Rayleigh, exponential and Gamma noise of respective variances of 0.17, 0.77 and 0.34 in the BF1 landscapes. Table-VI summarizes the rankings obtained by the above procedures.

TABLE V-A: COMPARISON OF ALGORITHMS FOR NSOOPs WITH RESPECT TO *FEV* FOR GAUSSIAN NOISE (OF  $\sigma^2=0.45$ ) IN FITNESS LANDSCAPE FOR BF1: f01 TO f23

Functions	NDE	IPOP-UH-CMA-ES	MUDE	LAPSO	NTGA	IAAS	PSOHT	BBPSO-CJ	ODE	DE-RSF-SS	MFEGA
f01	0.00e+00 (0.00e+00) NA	0.00e+00 (0.00e+00) NA	1.63e-08 (7.01e-09) [4.21e-05]	5.63e-08 (7.10e-09) [3.91e-05]	1.60e-07 (1.14e-08) [2.31e-05]	1.90e-07 (1.36e-08) [4.46e-06]	3.50e-07 (5.98e-08) [2.58e-06]	4.07e-07 (6.17e-08) [4.13e-07]	4.83e-07 (6.81e-07) [2.48e-07]	7.06e-07 (7.88e-06) [5.58e-08]	8.98e-06 (7.95e-06) [4.63e-08]
f02	0.00e+00 (0.00e+00) NA	0.00e+00 (2.43e-04) [3.40e-02]	2.85e-05 (2.73e-01) [5.28e-02]	4.94e-05 (5.34e+01) [4.01e-03]	7.82e-04 (1.22e+02) [4.74e-04]	1.07e-03 (1.38e+02) [5.50e-06]	1.36e-03 (2.57e+03) [5.10e-06]	1.36e-03 (2.18e+04) [5.41e-07]	2.62e-02 (1.33e+05) [4.23e-07]	3.53e-02 (1.54e+05) [3.74e-08]	4.96e-02 (2.52e+05) [5.60e-08]
f03	7.92e-04 (2.08e-04) NA	3.85e-03 (3.64e-03) [2.60e-01]	6.13e-03 (1.65e+01) [3.21e-01]	8.63e-03 (1.05e+02) [6.50e-02]	4.21e-03 (1.95e+02) [4.50e-02]	5.93e-02 (2.38e+02) [4.45e-02]	8.16e-02 (2.60e+02) [6.93e-03]	1.70e-01 (2.92e+02) [6.00e-03]	1.15e-01 (3.43e+02) [7.03e-04]	3.91e-01 (4.20e+02) [3.65e-04]	4.85e+01 (7.16e+02) [7.12e-04]
f04	0.00e+00 (9.11e-09) NA	7.40e-05 (3.36e-05) [3.32e-03]	1.30e-04 (3.56e-04) [2.96e-03]	1.03e-03 (5.54e-04) [2.57e-03]	4.36e-03 (8.67e-04) [3.25e-04]	6.21e-03 (1.04e-03) [2.97e-04]	7.23e-03 (2.38e-01) [2.83e-05]	1.37e-02 (2.64e+00) [2.61e-05]	2.45e-02 (5.27e+01) [2.32e-06]	3.42e-02 (1.54e+02) [3.17e-07]	4.16e-02 (3.85e+02) [3.30e-07]
f05	0.00e+00 (2.73e-04) NA	0.00e+00 (1.12e-05) [6.26e-05]	2.00e-08 (1.25e-03) [3.10e-05]	9.50e-07 (2.88e-03) [5.32e-07]	2.50e-06 (4.67e-03) [4.08e-07]	1.31e-06 (3.10e-03) [4.82e-07]	4.10e-05 (5.45e-03) [3.82e-07]	7.50e-05 (5.75e-03) [6.12e-08]	1.00e-04 (6.53e-03) [5.77e-08]	1.21e-04 (8.40e-03) [4.03e-08]	1.28e-04 (8.57e-03) [2.85e-08]
f06	4.94e-01 (0.00e+00) NA	2.54e+01 (2.53e-15) [2.18e-04]	3.01e+01 (1.00e-01) [4.47e-05]	4.05e+01 (6.00e+00) [2.40e-06]	4.08e+01 (6.56e+00) [5.84e-07]	3.78e+01 (2.72e+00) [3.67e-06]	4.23e+01 (7.56e+00) [6.02e-09]	4.28e+01 (9.37e+00) [3.84e-10]	4.32e+01 (1.41e+01) [5.26e-11]	4.34e+01 (3.65e+01) [8.14e-11]	4.33e+01 (2.22e+01) [3.64e-11]
f07	1.40e+00 (1.70e+00) NA	2.88e+01 (5.57e+00) [4.24e-03]	3.57e+01 (5.91e+00) [3.01e-05]	3.56e+01 (8.86e+00) [4.13e-05]	4.95e+01 (9.20e+00) [3.68e-08]	3.90e+01 (9.18e+00) [1.55e-06]	5.31e+01 (1.07e+01) [3.13e-09]	7.94e+01 (1.53e+01) [1.82e-12]	5.66e+01 (1.32e+01) [2.77e-11]	1.00e+02 (1.94e+01) [1.14e-13]	4.18e+02 (1.55e+01) [1.02e-13]
f08	2.08e+01 (2.83e-02) NA	2.08e+01 (3.55e-02) [4.28e-02]	2.10e+01 (4.10e-02) [4.10e-02]	2.10e+01 (4.36e-02) [3.04e-02]	2.10e+01 (4.67e-02) [5.43e-03]	2.10e+01 (4.57e-02) [5.34e-03]	2.10e+01 (6.10e-02) [5.13e-03]	2.10e+01 (6.21e-02) [5.70e-04]	2.11e+01 (1.38e-01) [4.88e-04]	2.11e+01 (2.37e-01) [6.58e-05]	2.11e+01 (1.42e+01) [6.80e-05]
f09	1.78e+01 (1.94e+00) [6.31e-02]	3.18e+00 (1.23e+00) NA	4.11e+01 (2.50e+00) [6.10e-02]	3.37e+01 (2.25e+00) [6.17e-02]	4.73e+01 (2.58e+00) [5.75e-02]	5.04e+01 (3.52e+00) [5.04e-02]	4.97e+01 (2.90e+00) [5.20e-02]	5.18e+01 (4.02e+00) [5.37e-03]	5.66e+01 (4.81e+00) [4.74e-03]	5.75e+01 (4.81e+00) [4.65e-03]	5.98e+01 (9.52e+00) [4.51e-03]
f10	0.00e+00 (0.00e+00) NA	6.33e-07 (4.43e-05) [5.28e-04]	1.55e-03 (1.60e-03) [4.10e-07]	8.70e-05 (7.72e-04) [4.68e-06]	6.27e-03 (4.38e-03) [3.96e-09]	2.28e-02 (1.32e-02) [2.37e-09]	1.46e-02 (6.16e-03) [3.12e-09]	3.48e-02 (1.83e-02) [3.86e-11]	5.31e-02 (3.10e-02) [4.08e-15]	7.01e-02 (3.84e-02) [3.80e-15]	1.26e-01 (6.50e-02) [1.60e-18]
f11	2.76e-01 (0.00e+00) NA	2.74e+00 (4.20e-03) [6.31e-03]	5.03e+00 (4.91e-02) [3.92e-05]	8.16e+00 (3.40e-01) [2.20e-07]	3.45e+00 (2.17e-02) [3.70e-04]	1.27e+01 (1.64e+00) [4.37e-08]	1.55e+01 (2.28e+00) [3.90e-08]	2.08e+01 (3.04e+00) [2.80e-08]	2.61e+01 (6.15e+00) [1.30e-08]	3.97e+01 (1.11e+01) [2.84e-10]	4.83e+01 (1.45e+01) [1.17e-10]
