



Differential evolution for noisy multiobjective optimization



Pratyusha Rakshit*, Amit Konar

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

ARTICLE INFO

Article history:

Received 20 December 2013

Received in revised form 15 June 2015

Accepted 15 June 2015

Available online 18 June 2015

Keywords:

Noise

Differential evolution for multiobjective optimization

Sampling

Interquartile range

Skewness

Dominance probability

ABSTRACT

We propose an extension of multiobjective optimization realized with the differential evolution algorithm to handle the effect of noise in objective functions. The proposed extension offers three merits with respect to its traditional counterpart. First, an adaptive selection of the sample size for the periodic fitness evaluation of a trial solution based on the fitness variance in its local neighborhood is proposed. This avoids the computational complexity associated with the unnecessary reevaluation of quality solutions without disregarding the necessary evaluations for relatively poor solutions to ensure accuracy in fitness estimates. The second strategy is concerned with determining the expected value of the noisy fitness samples on the basis of their distribution, instead of their conventional averaging, as the fitness measure of the trial solutions. Finally, a new crowding-distance-induced probabilistic selection criterion is devised to promote quality solutions from the same rank candidate pool to the next generation, ensuring the population quality and diversity in the objective spaces. Computer simulations performed on a noisy version of a well-known set of 23 benchmark functions reveal that the proposed algorithm outperforms its competitors with respect to inverted generational distance, spacing, error ratio, and hypervolume ratio metrics.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The multiobjective optimization (MOO) literature has witnessed a radically different perspective in solving real-world problems using evolutionary computing methods. A MOO is concerned with mathematical optimization problems involving two or more complex, nonlinear, conflicting objectives to be optimized simultaneously. Usually, a derivative-free single-objective optimization algorithm generates new trial solutions that are biased toward the better region of the objective space, and weeds out poor solutions using a competitive selection over iterations. However, for a nontrivial MOO problem, there exists no single solution that simultaneously optimizes each objective. To jointly optimize multiple objective functions in a MOO [1], selection of trial solutions is performed by Pareto ranking, which is concerned with judiciously identifying nondominated trial solutions from the rest of the population. Pareto ranking is induced by the fitness measure of all objective functions for individual trial solutions.

The objectives, being functions of certain variables describing a specific problem, usually return a unique value for the variables in their argument. However, in many scientific/engineering problems, it has been observed that even though the measurements of the variables remain constant, the objective functions return different values because of noise-induced dynamic variation of the objective surfaces. This class of problem is referred to as the “noisy optimization problem.” Noise

* Corresponding author. Tel.: +91 9477399645.

E-mail address: pratyushar1@gmail.com (P. Rakshit).

creeps into the picture because of technological limitations, modeling errors, and incomplete data, leading to different results from repeated evaluations for the same set of parameter values of the objective functions. In such circumstances, a quality trial solution in a MOO may be deprived of being promoted to the next generation because of its poor (noisy) fitness estimates, while a deceptive solution with illusive good fitness may not be discarded from the current population [2,3].

This paper addresses the issues of uncertainty management (regarding the selection of qualitative trial solutions) in MOO in the presence of noise by incorporating the following three policies: adaptation of the sample size of a trial solution for its periodic fitness evaluation, expected fitness estimation from the measured noisy fitness samples, and crowding-distance-induced stochastic selection. First, the sample size for periodic fitness evaluation of each trial solution is adapted by means of the fitness variance in their local neighborhood. “Sampling” refers to the periodic fitness evaluation of a trial solution to diminish the risk of promoting inferior solutions in the noisy environment. It is worth mentioning that the adaptive selection of the sample size is momentous as increasing the sample size augments the quality measure of fitness at the cost of additional runtime. Here a nonlinear form (capturing the relationship between the sample size of a trial solution and the fitness variance in its local neighborhood) induced by an exponential function is regarded to efficiently balance the trade-off between runtime complexity and computational accuracy.

Second, while measuring the fitness of a trial solution, traditional methods [4–6] refer to the average fitness of the samples. However, the average fitness presumes equal probability of occurrence of all fitness samples, and thus returns a poor fitness estimate when the noise variance (in the fitness measure of the solutions) in the local neighborhood of a selected trial solution is large. This problem is circumvented here by referring to the expected value of the fitness samples as the true fitness estimate of a trial solution. The expected fitness concerned with the occurrence probability of the fitness samples seems to give a better fitness measure of a given trial solution. We introduce a novel strategy to evaluate the expected fitness of the trial solutions from the distribution of the fitness samples in the entire sample space. In the present context, a density-based nonuniform partitioning of the fitness sample space is employed to capture the uncertainty involved in the fitness measurement of the noisy fitness samples.

Finally, we develop a probabilistic selection (PS) policy to encapsulate the diversity as well as the quality of the non-dominated trial solutions even in noisy fitness landscapes. It is observed that the deterministic selection scheme of the crowding-distance-based sorting (used for promoting trial solutions from the same rank candidate pool to the next generation), which is employed in traditional MOO algorithms, can lead to suboptimal or misleading sets of nondominated solutions in the noisy environment even when sampling is used [7]. The selection strategy here depends not only on the density of nondominated solutions surrounding an individual in the objective space, but also on the reliability of its measured fitness samples. We develop a new probabilistic measure of the reliability based on the skewness of the distribution of the fitness samples. The degree of asymmetry of the distribution of the fitness samples is captured by skewness. Consequently, it provides a unique approach for identifying the rare fitness samples lying in the tail of the distribution. These infrequent samples (far away from the expected fitness) are assumed to occur because of the creeping of noise in the fitness landscapes. The rarer the occurrence of the infrequent samples (i.e., the closer the fitness samples are to the expected value with small skewness of the distribution), the greater is the degree of credibility of the fitness estimates of a given trial solution. The trial solutions having a greater crowding distance and a high grade of reliability (assessed using the probability of occurrence of no rare samples) are given more precedence during ranking of solutions in the same front.

The evolutionary component of the proposed noisy MOO algorithm has been realized here by the differential evolution for MOO (DEMO) [8] algorithm for its proven merits in global optimization. Some of the attractive features of DEMO justifying its selection in the design of the proposed noisy optimization algorithm include the simplicity of its structure leading to ease of coding, very few control parameters, and faster convergence [48,49] in comparison with other MOO algorithms.

Performance analysis of the proposed noisy optimization algorithm realized with DEMO—referred to as “differential evolution for noisy MOO” (DENMO) henceforth—is studied using the noisy version of a set of 23 benchmark functions. Experiments were undertaken to compare the potency of the proposed algorithm with differential evolution for MOO with noise (DEMON) [9], nondominated sorting genetic algorithm II (NSGA-II) with α -dominance operator (NSGA-II-A) [10], confidence-based dynamic resampling (CDR) [11], simulated annealing for noisy MOO [12], elitist evolutionary multiagent system [13], multiobjective evolutionary algorithm with robust features (MOEA-RF) [14], modified NSGA-II [7], noise-tolerant strength Pareto evolutionary algorithm [15], and Pareto front-efficient global optimization [16]. In this study, the objective functions are contaminated with noise samples taken from five noise distributions—namely, Gaussian, Poisson, Rayleigh, exponential, and random (with positive and negative expeditions of the noise amplitude within $\pm 25\%$ of the true fitness function values). Experiments reveal that the proposed realization outperforms other algorithms for four important performance metrics—that is, *inverted generational distance* (IGD), *spacing*, *error ratio* (ER), and *hypervolume ratio* (HVR).