f12	4.08e+00 (1.34e+00) NA	2.56e+01 (2.20e+00) [3.52e-04]	3.15e+01 (7.18e+00) [3.37e-04]	6.45e+01 (8.42e+00) [3.08e-05]	8.54e+01 (1.24e+01) [2.76e-07]	8.94e+01 (2.15e+01) [3.36e-09]	1.04e+02 (2.35e+01) [2.96e-09]	1.66e+02 (3.30e+01) [2.92e-15]	1.53e+02 (2.83e+01) [2.78e-09]	1.74e+02 (4.21e+01) [3.50e-17]	2.15e+02 (4.62e+01) [2.63e-17]
f13	1.51e+01 (2.10e+00) NA	7.34e+01 (4.62e+00) [2.00e-05]	8.48e+01 (9.48e+00) [3.34e-06]	1.06e+02 (1.90e+01) [2.16e-06]	1.46e+02 (2.00e+01) [3.28e-07]	1.66e+02 (3.03e+01) [2.85e-07]	1.92e+02 (3.30e+01) [2.60e-07]	2.25e+02 (3.40e+01) [2.87e-09]	3.01e+02 (7.84e+01) [3.34e-13]	2.60e+02 (5.48e+01) [2.10e-12]	3.90e+02 (8.30e+01) [3.45e-16]
f14	3.63e-02 (1.98e-02) NA	1.76e+01 (8.01e+00) [5.43e-03]	2.10e+01 (9.62e+00) [2.11e-07]	5.41e+01 (3.26e+01) [1.56e-10]	3.84e+02 (1.20e+02) [4.87e-11]	6.36e+02 (2.22e+02) [2.37e-11]	7.75e+02 (2.97e+02) [3.57e-12]	8.36e+02 (4.67e+02) [3.17e-13]	1.38e+03 (6.11e+02) [3.03e-15]	1.00e+03 (5.52e+02) [4.47e-15]	1.70e+03 (7.18e+02) [1.22e-20]
f15	6.62e+02 (2.82e+02) NA	1.35e+03 (2.88e+02) [8.82e-03]	5.57e+03 (4.45e+02) [4.12e-04]	1.97e+03 (4.03e+02) [7.12e-03]	6.26e+03 (4.71e+02) [6.62e-05]	6.37e+03 (5.45e+02) [6.75e-06]	7.05e+03 (5.54e+02) [4.86e-06]	7.43e+03 (6.71e+02) [8.34e-10]	8.25e+03 (6.80e+02) [5.14e-16]	8.57e+03 (9.18e+02) [8.08e-17]	9.07e+03 (2.50e+02) [1.20e-19]
f16	5.46e-02 (2.44e-02) NA	7.80e-01 (4.21e-02) [4.80e-03]	8.27e-01 (4.38e-02) [5.21e-04]	1.45e+00 (1.93e-01) [4.95e-04]	1.60e+00 (2.00e-01) [4.83e-04]	1.68e+00 (2.05e-01) [5.06e-05]	1.80e+00 (2.74e-01) [5.01e-05]	2.13e+00 (3.15e-01) [5.17e-09]	2.35e+00 (4.45e-01) [5.04e-10]	2.55e+00 (6.40e-01) [4.90e-10]	2.77e+00 (8.25e-01) [5.26e-13]
f17	3.95e+01 (4.53e-04) NA	5.06e+01 (1.12e-02) [1.42e-06]	5.06e+01 (3.47e-02) [1.97e-07]	5.26e+01 (5.43e-01) [5.26e-08]	5.60e+01 (1.60e+00) [1.15e-08]	5.72e+01 (1.77e+00) [5.27e-15]	6.90e+01 (2.67e+00) [1.20e-18]	7.57e+01 (5.10e+00) [5.33e-19]	8.16e+01 (8.87e+00) [3.38e-20]	9.35e+01 (1.11e+01) [2.40e-20]	1.02e+02 (3.38e+01) [1.10e-20]
f18	5.98e+01 (8.00e+00) NA	1.08e+02 (1.17e+01) [3.06e-06]	1.24e+02 (1.46e+01) [2.92e-06]	1.27e+02 (2.23e+01) [3.00e-07]	1.33e+02 (2.88e+01) [2.42e-08]	1.51e+02 (3.58e+01) [2.97e-09]	1.62e+02 (3.72e+01) [2.58e-09]	2.06e+02 (5.01e+01) [2.50e-10]	1.80e+02 (3.84e+01) [2.50e-10]	2.26e+02 (6.08e+01) [2.95e-10]	3.93e+02 (8.10e+01) [2.30e-10]
f19	2.42e+00 (3.07e-01) NA	1.95e+00 (2.68e-01) [5.01e-04]	3.30e+00 (4.03e-01) [2.67e-04]	3.40e+00 (4.70e-01) [2.97e-06]	3.47e+00 (5.02e-01) [1.70e-12]	3.88e+00 (6.97e-01) [1.54e-15]	3.93e+00 (7.42e-01) [2.80e-16]	4.86e+00 (9.14e-01) [4.90e-17]	5.14e+00 (9.62e-01) [2.15e-18]	5.27e+00 (1.27e+00) [4.73e-19]	5.71e+00 (1.93e+00) [1.20e-20]
f20	1.30e+01 (3.81e-01) NA	1.86e+01 (3.92e-01) [4.15e-05]	1.91e+01 (5.02e-01) [2.40e-05]	1.91e+01 (5.60e-01) [3.86e-06]	1.92e+01 (6.34e-01) [2.90e-06]	1.95e+01 (6.86e-01) [2.08e-06]	2.04e+01 (7.88e-01) [1.94e-06]	2.07e+01 (8.64e-01) [4.37e-07]	2.25e+01 (1.20e+00) [2.77e-07]	2.20e+01 (9.62e-01) [2.51e-07]	2.43e+01 (5.75e+00) [1.28e-07]
f21	1.98e+02 (3.77e+00) NA	2.37e+02 (2.02e+01) [6.90e-04]	2.40e+02 (1.51e+02) [1.52e-04]	2.52e+02 (2.12e+02) [5.71e-05]	3.02e+02 (2.41e+02) [4.87e-06]	4.12e+02 (2.74e+02) [1.20e-09]	5.54e+02 (3.23e+02) [4.30e-11]	6.63e+02 (3.52e+02) [5.24e-15]	8.44e+02 (4.33e+02) [1.66e-16]	7.76e+02 (3.93e+02) [2.28e-16]	8.52e+02 (5.25e+02) [1.01e-20]
f22	3.90e+01 (8.53e+00) [3.10e-03]	2.01e+01 (1.76e+01) NA	4.28e+01 (2.80e+01) [1.50e-03]	1.81e+02 (4.91e+01) [1.40e-04]	2.88e+02 (2.43e+02) [3.04e-07]	4.06e+02 (2.92e+02) [2.98e-07]	9.16e+02 (3.80e+02) [2.47e-10]	5.85e+02 (3.50e+02) [1.96e-08]	1.98e+03 (4.66e+02) [1.70e-12]	2.46e+03 (5.77e+02) [1.45e-12]	2.68e+03 (6.36e+02) [1.33e-12]
f23	6.50e+02 (3.25e+02) NA	9.75e+02 (3.30e+02) [4.34e-07]	2.42e+03 (5.03e+02) [3.01e-08]	4.43e+03 (5.38e+02) [3.50e-09]	7.15e+03 (6.10e+02) [1.47e-10]	8.01e+03 (6.83e+02) [3.71e-14]	8.45e+03 (8.08e+02) [4.12e-16]	8.98e+03 (9.40e+02) [1.48e-16]	9.35e+03 (9.88e+02) [3.83e-18]	9.76e+03 (1.43e+03) [3.66e-18]	9.81e+03 (2.38e+03) [1.46e-18]

TABLE V-B: COMPARISON OF ALGORITHMS FOR NSOOPs WITH RESPECT TO *FEV* FOR GAUSSIAN NOISE (OF  $\sigma^2=0.45$ ) IN FITNESS LANDSCAPE FOR BF1: f24 TO f28

Functions	NDE	IPOP-UH-CMA-ES	MUDE	LAPSO	NTGA	IAAS	PSOHT	BBPSO-CJ	ODE	DE-RSF-SS	MFEGA
f24	<b>2.03e+02</b> <b>(1.46e+00)</b> NA	2.24e+02 (5.13e+00) [7.90e-03]	2.33e+02 (8.44e+00) [1.87e-03]	2.40e+02 (1.04e+01) [4.32e-05]	2.97e+02 (1.43e+01) [2.64e-05]	2.71e+02 (1.23e+01) [7.37e-07]	3.07e+02 (2.00e+01) [3.32e-10]	3.07e+02 (1.62e+01) [6.93e-11]	3.10e+02 (2.05e+01) [2.91e-12]	3.30e+02 (2.22e+01) [7.07e-13]	3.85e+02 (4.68e+01) [1.17e-15]
f25	<b>2.01e+02</b> <b>(3.77e+00)</b> NA	2.75e+02 (5.44e+00) [6.81e-05]	3.15e+02 (1.08e+01) [6.70e-06]	2.92e+02 (8.80e+00) [4.92e-05]	3.23e+02 (1.20e+01) [4.23e-11]	3.36e+02 (1.33e+01) [5.13e-12]	3.58e+02 (1.68e+01) [8.51e-13]	3.63e+02 (1.88e+01) [6.68e-13]	3.64e+02 (1.25e+02) [4.90e-14]	3.65e+02 (2.50e+02) [7.96e-17]	3.73e+02 (2.63e+02) [9.35e-17]
f26	<b>1.67e+02</b> <b>(3.78e-01)</b> NA	2.00e+02 (3.92e+00) [7.41e-03]	2.00e+00 (2.01e+01) [4.25e-05]	2.24e+02 (2.71e+01) [4.68e-06]	2.43e+02 (5.02e+01) [3.80e-07]	2.52e+02 (6.18e+01) [1.60e-08]	2.90e+02 (9.72e+01) [2.94e-10]	2.74e+02 (9.05e+01) [4.64e-10]	2.90e+02 (1.07e+02) [5.10e-11]	3.61e+02 (1.36e+02) [6.97e-13]	3.23e+02 (1.12e+02) [1.11e-14]
f27	<b>3.11e+02</b> <b>(3.11e+01)</b> NA	6.40e+02 (4.78e+01) [6.04e-04]	7.70e+02 (1.16e+02) [5.84e-06]	1.00e+03 (1.44e+02) [5.06e-08]	9.17e+02 (1.36e+02) [1.66e-07]	1.06e+03 (1.47e+02) [4.07e-09]	1.20e+03 (1.80e+02) [5.61e-10]	1.44e+03 (2.00e+02) [1.72e-10]	1.55e+03 (2.82e+02) [2.64e-13]	2.11e+03 (6.23e+02) [1.48e-13]	1.63e+03 (4.02e+02) [1.24e-16]
f28	<b>3.30e+02</b> <b>(2.56e-05)</b> NA	4.00e+02 (3.95e-04) [7.16e-05]	4.28e+02 (3.74e-01) [2.62e-05]	4.31e+02 (4.57e+01) [6.92e-07]	4.40e+02 (2.91e+02) [3.22e-09]	4.58e+02 (4.20e+02) [3.23e-13]	4.57e+02 (4.16e+02) [4.72e-10]	6.97e+02 (4.43e+02) [5.55e-14]	8.48e+02 (1.12e+03) [7.50e-18]	9.36e+02 (1.22e+03) [3.93e-18]	1.12e+03 (1.36e+03) [7.75e-18]

TABLE VI: RANKS ACHIEVED BY STATISTICAL TESTS WITH SIGNIFICANCE LEVEL OF 0.05

Statistical Tests		Friedman	Iman-Davenport	Friedman Aligned	Quade
Type of Noise in Fitness Landscape		Rayleigh ( $\sigma^2=0.17$ )	Rayleigh ( $\sigma^2=0.17$ )	Exponential ( $\sigma^2=0.77$ )	Gamma ( $\sigma^2=0.34$ )
Algorithms	NDE	1.21251	1.21251	61.6768	1.1115
	IPOP-UH-CMA-ES	1.89109	1.89109	65.3554	1.9391
	MUDE	2.85537	2.85537	71.4982	2.8603
	LAPSO	3.89109	3.89109	96.0339	3.9391
	NTGA	4.92680	4.92680	119.2128	4.8800
	IAAS	6.33394	6.33394	171.3908	5.9588
	PSOHT	6.81966	6.81966	187.7128	6.9218
	BBPSO-CJ	7.99823	7.99823	223.0698	7.9834
	ODE	8.81251	8.81251	227.6408	8.9470
	DE-RSF-SS	10.5697	10.5697	234.8558	10.6278
	MFEGA	10.6891	10.6891	240.2478	10.8312
Degrees of Freedom		10	10, 270	10	10, 270
Statistics		269.106	867.107	25.2538	73.6952
<i>p</i> -value		1.37e-10	2.85e-199	4.81e-03	2.10e-74

It is evident from Table-VI that *p*-values in all three cases are less than the significance level 0.05 indicating significant differences in the metric values obtained by the competitor algorithms. The results reported in Table-V also designate NDE as the best algorithm. So eight post-hoc analysis tests are employed [16], including the Bonferroni, the Holm, the Hochberg, the Hommel, the Holland, the Rom, the Finner and the Li tests, over the results of Table-V with NDE as the control algorithm. Fig. 6 pictorially presents the outcome of the Bonferroni-Dunn test undertaken with the Friedman ranks as reported in Table-VI. The aim of the Bonferroni-Dunn test is to quantitatively evaluate the statistical significance level of superiority of the control algorithm NDE over each of the remaining algorithms. For the Bonferroni-Dunn test in the present context, a critical difference is computed which appears as 1.6811. If the difference of Friedman ranks of any two algorithms exceeds the critical difference measure, their performances are considered to be significantly different. It can be noted from Fig. 6 that all the contenders are overshadowed by NDE in a statistically significant fashion except for IPOP-UH-CMA-ES and MUDE with level of significance 0.05.