The paper is divided into seven sections. Section 2 briefly reviews the literature on the strategies adopted by evolutionary algorithms to solve noisy MOO problems. In Section 3, we provide an overview of MOO and the DEMO algorithm. Section 4 provides the noise handling mechanism in DENMO. The experimental settings for the benchmarks and the simulation strategies are explained in Sections 5 and 6, respectively. Conclusions are given in Section 7.

2. Literature review

Recently, researchers have been very interested in developing robust MOO algorithms that can search for optimal solutions even when deceived by noise. Stagge et al. [17] employed the concept of *sampling* (fitness reevaluation of the same

trial solutions) to enhance the objective estimates in the presence of noise. The average fitness of the samples provides a relatively safe measure of the fitness of a trial solution, thereby reducing the possible risk of certifying a low-grade trial solution to enter the optimal Pareto front. Syberfeldt et al. [11] proposed a CDR strategy to adaptively select the sample size using a Welch confidence interval test. For a user-defined confidence level, the resampling of solutions is performed iteratively until the noise is sufficiently reduced, which is assessed by the Welch confidence interval. Inspired by the role of the momentum term in back-propagation in neural networks, Goh and Tan [14] adopted an *experiential-learning-directed perturbation* strategy to govern the candidate movement toward the direction of fitness improvement on the basis of the information acquired from the last few generations for faster convergence. Moreover, the trade-off between the explorative and exploitative capabilities of the MOO here are balanced by the proposed *gene adaptation selection strategy*. Finally a *possibility and necessity* measure induced *possibilistic archiving* method is used to resolve the uncertainty in including a solution in the nondominated set.

Babbar et al. [7] recommended a new Pareto ranking strategy to approve the ingress of apparently inferior trial solutions in the optimal Pareto front by satisfying a statistical comparison. The comparison is concerned with the difference between the mean values of the fitness samples of two individual solutions with respect to the scaled average of their fitness variance. The executed ranking policy thus prevents the probable dismissal of quality solutions from the optimal Pareto front as the true potential of such solutions may be incomprehensible because of noise. However, the inadequacy of their strategy lies in the context of uniform sampling. The scope of a new noise-aware α -dominance operator was explored by Boonma et al. [10] to evaluate the dominance relationship (with a confidence level of α) between two individuals on the basis of the measurement of their noisy fitness samples. The merit of the proposed strategy lies in no prerequisite of knowledge of the noise distribution in the objective space being needed.

The “soft” selection scheme of the multiagent system employed by Siwik et al. [13] ensures the survival of a quality solution (agent) even with its seemingly poor fitness estimates in a noisy uncertain environment. Some resources, denoted by *life energy*, are assigned to each solution (agent) on the basis of its fitness: life energy is an indirect measure of the agent’s survival and offspring-generation capabilities, and is reduced on reallocation of some resources to other dominating agents, and vice versa. Hence, if a quality solution seems to be dominated, it loses only a finite part of its energy, and it can prolong its functionality until its energy is reduced to zero. The strategy also guarantees that under little influence of noise, such a quality solution may get an opportunity to enhance its energy in the future by communicating with the agents it dominates. Buche et al. [15] proposed a noise-tolerant MOO algorithm. Here, each population member is assigned with a *dominance-dependant lifetime*, which varies inversely with the number of solutions it dominates. Thus, the strategy defends the overall population against the impact of the illusive fitness of unreliable solutions. Conversely, the survival of quality solutions is ensured by reevaluating the archived solutions with expired lifetime and adding them to the current population. Finally, the loss of information in noisy environments is hindered by appending all the solutions with nonexpired lifetime to the archive along with the nondominated members.

Rakshit et al. [9] implemented a nonuniform sampling strategy utilizing a linear relationship between the sample size of a trial solution and the fitness variance in its local neighborhood. Moreover, the statistical expectation of the fitness samples is referred to as the “fitness measure of the trial solutions.” On the basis of the proposed strategies, Goldberg’s method [7] is extended to verify possible accommodation of slightly inferior population members in the optimal Pareto front. Das et al. [18] integrated a threshold-based selection policy with a random-scale-factor-induced stochastic model of the differential evolution to proficiently explore the dynamic fitness landscape contaminated with noise. The strategy promotes an offspring vector to the next generation only if its fitness is better than that of its respective parent by a certain threshold value. This, in turn, circumvents the approval of misleading solutions over generations.

A *probabilistic dominance* approach was adopted by Hughes [19] to deal with the uncertainty of decisions concerned with the supremacy of a trial solution over another in the context of noisy fitness landscapes. The placement of trial solutions in the Pareto fronts is thus governed by the quantified dominance probability. Singh et al. [20] utilized a hypothesis test based on a Student *t* distribution for stochastic selection of nondominated solutions. Knowles and Hughes [16] used a *Gaussian-process-assisted surrogate model* to discover the most competent solution possessing the largest expected improvement at the minimum cost. It is designed with the aim of optimization on a very restricted evaluation budget by weighing up both the predicted fitness of solutions and the associated error. Knowles et al. [21] proposed a fitness-distance-based correlation strategy for quality performance of a MOO problem in the presence of noise constrained by a very limited evaluation budget. Mattila et al. [12] devised a new criterion to selectively include the candidate solutions in the optimal Pareto front based on the probability that the solution is dominated by the nondominated solutions of the current generation. In addition, the generation of offsprings is guided by the information content of the nondominated solutions of the previous generations. The strategy proposed here is realized in the context of simulated annealing.

3. Preliminaries

In this section, we provide an overview of two important concepts—MOO and DEMO—which will be referred to frequently in the rest of the paper to develop a solution of noisy MOO problems.

3.1. Multiobjective optimization

The principal objective of MOO algorithms is to attain Pareto optimal solutions in a single run. The following definitions will be used to explain DEMO and its extension.

Definition 1. Let us consider two trial solutions \vec{X}_i and \vec{X}_j of a MOO problem where all N objectives need to be minimized. The k th objectives of \vec{X}_i and \vec{X}_j are represented by $f_k(\vec{X}_i)$ and $f_k(\vec{X}_j)$, respectively. Then \vec{X}_i is said to **dominate** \vec{X}_j , denoted by $\vec{X}_i < \vec{X}_j$, if the following conditions are jointly satisfied:

1. \vec{X}_i is no worse than \vec{X}_j in all objective functions—that is, $f_k(\vec{X}_i) \leq f_k(\vec{X}_j)$ for $k = [1, N]$.
2. \vec{X}_i is strictly better than \vec{X}_j for at least one objective function—that is, $f_l(\vec{X}_i) < f_l(\vec{X}_j)$ for at least one $l \in [1, N]$.