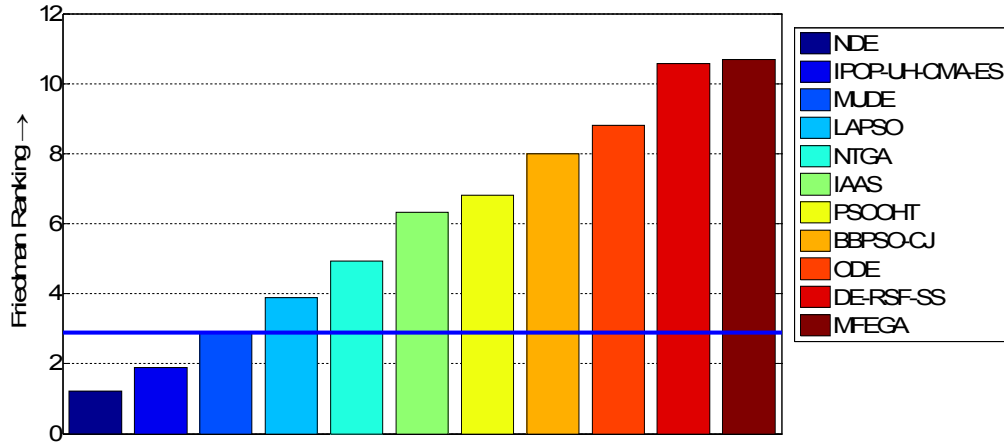
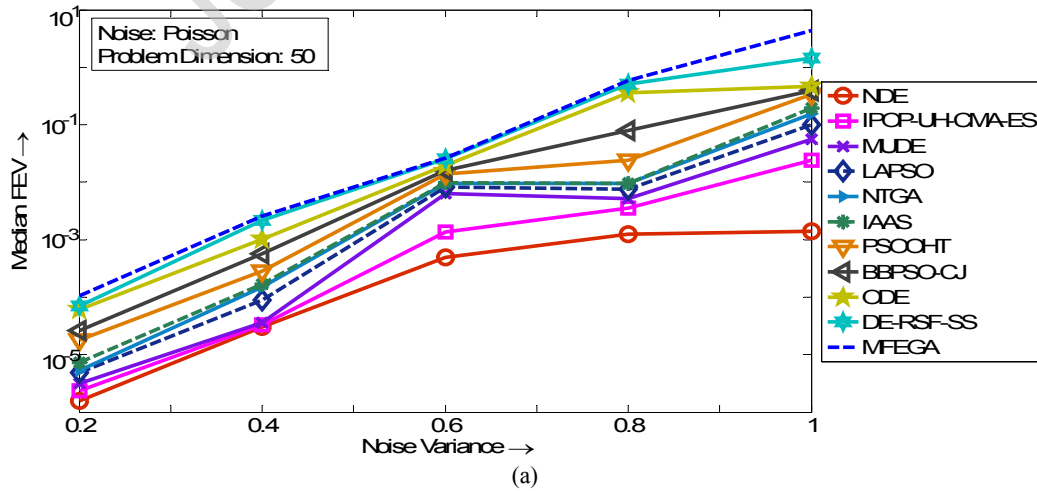


Fig. 6 Bonferroni-Dunn test with NDE as control algorithm

Table-VII reports the adjusted  $p$ -values for the Friedman test obtained with these eight statistical methods for multiple comparisons. The null hypothesis here considers that the performance of an algorithm is equally good to the performance of the control algorithm, i.e., NDE. The performance of NDE and its contender is inferred to be significantly different if the null hypothesis is rejected on obtaining an adjusted  $p$ -value below 0.05. The comment on accepting (A) or rejecting (R) the null hypothesis is presented in the parenthesis.

TABLE VII: ADJUSTED  $P$ -VALUES FOR FRIEDMAN TEST WITH NDE AS CONTROL ALGORITHM FOR RAYLEIGH NOISE (OF  $\sigma^2=0.07$ ) IN BF1 FITNESS LANDSCAPE

Algorithms	z	Unadjusted $p$	Bonferroni	Holm	Hochberg	Hommel	Holland	Rom	Finner	Li
MFEGA	10.797	3.515e-27	3.515e-26 (R)	3.515e-26 (R)	3.515e-26 (R)	3.515e-26 (R)	0.000000 (R)	3.342e-26 (R)	0.000000 (R)	6.323e-27 (R)
DE-RSF-SS	9.9921	1.647e-23	1.647e-22 (R)	1.482e-22 (R)	1.482e-22 (R)	1.482e-22 (R)	0.000000 (R)	1.410e-22 (R)	0.000000 (R)	2.964e-23 (R)
ODE	8.4610	2.646e-17	2.646e-16 (R)	2.117e-16 (R)	2.117e-16 (R)	2.117e-16 (R)	0.000000 (R)	2.012e-16 (R)	0.000000 (R)	4.760e-17 (R)
BBPSO-CJ	7.6552	1.927e-14	1.927e-13 (R)	1.348e-13 (R)	1.348e-13 (R)	1.348e-13 (R)	1.351e-13 (R)	1.282e-13 (R)	4.828e-14 (R)	3.466e-14 (R)
PSOCHT	6.3256	2.520e-10	2.520e-09 (R)	1.511e-09 (R)	1.511e-09 (R)	1.511e-09 (R)	1.511e-09 (R)	1.437e-09 (R)	5.040e-10 (R)	4.532e-10 (R)
IAAS	5.4392	5.347e-08	5.347e-07 (R)	2.673e-07 (R)	2.673e-07 (R)	2.673e-07 (R)	2.673e-07 (R)	2.542e-07 (R)	8.913e-08 (R)	9.618e-08 (R)
NTGA	4.1901	2.785e-05	2.785e-04 (R)	1.113e-04 (R)	1.113e-04 (R)	1.113e-04 (R)	1.113e-04 (R)	1.061e-04 (R)	3.980e-05 (R)	5.010e-05 (R)
LAPSO	3.0217	2.511e-03	2.511e-02 (R)	7.536e-03 (R)	7.536e-03 (R)	7.536e-03 (R)	7.517e-03 (R)	7.536e-03 (R)	3.138e-03 (R)	4.497e-03 (R)
MUDE	1.8532	6.381e-02	6.381e-01 (A)	1.275e-01 (A)	1.275e-01 (A)	1.275e-01 (A)	1.234e-01 (A)	1.275e-01 (A)	7.065e-02 (A)	1.028e-01 (A)
IPOP-UH-CMA-ES	0.7654	4.438e-01	1.000e+00 (A)	4.438e-01 (A)	4.438e-01 (A)	4.438e-01 (A)	4.438e-01 (A)	4.438e-01 (A)	4.438e-01 (A)	4.438e-01 (A)



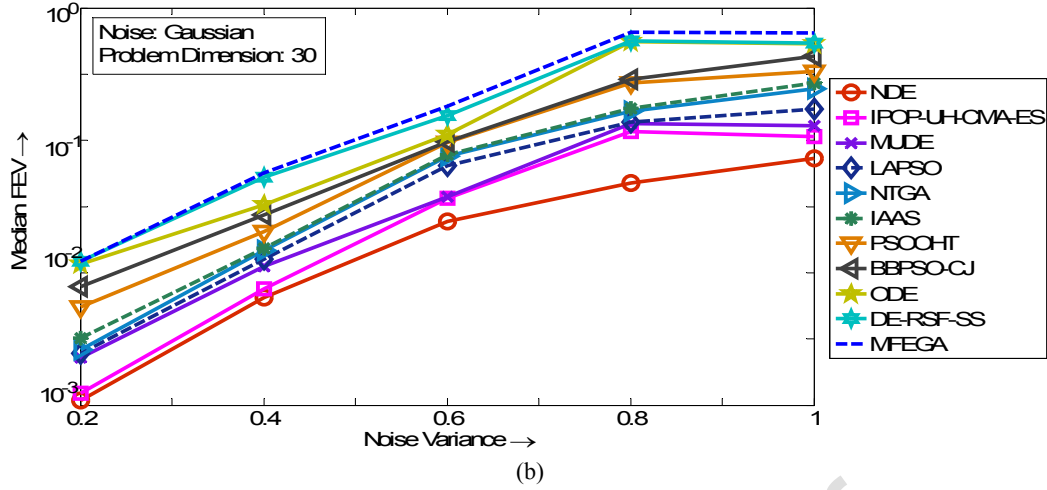


Fig. 7 Plot of median FEV metric value with the noise variance for (a) f03 and (b) f10 of BF1 set

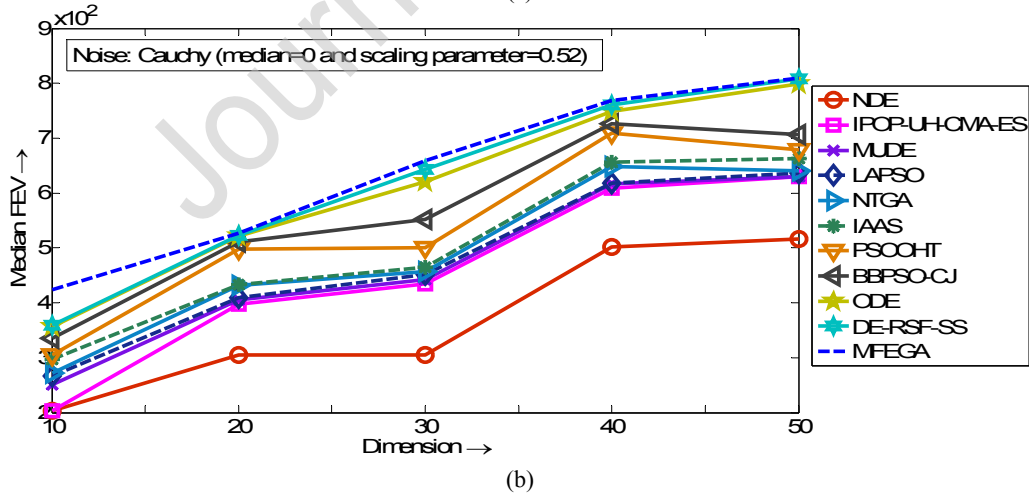
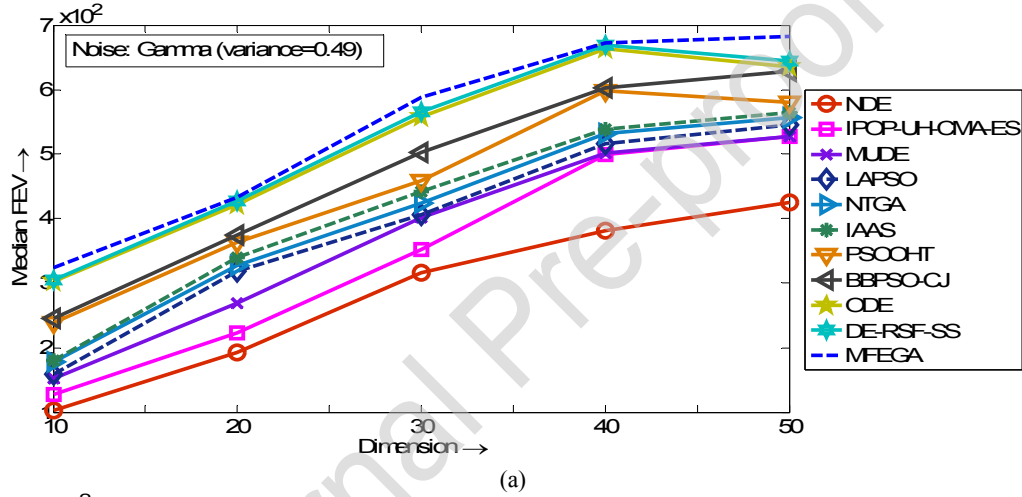


Fig. 8 Plot of median FEV metric values with problem dimension for (a) f17 and (b) f24 of BF1 set

The next experiments are concerned with robustness analysis of NDE and its competitors with respect to noise-variance and problem dimension. Fig. 7 shows the evolution of the median FEV metric value with the variance of noise samples drawn from Poisson and Gaussian distribution respectively. Fig. 7 reveals deterioration of the performance of each algorithm (in achieving the lower FEV metric values)

with increase in level of infiltration of noise in the fitness landscape. However, NDE appears to be the most robust algorithm against the prevalent presence of noise, even in the complex fitness landscape.

The plot of the same metric values against problem dimension (within [10, 50]) is presented in Fig. 8 for Gamma and Cauchy noise contaminating the objective surface respectively. It is evident from Fig. 8 that for specific settings of noise, an increase in problem dimension also increases FEV reflecting the gradual loss of effectiveness of all the contender algorithms to precisely locate the global optimum in a complex terrain. However, Fig. 8 substantiates that NDE still remains the conqueror among all by attaining the least FEV metric values. The next experiment deals with comparative analysis of convergence speed of all contender algorithm in terms of the FEV metric over FEs. The results are pictorially presented in Fig. 9 for different settings of search space dimensions and noise distribution in the fitness landscape. It reveals that NDE outperforms all the contender algorithms after  $10^4 \times D$  FEs.