Definition 2. In a MOO, a trial solution \vec{X}^* is said to be nondominated if there is no \vec{X} that dominates \vec{X}^* .

Definition 3. Let P be a set of potential solutions to a MOO problem. $P' \subseteq P$ is called the **optimal Pareto set** of solutions if the members of P' are not dominated by any member of P .

Definition 4. Let f_1 and f_2 be two objective functions in a MOO problem, and let \vec{X} , \vec{X}_i , and \vec{X}_j be the members of a nondominated list of solutions. Furthermore, \vec{X}_i and \vec{X}_j are the nearest neighbors of \vec{X} in the objective spaces. The **crowding distance** of a trial solution \vec{X} in a nondominated set depicts the perimeter of a hypercube formed by its nearest neighbors—that is, \vec{X}_i and \vec{X}_j —at the vertices in the fitness landscapes. In other words, the crowding distance of \vec{X} is computed by $CD(\vec{X}) = |f_1(\vec{X}_i) - f_1(\vec{X}_j)| + |f_2(\vec{X}_i) - f_2(\vec{X}_j)|$.

3.2. Differential evolution for MOO

DEMO [8] employs an evolutionary strategy that utilizes the advantages of the differential evolution [22] with the mechanisms of Pareto-based ranking and crowding distance sorting. An overview of the main steps of the DEMO algorithm is presented next:

- (a) **Initialization:** The i th member $\vec{X}_i(t)$ of the current population P_t for $i = [1, NP]$ at generation $t = 0$ is selected by uniformly randomizing individuals in the range $[\vec{X}^{\min}, \vec{X}^{\max}]$, where $\vec{X}^{\min} = \{x_1^{\min}, x_2^{\min}, \dots, x_D^{\min}\}$ and $\vec{X}^{\max} = \{x_1^{\max}, x_2^{\max}, \dots, x_D^{\max}\}$, and thus the j th component of the i th member at $t = 0$ is initialized by

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

for $j = [1, D]$. Here $\text{rand}_{i,j}(0, 1)$ is a uniformly distributed random number lying between 0 and 1. The crossover rate is initialized in $[0, 1]$. The k th objective function $f_k(\vec{X}_i(0))$ is evaluated for the target vector $\vec{X}_i(0)$ with $i = [1, NP]$ and $k = [1, N]$.

- (b) **Mutation:** A donor vector $\vec{V}_i(t)$ corresponding to each $\vec{X}_i(t)$ is created by randomly selecting three other members $\vec{X}_{\text{rand1}}(t)$, $\vec{X}_{\text{rand2}}(t)$, and $\vec{X}_{\text{rand3}}(t)$ ($i \neq \text{rand1} \neq \text{rand2} \neq \text{rand3}$) from P_t , where

$$\vec{V}_i(t) = \vec{X}_{\text{rand1}}(t) + F \times (\vec{X}_{\text{rand2}}(t) - \vec{X}_{\text{rand3}}(t)) \quad (2)$$

and F is a scaling factor in $[0, 2]$. The mutation operation given in (2) is referred to as DE/rand/1 [23]. This is done for $i = 1$ to NP . There are other mutation operations, details of which are available in [23].

- (c) **Crossover:** There are two types of crossover—binomial and exponential [23].

In the case of binomial crossover, a trial vector $\vec{U}_i(t)$ is generated for each pair of $\vec{V}_i(t)$ and $\vec{X}_i(t)$ by the following operation:

$$u_{i,j}(t) = \begin{cases} v_{i,j}(t) & \text{if } \text{rand}_{i,j} \leq CR \text{ or } j = j_{\text{rand}}, \\ x_{i,j}(t) & \text{otherwise,} \end{cases} \quad \forall j = [1, D], \quad (3)$$

where $j_{\text{rand}} \in [1, D]$ is a randomly chosen index.

In the case of exponential crossover, we randomly select an integer n from $[1, D]$. It represents the starting point of the target vector $\vec{X}_i(t)$ to commence the exchange of components with the corresponding donor vector $\vec{V}_i(t)$. We also select another integer L from $[1, D]$, where L denotes the number of components the donor vector contributes to the target. The selection of L is given in [24]. Now, we obtain the trial vector $\vec{U}_i(t)$ with

$$u_{i,j}(t) = \begin{cases} v_{i,j}(t) & \text{for } j = \langle n \rangle_D, \langle n+1 \rangle_D, \dots, \langle n+L-1 \rangle_D, \\ x_{i,j}(t) & \text{for all other } j \in [1, D], \end{cases} \quad \forall j = [1, D], \quad (4)$$

where $\langle \rangle_D$ denotes a modulo function with modulus D . The k th objective function $f_k(\vec{U}_i(t))$ is evaluated for the trial vector $\vec{U}_i(t)$ with $i = [1, NP]$ and $k = [1, N]$.

- (d) **Selection:** Now the trial vector $\bar{U}_i(t)$ replaces the corresponding target vector $\bar{X}_i(t)$ if $\bar{U}_i(t) < \bar{X}_i(t)$ based on the measures of their objective functions. However, when $\bar{U}_i(t)$ and $\bar{X}_i(t)$ are nondominated, $\bar{U}_i(t)$ is inserted into the current population P_t ; otherwise, $\bar{U}_i(t)$ is discarded. This assists in a faster convergence to the true Pareto front. This step is repeated for all the trial vectors, and hence a population of solution vectors is obtained with size S lying in $[NP, 2NP]$.
- (e) **Nondominated sorting:** The population P_t of size S ($NP < S < 2NP$) thus obtained is sorted into a number of Pareto fronts according to nondominating criteria. All the nondominated solutions of the current population are given same rank 1 (named optimal Pareto front $Front_Set(1)$). The second front is formed by identifying the nondominated solutions from the set $P_t - Front_Set(1)$. The ranking process continues until all the nondominated sets are identified and ranked as $Front_Set(1)$, $Front_Set(2)$, $Front_Set(3)$, and so on. The pseudocode for nondominated sorting is given in [25].
- (f) **Truncation of the extended population using the crowding-distance-based ranking:** The parent population for the next iteration P_{t+1} is constructed by selecting the nondominated sets of solutions from P_t according to the ascending order of their Pareto ranking. Let $Front_Set(l)$ be the set beyond which no other set can be accommodated in P_{t+1} (i.e., by adding $Front_Set(l)$, $|P_{t+1}|$ exceeds NP). Then the solutions in $Front_Set(l)$ are sorted in descending order of the crowding distance. To ensure diversity in the population, the solutions with the highest crowding distances are included in P_{t+1} from $Front_Set(l)$ until $|P_{t+1}|$ becomes NP .
- (g) **Convergence:** After each evolutionary step, we repeat the process from step (b) until one of the following conditions for convergence is satisfied. The stop criteria include a bound by the number of iterations, achieving a sufficiently low error or aggregations thereof.