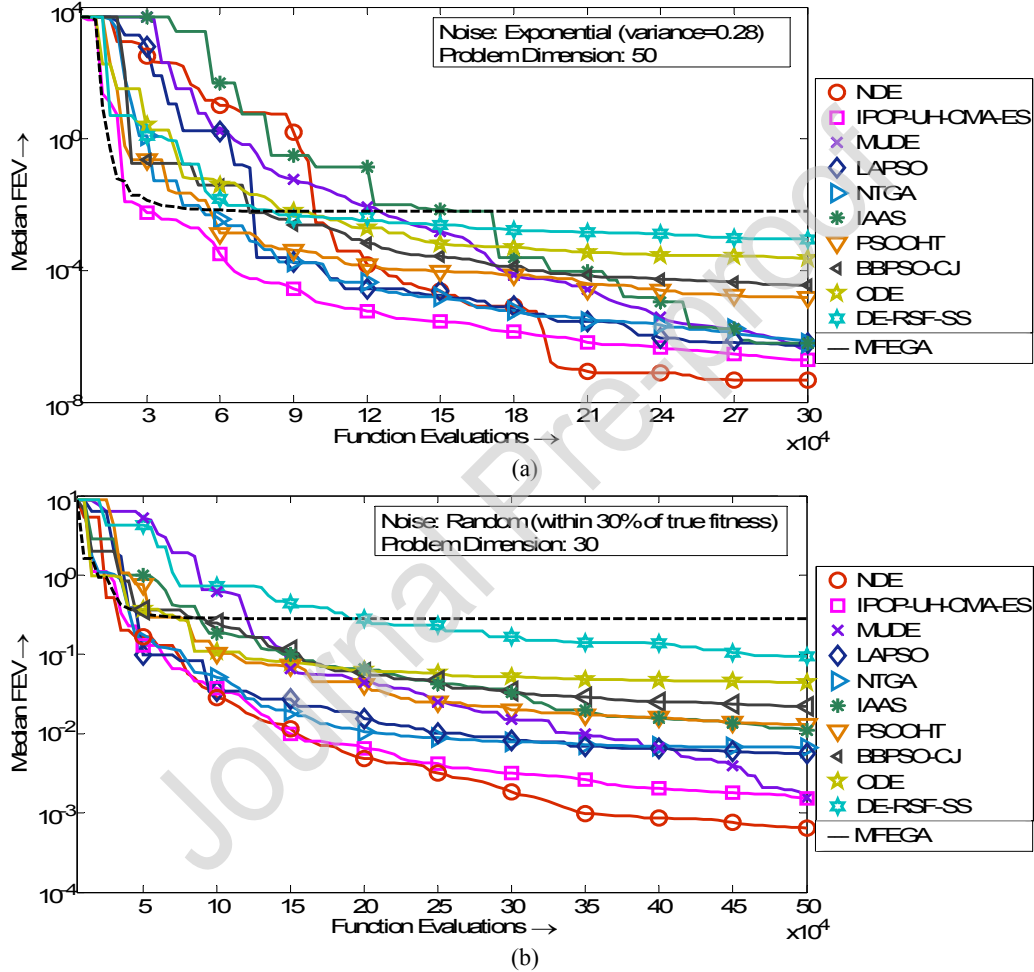
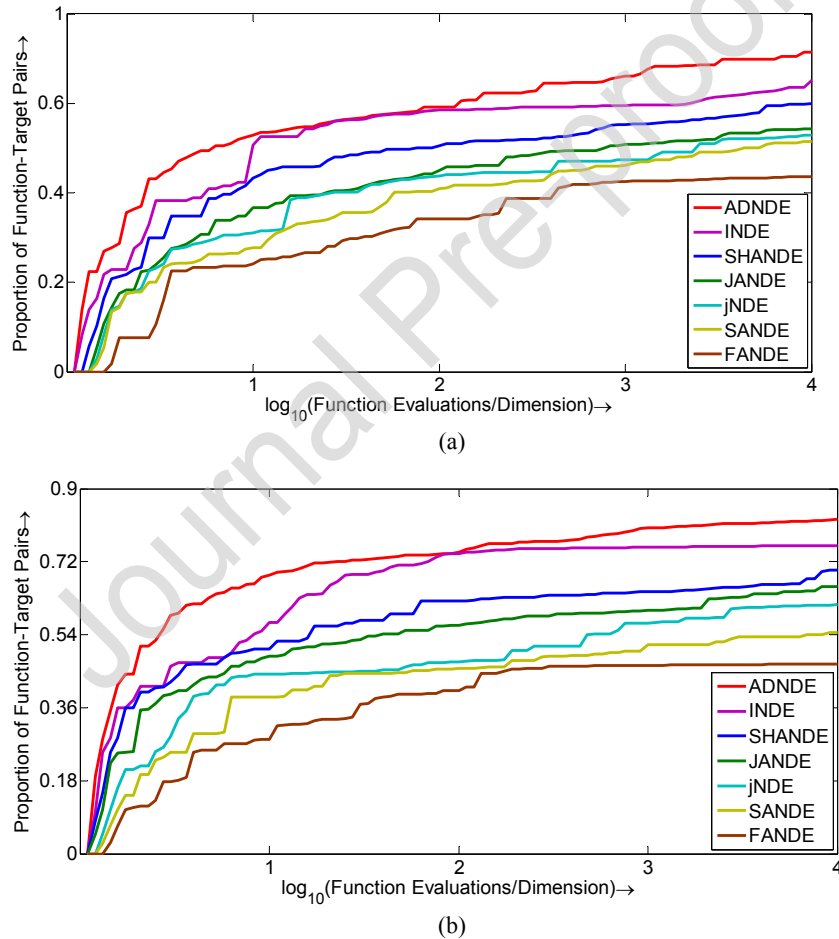


Fig. 9 Plot of average FEV metric values with FEs for (a) f04 and (b) f07 of BF1 set

#### 4. Comparative Analysis of Parameter Adaptive DE Variants

The latter part of the experiment attempts to improve the performance of the most popular variants of DE, including adaptive distributed DE (ADDE) [72], DE with individual dependent mechanism (IDE) [69], success history based parameter adaptation for DE (SHADE) [68], adaptive DE with optional external archive (JADE) [74], DE with self-adapting control parameters (jDE) [12], self-adaptive DE (SaDE) [46], and fuzzy adaptive DE (FADE) [31], by incorporating the proposed noise handling strategies in the algorithms. Fig. 10 provides a comparative estimate of the relative performance of their noisy counterparts, called JANDE, jNDE, SANDE and FANDE, with respect to the empirical cumulative distribution function (ECDF) of the expected run-time (ERT) for all BF2 benchmarks. The ECDF of the ERT is constructed as a bootstrap distribution of ERT (the number of FEs divided by the problem dimension  $D$ ) for 100 function-target pairs in the range  $[10^{-8}, 10^2]$  for each BF2 optimization problem. Since the ECDF graphs reveal the proportion of solved problems, instead of the fitness function values obtained by the algorithms, the ECDF graphs for all 30 functions of the BF2 framework are meaningfully aggregated into one graph for a fixed problem dimension  $D$ . It is evident from Fig. 10 that ADNDE outperforms its competitors with respect to the ECDF of ERT.



**Fig. 10** Comparative analysis of improved self-adaptive DE variants for noisy optimization based on ECDF of their bootstrapped distribution of the number of objective function evaluations divided by dimension for 100 function-targets in  $10^{-8}$  to  $10^2$  for all functions of BF2 framework in (a) 10- $D$  and (b) 20- $D$

#### V. CONCLUSION

The paper introduced a novel approach to handle uncertainty in selecting quality solutions in differential evolution (DE) algorithm in the presence of noise in the objective surface. This is realized

by incorporating stochastic learning automata (SLA) and the niching based selection strategy in the traditional DE. The sample size used for periodic fitness evaluation of a solution is adapted based on the fitness landscape in its local neighborhood using the reinforcement learning policy of the SLA. The strategy assigns a large sample size to solutions in a largely noise-affected area for accuracy in fitness estimate while significantly reducing the computational complexity associated with the unnecessary re-evaluation of less noisy solutions. The niching strategy ensures population quality and diversity as well, which in turn helps to avoid the deceptive effect of noise during selection.

The paper additionally addresses two other significant issues of noisy optimization, including estimation of the effective fitness of a trial solution and enhancing the search dynamics of DE in the presence of noise in the fitness landscape. The fitness estimate here is computed by taking weighted average of all fitness samples of the given solution. The weight assigned to a fitness sample varies inversely with its difference from the median fitness of the all remaining samples. It thus reduces the significance of noisy rare fitness samples towards evaluation of effective fitness of the given solution. Lastly, the traditional DE/current-to-best/1 mutation strategy is ameliorated with an aim of selecting the concerned members (participating in the difference term of the mutation dynamic) from relatively less noisy regions of the fitness landscape. It in turn prohibits the illusive effect of noise to misguide the search direction.