4. Optimization in the presence of noise

This paper is concerned with real-world MOO problems, where measured variables are contaminated with stochastic noise due to the aging of sensors and/or measurement inaccuracies. The objectives in MOO, being functions of measurement variables, are also contaminated with noise and thus lack precision. Traditional deterministic MOO algorithms are unable to capture the global optima on the fitness landscapes in the presence of stochastic noise [7]. This section extends the traditional DEMO algorithm to improve its optimization efficacy in the presence of stochastic noise [26,27] in the ways described in the following sections.

4.1. Adaptive selection of sample size

First, the repercussion of noise on the fitness measurements is diminished by taking repeated samples of the objective functions for each trial solution. Because objective functions in noisy MOO are contaminated with noise, a single measure of the fitness estimate is likely to be largely influenced by noise. To ensure the fitness estimates are minimally influenced by noise, one approach is to take a large sample size of the fitness estimates. Apparently, the larger the sample size, the higher the precision of the fitness estimates of a given trial solution. However, a very large sample size adds computational overhead without much contribution to improving the fitness estimates of a quality solution. Contrarily, if very few samples are considered, the fitness estimates will be inaccurate, leading to selection of a deceptive trial solution over a qualitative one. Sample size selection is thus constrained by a trade-off between accurate fitness estimation and computational complexity. Consequently, the incorporation of a method for an adaptive selection of the sample size in a MOO problem in the presence of noise is envisaged.

We attempt to realize this strategy in the noisy MOO algorithm by introducing a test criterion on the fitness variance in the local neighborhood of a trial solution to control its sample size. In other words, the essence of the current work is to induce a trial solution with the information about the sample size (for its periodic fitness evaluation) acquired from the fitness variance in its local neighborhood. The design philosophy adopted here relies on an underlying premise that a possible measure of creeping of noise in the neighborhood of a trial solution is anticipated from the fitness variance of a subpopulation around it. A large fitness variance (among the trial solutions) in the subpopulation indicates a large-scale detrimental effect of noise in the local neighborhood (of the given trial solution) in the fitness landscape. Under this situation, it is obvious to draw a large sample size to accurately estimate the fitness of the given trial solution. On the other hand, small fitness variance in the local neighborhood portrays a smaller contamination effect of noise, and hence, in turn, requires a small sample size to reduce computational cost without sacrificing the quality fitness measurement.

- (a) **Mathematical model for sample size adaptation:** Several formulations of the sample size selection can be adopted by maintaining a smaller sample size at lower fitness variance and a larger sample size at larger fitness variance in the local neighborhood of a trial solution. One simple approach could be proportional selection, where the sample size n is proportional to the fitness variance v in the local neighborhood of a trial solution. However, simply setting a proportional law demands a very large sample size n at very large variance v , which may not be economic in many contexts, particularly for practical MOO problems. Alternatively, a logistic function may be employed to serve the purpose. Expression (5) refers to such a model with sample size $n_{k,i}$ corresponding to the k th objective function $f_k(\bar{X}_i)$ of the trial solution \bar{X}_i for $k = [1, N]$, having a lower bound $n_{k,i} \geq n^{\min}$ and a largest bound $n_{k,i} \leq n^{\max}$. As shown in Fig. 1, the function within these two bounds is a continuous nondecreasing function, and the fitness variance $v_{k,i}$ in the local neighborhood of \bar{X}_i (corresponding to the k th objective) controls the value of $n_{k,i}$ within $[n^{\min}, n^{\max}]$ for $k = [1, N]$.

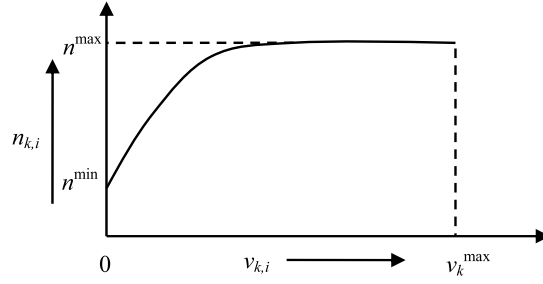


Fig. 1. The nonlinearity used to adapt the sample size with fitness variance in the local neighborhood.

In Fig. 1, v_k^{\max} denotes the maximum fitness variance in the local neighborhood of all trial solutions in the current population for the k th objective with $k = [1, N]$.

$$n_{k,i} = n^{\min} + (n^{\max} - n^{\min}) \times (1 - \exp(-v_{k,i})). \quad (5)$$

- (b) **Local neighborhood selection:** To determine the neighborhood of a D -dimensional trial solution \vec{X}_i from a population of NP members in the parameter space, the search range $[x_j^{\min}, x_j^{\max}]$ is first divided into equally spaced intervals of $\Delta x_j = (x_j^{\max} - x_j^{\min})/NP$ for $j = [1, D]$. Now the neighborhood of \vec{X}_i , denoted by N_i , is constructed by selecting the trial solutions lying within a hyperspace bounded by $[\Delta \vec{X}_i^{\min}, \Delta \vec{X}_i^{\max}]$, where

$$\Delta \vec{X}_i^{\min} = \{x_{i,1} - \Delta x_1, x_{i,2} - \Delta x_2, \dots, x_{i,D} - \Delta x_D\}, \quad (6.1)$$

$$\Delta \vec{X}_i^{\max} = \{x_{i,1} + \Delta x_1, x_{i,2} + \Delta x_2, \dots, x_{i,D} + \Delta x_D\}. \quad (6.2)$$

- (c) **Fitness variance in the local neighborhood:** Once all trial solutions \vec{X}_j in the local neighborhood N_i around \vec{X}_i have been identified, the fitness variance in N_i is determined as follows:

$$v_{k,i} = \frac{\sum_{\forall \vec{X}_j \in N_i} (n_{k,j} \times s_{k,j})}{\sum_{\forall \vec{X}_j \in N_i} n_{k,j}}, \quad (7)$$

where $n_{k,j}$ symbolizes the sample size and $s_{k,j}$ represents the spread of the samples of $f_k(\vec{X}_j)$ for $k = [1, N]$, which will be discussed very shortly.

4.2. Sample-distribution-based fitness estimation

Because of the adaptive selection of the sample size, we have multiple fitness measurements of a given trial solution. These measurements in general take different values with a nonzero variance. Traditional noisy MOO algorithms estimate the true fitness estimate of a given trial solution by averaging over its noisy fitness samples. However, this averaging strategy presumes equal probability of occurrence of all fitness samples, thereby offering a poor fitness estimate in most of the real situations. This can be circumvented by the second extension of the traditional DEMO algorithm in this paper. The alternative approach proposed is concerned with the fitness-sample-density-based partitioning of the sample space and then referring to the expected value of the entire sample space as the true fitness estimate of a trial solution. In other words, the fitness estimate of a trial solution is more biased toward the fitness samples in the crowded regions in the sample space, while imposing less importance on the rare fitness samples. It is assumed here that the fitness samples in the sparse zones in the sample space result from noise contamination. Sample-distribution-based fitness estimation (SDFE) includes four main steps:

- (a) **Fitness interval selection in the sample space:** We first record the variance $V_k(\vec{X}_i)$ of the measured samples of the k th fitness of trial solution \vec{X}_i . We also note down the minimum and the maximum values of the observed fitness samples—say, $f_k^{\min}(\vec{X}_i)$ and $f_k^{\max}(\vec{X}_i)$, respectively. Now the entire range $[f_k^{\min}(\vec{X}_i), f_k^{\max}(\vec{X}_i)]$ is first divided into two intervals of equal length. The resulting intervals are represented by $[f_k^{\min}(\vec{X}_i), f_k^{\text{mid}}(\vec{X}_i)]$ and $[f_k^{\text{mid}}(\vec{X}_i), f_k^{\max}(\vec{X}_i)]$, respectively where $f_k^{\text{mid}}(\vec{X}_i) = (f_k^{\min}(\vec{X}_i) + f_k^{\max}(\vec{X}_i))/2$. If the variance of the fitness samples lying in the first interval is found to be greater than $V_k(\vec{X}_i)/n_{k,i}$, it is again subdivided into two more equal intervals, represented by $[f_k^{\min}(\vec{X}_i), f_k^{\text{mid},1}(\vec{X}_i)]$ and $[f_k^{\text{mid},1}(\vec{X}_i), f_k^{\text{mid}}(\vec{X}_i)]$, respectively, where $f_k^{\text{mid},1}(\vec{X}_i) = (f_k^{\min}(\vec{X}_i) + f_k^{\text{mid}}(\vec{X}_i))/2$. The same approach is used in the second interval also. The same procedure is repeated for all subsequent intervals until the variance of the fitness samples in each interval falls below $V_k(\vec{X}_i)/n_{k,i}$. As a consequence, the entire sample space $[f_k^{\min}(\vec{X}_i), f_k^{\max}(\vec{X}_i)]$ is now divided into L intervals of unequal length as indicated in Fig. 2.

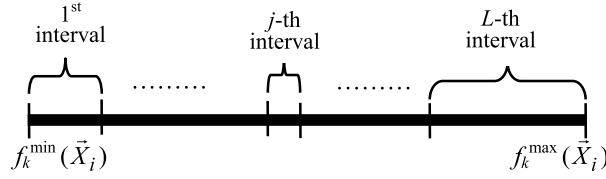


Fig. 2. Fitness intervals in the sample space.

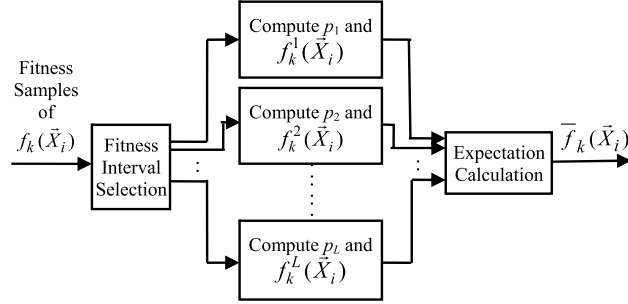


Fig. 3. Expected fitness calculation.

- (b) **Occurrence probability of the fitness samples:** It is evident from the proposed nonuniform partitioning of the sample space that the fitness samples in the longer intervals are rare samples, probably owing to noise, and hence their contribution to the true fitness estimate of \vec{X}_i should not be regarded similarly to that of the fitness samples in the smaller interval. We measure the contribution of the fitness samples of the interval j to the fitness estimate of the trial solution \vec{X}_i by a probability measure p_j :

$$p_j = n_{k,i}^j / n_{k,i} \quad \text{for } j = [1, L], \quad (8)$$

where $n_{k,i}^j$ represents the number of samples of $f_k(\vec{X}_i)$ in the j th interval for $j = [1, L]$.

- (c) **Expected fitness estimation:** Let $f_k^j(\vec{X}_i)$ be the median value of the fitness samples of \vec{X}_i in the j th interval, for $j = [1, L]$. We obtain the expected value—say, $E(f_k(\vec{X}_i)) = \bar{f}_k(\vec{X}_i)$ —of the k th fitness of \vec{X}_i by (9):

$$\bar{f}_k(\vec{X}_i) = \sum_{j=1}^L p_j \times f_k^j(\vec{X}_i). \quad (9)$$

The expected value thus obtained provides a unique measure of the k th objective of \vec{X}_i from the noisy local distribution of its fitness samples over a wide space. The fitness samples in a particular interval are represented here by their median because the median value of a frequency distribution is less prone to noisy measurements [28]. A schematic diagram of the evaluation of the expected fitness of the k th objective is given in Fig. 3. The entire procedure is performed for each objective—that is, $k = [1, N]$.

- (d) **Spread of the fitness samples:** The level of contamination of noise on $f_k(\vec{X}_i)$ can be captured by the spread of sample values of $f_k(\vec{X}_i)$ away from $\bar{f}_k(\vec{X}_i)$, denoted by $s_{k,i}$, for $k = [1, N]$. To evaluate $s_{k,i}$, the median values of fitness samples of each of the L intervals are sorted in ascending order of magnitude. The lower and upper quartiles from the sorted list are identified and recorded as $Q_{k,0.25}(\vec{X}_i)$ and $Q_{k,0.75}(\vec{X}_i)$, respectively. The interquartile range corresponding to $f_k(\vec{X}_i)$ is then used as a measure of $s_{k,i}$ as defined by (10) for $k = [1, N]$:

$$s_{k,i} = Q_{k,0.75}(\vec{X}_i) - Q_{k,0.25}(\vec{X}_i). \quad (10)$$

To eliminate the impact of the extreme values of the noisy fitness samples, the interquartile range is referred to as a measure of the spread of the samples of $f_k(\vec{X}_i)$ from the respective expected measurement $\bar{f}_k(\vec{X}_i)$, instead of the sample variance.

4.3. PS during truncation of the extended population

The next-generation population (of size NP) of a MOO is formed by identifying the trial solutions with higher Pareto ranks from the merged population (of size lying in $[NP, 2NP]$) in the current generation. It is apparent that the en-

tire accommodation of all the nondominated members of a definite front may not be granted in the next generation to keep the next-generation population size fixed at NP. A greedy selection process is then employed to promote the members from the same front to the next generation, ensuring diversity in the distribution of trial solutions in the objective spaces. To rank the solutions in the same front, the crowding distance metric is used. The crowding distance of a solution \vec{X}_i , denoted by $CD(\vec{X}_i)$, offers an estimate of the density of nondominated solutions neighboring \vec{X}_i . It is the sum of the distances between each of the fitness measures of its two nearest neighboring solutions in the objective spaces. In traditional MOO, the conqueror of the two same-rank solutions is the one having the greater crowding distance. This, in turn, improves the diversity performance of the algorithm, enriching its explorative capability.