First experiment is undertaken to compare the efficiency of the proposed individual extensions and their combinations to overcome the jeopardizing effect of noise with respect to function error value (FEV) metric. The comparative analysis indicates that the proposed noisy differential evolution (NDE) significantly supersedes other noisy DE variants. Two sets of benchmark functions are used to arrive at this conclusion. They include i) a test suit of 28 CEC'2013 benchmark functions, each contaminated with noise samples of seven stochastic distributions and ii) a test bed of 30 GECCO'2010 noisy benchmark functions.

The proposed NDE algorithm is also compared with ten state-of-the-art contender algorithms with respect to the performance metrics and same set of noisy benchmark functions. Statistical significance of the results is inferred using four nonparametric statistical tests, and eight post hoc analyses. Experimental results reveal statistically significant superiority of NDE over its competitor algorithms. NDE also overshadows its contenders both in robustness (against noise and search space dimension) and convergence speed.

There remains ample scope for future researches in this new domain of research. The first open problem is of great importance to enhance the performance of the noisy multi-objective optimization (MOO) algorithm using the proposed noise-handling strategies. The challenge lies in effective quantization of states and selection of reward function for reinforcement learning based sampling policy to solve noisy multi-modal MOO problems. The second open research may address the parallel evolution of multiple population of DE in the complex search space to avoid dismissal promising locations in the presence of noise. The future research may exploit the scope of integrating transfer learning approaches into an EA to utilize past experience to significantly enhance performance and robustness for developing better noisy MOO algorithms. Finally, the noisy optimization algorithms should be devised to solve real-world engineering problems including robotics, material processing, control and guidance, and many others.

## REFERENCES

- [1] Ahrens J. H., and U. Dieter, "Generating gamma variates by a modified rejection technique," *Communications of the ACM*, vol. 25, no. 1, 1982, pp. 47-54.
- [2] Aizawa A. N., and B. W. Wah, "Dynamic control of genetic algorithms in a noisy environment," In *Genetic Algorithms*, pp. 48-55.
- [3] Aizawa A. N., and B. W. Wah, "Scheduling of genetic algorithms in a noisy environment," *Evolutionary Computation*, vol. 2, no. 2, 1994, pp. 97-122.
- [4] Arnold D. V., and H. G. Beyer, "On the benefits of populations for noisy optimization," *Evolutionary Computation*, vol. 11, no. 2, 2003, pp. 111-127.
- [5] Awad N. H., M. Z. Ali, and P. N. Suganthan, "Ensemble of parameters in a sinusoidal differential evolution with niching-based population reduction," *Swarm and Evolutionary Computation*, vol. 39, 2018, pp. 141-156.
- [6] B. L. Miller, and D. E. Goldberg, "Genetic algorithms, selection schemes, and the varying effects of noise," *Evolutionary Computation*, vol. 4, no. 2, 1996, pp. 113-131.
- [7] Beilstein T., and S. Markon, "Threshold selection, hypothesis tests, and DOE methods," in *IEEE Congress on Evolutionary Computation*, 2002, pp. 777-782.



- [8] Bolte J., *Linear congruential generators*, Wolfram Demonstrations Project.
- [9] Box G. E. P., and M. E. Muller, "A note on the generation of random normal deviates," *The annals of mathematical statistics*, vol. 29, 1958, pp. 610-611.
- [10] Branke J., and C. Schmidt, "Selection in the presence of noise," In *Genetic and Evolutionary Computation—GECCO*, Springer Berlin Heidelberg, 2003, pp. 766-777.
- [11] Branke J., and C. Schmidt, "Sequential sampling in noisy environments," in *Parallel Problem Solving from Nature*, Springer Berlin Heidelberg, 2004, pp. 202-211.
- [12] Brest J., S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, 2006, pp. 646-657.
- [13] Chakraborty U.K., *Advances in differential evolution*, Springer, Heidelberg, New York, 2008.
- [14] Das S., A. Konar, and U. K. Chakraborty, "Improved differential evolution algorithms for handling noisy optimization problems," in *IEEE Congress on Evolutionary Computation*, vol. 2, 2005, pp. 1691-1698.
- [15] Das S., and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, 2011, pp. 4-31.
- [16] Derrac J., 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 and Evolutionary Computation*, vol. 1, no. 1, 2011, pp. 3-18.
- [17] Finck S., N. Hansen, R. Ros, and A. Auger, *Real-Parameter Black-Box Optimization Benchmarking 2010: Presentation of the Noisy Functions*, Working Paper 2009/21, compiled December 4, 2014.
- [18] Fitzpatrick J. M., and J. J. Grefenstette, "Genetic algorithms in noisy environments," *Machine learning*, vol. 3, no. 2-3, 1988, pp. 101-120.
- [19] Goldberg D. E., K. Deb, and J. H. Clark, "Genetic algorithms, noise, and the sizing of populations," *Complex Systems*, vol. 6, 1992, pp. 333-362.
- [20] Hansen N., A. S. P. Niederberger, L. Guzzella, and P. Koumoutsakos, "A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, 2009, pp. 180-197.
- [21] Hansen N., and A. Auger, "Principled design of continuous stochastic search: From theory to practice," in *Theory and Principled Methods for the Design of Metaheuristics*, Springer Berlin Heidelberg, 2014, pp. 145-180.
- [22] Hörmann W., J. Leydold, and G. Derflinger, "General principles in random variate generation," in *Automatic Nonuniform Random Variate Generation*, Springer Berlin Heidelberg, 2004, pp. 13-41.
- [23] Jang H., R. Choe, and K. R. Ryu, "Deriving a robust policy for container stacking using a noise-tolerant genetic algorithm," in *ACM Research in Applied Computation Symposium*, 2012, pp. 31-36.
- [24] Jin Y., and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, 2005, pp. 303-317.
- [25] Kita H., and Y. Sano, "Genetic algorithms for optimization of uncertain functions and their applications," In *IEEE SICE 2003 Annual Conference*, vol. 3, 2003, pp. 2744-2749.
- [26] Knuth D. E., *Seminumerical Algorithms, The art of computer programming*, vol. 2, 1981.
- [27] Lakshmivarahan S., and M. A. L. Thathachar, "Absolutely expedient learning algorithms for stochastic automata," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 3, 1973, pp. 281-286.
- [28] Li X., "Niching without niching parameters: particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, 2010, pp. 150-169.
- [29] Liang J. J., B. Y. Qu, P. N. Suganthan, and A. G. H. Díaz, *Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization*, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report 201212, 2013.
- [30] Liu B., X. Zhang, and H. Ma, "Hybrid differential evolution for noisy optimization," in *IEEE Congress on Evolutionary Computation*, 2008, pp. 587-592.
- [31] Liu J., and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, 2005, pp. 448-462.
- [32] Mahfoud S. W., "Crowding and preselection revisited," in *Parallel Problem Solving from Nature*, vol. 2, 1992, pp. 27-36.
- [33] Markon S., D. V. Arnold, T. Back, T. Beielstein, and H. G. Beyer, "Thresholding-a selection operator for noisy ES," in *IEEE Congress on Evolutionary Computation*, vol. 1, 2001, pp. 465-472.
- [34] Marsaglia G., and W. W. Tsang, "The ziggurat method for generating random variables," *Journal of Statistical Software*, vol. 5, no. 8, 2000, pp. 1-7.
- [35] Mendel E., R. A. Krohling, and M. Campos, "Swarm algorithms with chaotic jumps applied to noisy optimization problems," *Information Sciences*, vol. 181, no. 20, 2011, pp. 4494-4514.
- [36] Mengshoel O. J., and D. E. Goldberg, "Probabilistic crowding: deterministic crowding with probabilistic replacement," in *Genetic and Evolutionary Computation*, 1999, pp. 409-416.
- [37] Miller B. L., *Noise, sampling, and efficient genetic algorithms*, IlliGAL Report no. 97001, 1997.
- [38] Mininno E., and F. Neri, "A memetic differential evolution approach in noisy optimization," *Memetic Computing*, vol. 2, no. 2, 2010, pp. 111-135.
- [39] Narendra K. S., and M. L. A. A. Thathachar, "Learning automata-a survey," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 4, 1974, pp. 323-334.
- [40] Pan H., L. Wang, and B. Liu, "Particle swarm optimization for function optimization in noisy environment," *Applied Mathematics and Computation*, vol. 181, no. 2, 2006, pp. 908-919.
- [41] Park T., and K. R. Ryu, "Accumulative sampling for noisy evolutionary multi-objective optimization," in *ACM Genetic and Evolutionary Computation*, 2011, pp. 793-800.