This conventional process for ranking solutions in the same front using the crowding distance metric does not take into account the uncertainty in the measurement of the noisy fitness samples. The possible promotion of a trial solution \vec{X}_i to the next generation (from the same-rank candidate pool) in the presence of noise depends on two important issues. First, the greater the crowding distance measure $CD(\vec{X}_i)$, the higher is the chance of selecting \vec{X}_i from its competitors residing in the same front. The uncertainty involved in the fitness measurement is taken care of by the second criterion—the probability of the nonoccurrence of the rare samples of $f_k(\vec{X}_i)$. A higher value of this probability ensures proximity of the measured fitness samples to $\bar{f}_k(\vec{X}_i)$ in the sample space. The intermittent samples, far away from $\bar{f}_k(\vec{X}_i)$, are supposed to result from the contamination effect of noise, and evidently their occurrence produces a longer tail in the sample distribution. Finally, a new selection strategy, induced by the crowding distance and the probability of nonoccurrence of rare samples, is devised to promote quality solutions to the next generation from the same front.

The contamination effect of noise on fitness samples is realized here using quartile skewness (of the distribution of fitness samples) to capture the possible occurrence of rare fitness samples away from the expected fitness estimate. The quartile skewness $\gamma_{k,i}$, defined by (11), provides a robust measure of the degree of asymmetry of the distribution of fitness samples of $f_k(\vec{X}_i)$ with respect to the expected value $\bar{f}_k(\vec{X}_i)$ for $k = [1, N]$:

$$\gamma_{k,i} = \frac{(Q_{k,0.25}(\vec{X}_i) - \bar{f}_k(\vec{X}_i)) - (\bar{f}_k(\vec{X}_i) - Q_{k,0.75}(\vec{X}_i))}{Q_{k,0.75}(\vec{X}_i) - Q_{k,0.25}(\vec{X}_i)}. \quad (11)$$

It is apparent from (11) that $\gamma_{k,i} < 0$ indicates the tail on the left of the fitness sample distribution is more pronounced than the tail on the right, signifying that fitness samples with values comparatively less than $\bar{f}_k(\vec{X}_i)$ are rare. The reverse case is true for $\gamma_{k,i} > 0$. It is apparent that for $\gamma_{k,i}$ approaching -1 (or $\gamma_{k,i}$ approaching $+1$), the frequency of the occurrence of fitness samples lying on the left (or right) tail of the fitness sample distribution, far away from $\bar{f}_k(\vec{X}_i)$, is extremely small and these may be regarded as noisy samples [29]. It is thus evident that skewness is greatly influenced by the occurrence of rare noisy fitness samples far away from the expected value $\bar{f}_k(\vec{X}_i)$. Hence, it is expected that for $f_k(\vec{X}_i)$ less affected by noise, the measured fitness samples will be very close to $\bar{f}_k(\vec{X}_i)$ and hence $\gamma_{k,i} \approx 0$. This observation motivated us to denote $0 < |\gamma_{k,i}| < 1$ as the probability of occurrence of rare samples encompassing the uncertainty involved in the measurement of fitness samples of $f_k(\vec{X}_i)$. Thus, we define the probability of nonoccurrence of rare samples, providing a measure of the degree of reliability on the samples of $f_k(\vec{X}_i)$, as follows:

$$p_{k,i} = 1 - |\gamma_{k,i}|. \quad (12)$$

The normalized measure of $CD(\vec{X}_i)$, denoted by $\overline{CD}(\vec{X}_i)$, is given in (13), where \vec{X}_i and \vec{X}_j lie in the same Pareto front:

$$\overline{CD}(\vec{X}_i) = \frac{CD(\vec{X}_i)}{\sum_{\forall j} CD(\vec{X}_j)}. \quad (13)$$

Now, treating $\overline{CD}(\vec{X}_i)$ like probability and presuming that $p_{k,i}$ for $k = [1, N]$ and $\overline{CD}(\vec{X}_i)$ are independent, we define the selection probability of \vec{X}_i for the next generation as

$$ps_i = \prod_{k=1}^N p_{k,i} \times \overline{CD}(\vec{X}_i). \quad (14)$$

The product function introduced in (14) reveals that an increase in either $p_{k,i}$ for $k = [1, N]$ or $\overline{CD}(\vec{X}_i)$ or in both of them ensures an increase in ps_i .

The pseudocode for the proposed DENMO algorithm with N objectives is given below. Termination criteria for the algorithm include a bound by the maximum number of function evaluations (FEs) or the achievement of a sufficiently low difference between the performance metric values (discussed in Section 5.3) of two successive generations or aggregations thereof.

Procedure DENMO

Begin**I. Initialization:**

- I.(a). Randomly initialize a population of NP, D -dimensional individuals $P_t = \{\vec{X}_1(t), \vec{X}_2(t), \dots, \vec{X}_{NP}(t)\}$ following (1) at $t = 0$.
- I.(b). Set the sample size $n_{k,i}$ of the k th objective function $f_k(\vec{X}_i(t))$ corresponding to the target vector $\vec{X}_i(t)$ to n^{\min} for $i = [1, NP]$ and $k = [1, N]$.
- I.(c). Evaluate the expected fitness $\tilde{f}_k(\vec{X}_i(t))$ and the spread $s_{k,i}$ of the measured fitness samples of $f_k(\vec{X}_i(t))$ for $i = [1, NP]$ and $k = [1, N]$ using (9) and (10), respectively.

II. While the stopping criterion is not reached **do begin****II.(a) For** $i = 1$ to NP **do begin**

1. **Mutation:** Generate a donor vector $\vec{V}_i(t)$ corresponding to the i th target vector $\vec{X}_i(t)$ using (2).
2. **Crossover:** Generate the trial vector $\vec{U}_i(t)$ for the pair $\vec{X}_i(t)$ and $\vec{V}_i(t)$ using binomial crossover as in (3).
3. **Neighborhood selection:** The neighborhood of $\vec{U}_i(t)$ is obtained by selecting the members within a hyper-space bounded by $[\Delta \vec{U}_i^{\min}, \Delta \vec{U}_i^{\max}]$ as given in (6).

End For.**II.(b) /* Formation of merged population */****For** $i = 1$ to NP **do begin**

1. **Adaptive selection of sample size of** $\vec{U}_i(t)$: Evaluate the sample size of $f_k(\vec{U}_i(t))$ for $k = [1, N]$ using (5).
2. **Fitness estimation of** $\vec{U}_i(t)$: Evaluate the expected fitness and the spread of the measured fitness samples of $f_k(\vec{U}_i(t))$ for $k = [1, N]$ using (9) and (10), respectively.
3. **Selection of** $\vec{U}_i(t)$: Include $\vec{U}_i(t)$ in the current generation or discard it on the basis of the dominance relationship between $\vec{X}_i(t)$ and $\vec{U}_i(t)$ as discussed in Section 3.2, list item (d).

End For.

- II.(c) Sort the solutions of P_t into a number of Pareto fronts, denoted by *Front_Set*, following the nondominated sorting strategy as given in Section 3.2 list item (e).

II.(d) /* Stochastic selection from same-rank solutions */

1. Set $P_{t+1} \leftarrow \text{NULL}$ and $l \leftarrow 1$.
2. **Repeat**

$$P_{t+1} \leftarrow P_{t+1} \cup \text{Front_Set}(l); l \leftarrow l + 1;$$

Until $|P_{t+1}| + |\text{Front_Set}(l)| > \text{NP}$.

3. Calculate the selection probability of the trial solutions in *Front_Set*(l) using (14) and sort them in descending order of selection probability (such that the solutions with rank 1 and $|\text{Front_Set}(l)|$ have the highest and the least selection probabilities, respectively).
4. Set $P_{t+1} \leftarrow P_{t+1} \cup \text{top}(\text{NP} - |P_{t+1}|)$ trial solutions of *Front_Set*(l) with highest selection probability.

- II.(e) Increase the counter value $t \leftarrow t + 1$.

End While;**End.**

4.4. Comparative framework and parameter setting

The MOO algorithms used for the comparative study included DEMON [9], NSGA-II-A [10], CDR [11], simulated annealing for noisy MOO [12], elitist evolutionary multiagent system [13], MOEA-RF [14], modified NSGA-II [7], noise-tolerant strength Pareto evolutionary algorithm [15], and Pareto front-efficient global optimization [16]. These algorithms were selected for the comparative framework for their wide popularity in the realm of noisy MOO. The population size and the maximum number of FEs for all the algorithms were fixed at 50 and $10^4 \times D$ for a D -dimensional problem, respectively. To make the comparison fair, the population of all the algorithms (over all the benchmark functions discussed in Section 5.1) was initialized using the same random seeds. The best parametric setups for all these algorithms were chosen following their respective sources. In our proposed DENMO algorithm, the minimum and maximum sample sizes are considered to be $n^{\min} = 10$ and $n^{\max} = 30$, respectively, with a crossover rate of 0.9. The parameter settings are given in Tables 1 and 2.

5. Benchmark functions and evaluation metrics

5.1. Benchmark functions

The most challenging issue in substantiating the performance of a MOO algorithm is to identify the right benchmark functions with diverse characteristics such as multimodality, deception, isolation, and particularly, the location of the true

Table 1Parameter settings in D -dimensional search space.

DENMO		DEMON		NSGA-II-A		CDR		SA	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
CR	0.9	Crossover factor	0.9	Number of samples	30	Number of offsprings	25	Temperature	1
η^{\min}	10	Minimum sample size	10	Crossover rate	0.9	Mutation step size	0.5	Minimum sample size	5
η^{\max}	30	Maximum sample size	30	Crossover operator	SBX	Crossover operator	Single point	Maximum sample size	30
		Initial value of neighborhood restriction factor	10	Crossover distribution index	20	Crossover probability	0.8	Number of nondominated solutions used for generating candidate solutions	20
				Mutation rate	$1/D$				
				Mutation operator	Polynomial				
				Distribution index for mutation	20				
				SVM type	C-support vector classification				
				SVM kernel	Linear				
				Stopping criterion for SVM	$1e-3$				
				C parameter for SVM	1				
				Minimum confidence level	0.9				
				Maximum confidence level	0.99				

CDR, confidence-based dynamic resampling; CR, crossover rate; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; SA, simulated annealing; SBX, simulated binary crossover; SVM, support vector machine.

optimal Pareto front in the objective spaces to resemble complicated real-world problems. Traditional benchmark functions [31,32] usually have the global optima located either in the center of the parameter space or on the search bounds. Naturally, these benchmark functions are insufficient to profoundly validate the performance of a MOO algorithm. To surmount the above problem, a set of benchmark functions [30] was recommended in the Congress of Evolutionary Computation (CEC'2009) conference. The proposed benchmarks include extension, stretching, and rotation of the objective functions, thereby assimilating diversity in the optimization problems of the traditional benchmark functions.

The performance of the proposed DENMO algorithm is analyzed here with respect to the noisy-version of 23 CEC'2009–recommended multiobjective benchmark functions [30]. Among these benchmarks, seven (UF1–UF7) are two-objective, three (UF8–UF10) are three-objective, and three (UF11–UF13) are five-objective unconstrained (bound-constrained) test functions. UF11–UF13 are the extended and rotated versions of two immensely popular test suites—the Deb–Thiele–Laumanns–Zitzler test suites [31,32]—and one test function from the Walking Fish Group test suite [33]. The set of 23 benchmark functions also includes seven (CF1–CF7) two-objective and three (CF8–CF10) three-objective general constrained test instances.

5.2. Noise models

The noisy version of the k th objective $f_k(\vec{X})$, of any trial solution \vec{X} , is given by

$$f_{k\text{-noisy}}(\vec{X}) = f_k(\vec{X}) + \eta_k \quad \text{for } k = [1, N], \quad (15)$$

where η_k represents the injected stochastic noise amplitude that follows a certain probability distribution function (PDF). The following five variants of η_k are considered here:

- (a) **Gaussian:** η_k has a **Gaussian PDF**, which is given by

Table 2Parameter settings in D -dimensional search space.

MOEA-RF		Modified NSGA-II		NT-SPEA		ParEGO		eIEMAS	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Archive size	50	Crossover probability	0.8	Death threshold	0	μ	60	Initial population in Latin hypercube	$11D-1$
Selection operator	Binary tournament	Mutation probability	$1/D$	Reproduction threshold	50	λ	60	Number of scalarizing vectors for two objectives	11
Crossover operator	Uniform	Initial value of neighborhood restriction factor	10	Migration threshold	20	Archive size for two objectives	20	Number of scalarizing vectors for three objectives	15
Crossover rate	0.8			Niche size	0.2	Archive size for three objectives	50	Scalarizing function	Augmented Tchebycheff
Mutation operator	Bit flip			Niche adaptation factor	0.99975	Recombination operator		Internal GA evaluations per iteration	200,000
Mutation rate	$1/D$			Crowding factor	2	Mutation operator		Crossover probability	0.1
Ranking scheme	Pareto ranking			Migration chance	0.2	c_1	0.1	Real-value mutation probability	$1/D$
Diversity operator	Niche count with radius 0.01 in Normalized Objective Space			Reproduction chance	0.55	c_2	0.3	Real-value SBX parameter	10
				Energy transferred	0/100	Maximum lifetime	4	Real-value mutation parameter	50

eIEMAS, elitist evolutionary multiagent system; GA, genetic algorithm; MOEA-RF, multiobjective evolutionary algorithm with robust features; NSGA-II, non-dominated sorting genetic algorithm II; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SBX, simulated binary crossover.

$$\text{pdf}(\eta_k) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\eta_k-m)^2}{2\sigma^2}}, \quad (16)$$

where m and σ^2 stand for the mean and variance of the Gaussian PDF. We have used a well-known technique called the Box–Muller method [34] to inject η_k into $f_k(\vec{X})$ following the Gaussian distribution.