- [42] Paz E. C., "Adaptive sampling for noisy problems," in *Genetic and Evolutionary Computation*, Springer Berlin Heidelberg, 2004, pp. 947-958.
- [43] Peng L., S. Liu, R. Liu, and L. Wang, "Effective long short-term memory with differential evolution algorithm for electricity price prediction," *Energy*, vol. 162, 2018, pp. 1301-1314.
- [44] Pietro A. D., L. While, and L. Barone, "Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions," in *IEEE Congress on Evolutionary Computation*, vol. 2, 2004, pp. 1254-1261.
- [45] Price K., R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*, Springer Science & Business Media, 2006.
- [46] Qin A. K., and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *IEEE Congress on Evolutionary Computation*, vol. 2, 2005, pp. 1785-1791.
- [47] Rahnamayan S., H. R. Tizhoosh, and M. Salama, "Opposition-based differential evolution for optimization of noisy problems," in *IEEE Congress on Evolutionary Computation*—, 2006, pp. 1865-1872.
- [48] Rakshit P., A. Konar, and A. K. Nagar, "Artificial Bee Colony induced multi-objective optimization in presence of noise," in *IEEE Congress on Evolutionary Computation*, 2014, pp. 3176-3183.
- [49] Rakshit P., A. Konar, and Atulya K. Nagar, "Type-2 fuzzy induced non-dominated sorting bee colony for noisy optimization," in *IEEE Congress on Evolutionary Computation*, 2015, pp. 3176-3183.
- [50] Rakshit P., A. Konar, and S. Das, "Noisy Evolutionary Optimization algorithms-A comprehensive survey," *Swarm and Evolutionary Computation*, vol. 33, 2017, pp. 18-45.
- [51] Rakshit P., A. Konar, P. Bhowmik, I. Goswami, S. Das, L. C. Jain, and A. K. Nagar, "Realization of an adaptive memetic algorithm using differential evolution and Q-learning: a case study in multirobot path planning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 4, 2013, pp. 814-831.
- [52] Rakshit P., A. Konar, S. Das, L. C. Jain, and A. K. Nagar, "Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 7, 2014, pp. 922-937.
- [53] Rakshit P., and A. Konar, "Differential evolution for noisy multiobjective optimization," *Artificial Intelligence*, vol. 227, 2015, pp. 165-189.
- [54] Rakshit P., and A. Konar, "Extending multi-objective differential evolution for optimization in presence of noise," *Information Sciences*, vol. 305, 2015, pp. 56-76.
- [55] Rakshit P., and A. Konar, "Non-dominated sorting bee colony optimization in the presence of noise," *Soft Computing*, vol. 20, no. 3, 2016, pp. 1139-1159.
- [56] Rakshit P., and A. Konar, *Principles in Noisy Optimization: Applied to Multi-agent Coordination*, Springer, 2018, ISBN 978-981-10-8641-0.
- [57] Rakshit P., and A. Konar, "Realization of learning induced self-adaptive sampling in noisy optimization," *Applied Soft Computing*, vol. 69, 2018, pp. 288-315.
- [58] Rattray M., and J. Shapiro, "Noisy fitness evaluation in genetic algorithms and the dynamics of learning," *Foundations of genetic algorithms: 4th workshop: revised papers*, ed. / Richard Belew, Michael Vose, Morgan Kaufmann, San Francisco, vol. 4, 1998, pp. 117-139.
- [59] Rudolph G., Günter, "A partial order approach to noisy fitness functions," in *IEEE Congress on Evolutionary Computation*, vol. 1, 2001, pp. 318-325.
- [60] Samma H., C. P. Lim, and J. M. Saleh, "A new reinforcement learning-based memetic particle swarm optimizer," *Applied Soft Computing*, vol. 43, 2016, pp. 276-297.
- [61] Sano Y., and H. Kita, "Optimization of noisy fitness functions by means of genetic algorithms using history of search with test of estimation," in *IEEE Congress on Evolutionary Computation*, vol. 1, 2002, pp. 360-365.
- [62] Sano Y., H. Kita, I. Kamihira, and M. Yamaguchi, "Online optimization of an engine controller by means of a genetic algorithm using history of search," in *IEEE Industrial Electronics Society*, vol. 4, 2000, pp. 2929-2934.
- [63] Siegmund F., A. H. C. Ng, and K. Deb, "A comparative study of dynamic resampling strategies for guided evolutionary multi-objective optimization," in *IEEE Congress on Evolutionary Computation*, 2013, pp. 1826-1835.
- [64] Stagge P., "Averaging efficiently in the presence of noise," in *Parallel Problem Solving from Nature*, Springer Berlin Heidelberg, 1998, pp. 188-197.
- [65] Storn R., and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, 1997, pp. 341-359.
- [66] Stroud P. D., "Kalman-extended genetic algorithm for search in nonstationary environments with noisy fitness evaluations," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, 2001, pp. 66-77.
- [67] Syberfeldt A., A. Ng, R. I. John, and P. Moore, "Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling," *European Journal of Operational Research*, vol. 204, no. 3, 2010, pp. 533-544.
- [68] Tanabe R., and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *IEEE Congress on Evolutionary Computation*, 2013, pp. 71-78.
- [69] Tang L., Y. Dong, and J. Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, 2014, pp. 560-574.
- [70] Wang L., H. Hu, XY. Ai, and H. Liu, "Effective electricity energy consumption forecasting using echo state network improved by differential evolution algorithm," *Energy*, vol. 153, 2018, pp. 801-815.
- [71] Zeng YR., Y. Zeng, B. Choi, and L. Wang, "Multifactor-influenced energy consumption forecasting using enhanced back-propagation neural network," *Energy*, vol. 127, 2017, pp. 381-396.
- [72] Zhan ZH., ZJ. Wang, H. Jin, and J. Zhang, "Adaptive distributed differential evolution," *IEEE Transactions on Cybernetics*, 2019, pp. 1-15.

- [73] Zhang J. Q., L. W. Xu, J. Ma, M. C. Zhou, "A learning automata-based particle swarm optimization algorithm for noisy environment," in *IEEE Congress on Evolutionary Computation*, 2015, pp. 141-147.
- [74] Zhang J., and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, 2009, pp. 945-958.
- [75] Zhang Z., and T. Xin, "Immune algorithm with adaptive sampling in noisy environments and its application to stochastic optimization problems," *IEEE Computational Intelligence Magazine*, vol. 2, no. 4, 2007, pp. 29-40.

**Declaration of interests**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

--