(b) **Poisson:** η_k has a **Poisson PDF**, which is given by

$$\text{pdf}(\eta_k) = \frac{\lambda^{\eta_k} \times e^{-\lambda}}{\eta_k!}, \quad (17)$$

where λ represents the mean (as well as the variance) in the Poisson distribution. The injection of Poisson noise η_k on $f_k(\vec{X})$ is performed here using Knuth's algorithm [35].

(c) **Rayleigh:** η_k has a **Rayleigh PDF**, which is given by

$$\text{pdf}(\eta_k) = \begin{cases} \frac{\eta_k}{b^2} \exp(-\eta_k^2/(2b^2)) & \text{if } \eta_k \geq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (18)$$

where the mean and the variance of the noise distribution are $m = b\sqrt{\pi/2}$ and $\sigma^2 = b^2(4 - \pi)/2$, respectively. The injection of Rayleigh noise η_k on $f_k(\vec{X})$ is performed here using inverse transform sampling [36].

(d) **Exponential:** η_k has an **exponential PDF**, which is given by

$$\text{pdf}(\eta_k) = \begin{cases} a \exp(-a\eta_k) & \text{if } \eta_k \geq 0, \\ 0 & \text{otherwise,} \end{cases} \quad (19)$$

where the mean and the variance of the noise distribution are $m = 1/a$ and $\sigma^2 = 1/a^2$, respectively. We have used the Ziggurat method [37] to inject η_k into $f_k(\vec{X})$ following an exponential distribution.

- (e) **Random:** Lastly, we consider η_k to be **random** noise with maximum noise amplitude within $\pm 25\%$ of the true fitness amplitudes $f_k(\vec{X})$. Linear congruential pseudorandom number generator code [38] is used to generate the random noise.

5.3. Performance metrics

- (a) **Inverted generational distance (IGD):** Let P^* be a set of uniformly distributed points along the true optimal Pareto front and let A be an approximate set to the optimal Pareto front, both defined in the objective space of a MOO problem. Then, the average distance from P^* to A [30] may be defined as follows:

$$\text{IGD}(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}. \quad (20)$$

Here $d(v, A)$ represents the minimum Euclidean distance between v and the points in A . A lower value of IGD ensures that the approximate Pareto front A , obtained by the proposed MOO, is very close to the optimal Pareto front.

- (b) **Spacing (Δ):** Schott [39] proposed a metric to measure “the range variance of the neighboring vectors of the nondominated solutions” obtained by the algorithm in question. The metric provides a distinctive measure of the spread and distribution of the vectors. The metric is defined as follows:

$$\Delta = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (\bar{d} - d_i)^2}, \quad \bar{d} = \frac{1}{M} \sum_{i=1}^M d_i. \quad (21)$$

Here $d_i = \min_{j=1, j \neq i}^M \sum_{k=1}^N |\bar{f}_k(\vec{X}_i) - \bar{f}_k(\vec{X}_j)|$ with M as the nondominated vectors found by the method. Here \vec{X}_i and \vec{X}_j are nondominated vectors belonging to the approximate Pareto front A . A value of zero for this metric signifies all the members of the approximate Pareto front are equidistantly spaced.

- (c) **Error ratio (ER):** This metric was introduced by van Veldhuizen [40] and is defined as follows:

$$\text{ER} = \frac{\sum_{i=1}^M e_i}{M}, \quad e_i = \begin{cases} 0, & \text{if } \vec{X}_i \in A \text{ and } \vec{X}_i \in P^*, \\ 1, & \text{if } \vec{X}_i \in A \text{ and } \vec{X}_i \notin P^*. \end{cases} \quad (22)$$

An ideal value of zero for this metric designates that all the nondominated solutions in the approximate Pareto front A belong to the optimal Pareto front P^* .

- (d) **Hypervolume ratio (HVR):** This metric was proposed by Coello et al. [41], and is defined as follows.

$$\text{HVR}(A) = \frac{\text{HV}(A)}{\text{HV}(P^*)}. \quad (23)$$

Here, $\text{HV}(A)$ and $\text{HV}(P^*)$ denote the hypervolume of the approximate Pareto front A and optimal Pareto front P^* , respectively. The size of the objective spaces covered by a set of nondominated solutions S is termed as its hypervolume $\text{HV}(S)$. Mathematically,

$$\text{HV}(S) = \Lambda \left(\bigcup_{\vec{X} \in S} \{ \vec{X}' | \vec{X} < \vec{X}' < \vec{X}_{\text{ref}} \} \right). \quad (24)$$

Here Λ symbolizes the Lebesgue measure. \vec{X}_{ref} denotes the reference point with the lowest fitness (i.e., the maximum objective function value in the case of a minimization problem). $\text{HVR}(A)$ attains its maximum (ideal) value of 1 [42] provided the nondominated vectors belonging to A (in the objective spaces) are identical to the members of the optimal Pareto front P^* .

However, under the circumstances of noisy fitness landscapes, there is a need to address the uncertainty related to the fitness measurements while determining the hypervolume of the approximate Pareto front A . The uncertainty problem is resolved here by weighting the size of the objective space covered by a nondominated solution \vec{X} with the probability that \vec{X} dominates at least one population member. The higher the probability, the more reliable the measure of the objective space covered by \vec{X} . For a MOO problem with all objectives to be minimized, the probability of a trial solution \vec{X} dominating another solution \vec{X}_j is given in (25):

$$p(\vec{X} < \vec{X}_j) = \prod_{k=1}^N p(\bar{f}_k(\vec{X}_j) > \bar{f}_k(\vec{X})) = \prod_{k=1}^N \frac{1}{1 + \exp(-c(\bar{f}_k(\vec{X}_j) - \bar{f}_k(\vec{X})))}. \quad (25)$$

The validation of the above probability distribution is evident from the following observations:

1. If c approaches ∞ and $\bar{f}_k(\bar{X}_j) > \bar{f}_k(\bar{X})$ for $k = [1, N]$, then $p(\bar{X} < \bar{X}_j) = 1$, indicating the dominance of \bar{X} over \bar{X}_j .
2. If c approaches ∞ and $\bar{f}_k(\bar{X}_j) < \bar{f}_k(\bar{X})$ for $k = [1, N]$, then $p(\bar{X} < \bar{X}_j) = 0$, indicating the dominance of \bar{X}_j over \bar{X} .
3. If $\bar{f}_k(\bar{X}_j) \cong \bar{f}_k(\bar{X})$ for $k = [1, N]$, then $p(\bar{X} < \bar{X}_j) = 1/2^N$, indicating the nondominance relationship between \bar{X} and \bar{X}_j .

It is apparent that the probability that at least one population member is dominated by \bar{X} is given by (26):

$$p_{\bar{X}} = 1 - \prod_{\substack{j=1 \\ \bar{X}_j \neq \bar{X}}}^{NP} p(\bar{X}_j < \bar{X}). \quad (26)$$

Hence the hypervolume of the approximate front A can now be evaluated using (24) with the objective space covered by any trial solution \bar{X} being weighted by $p_{\bar{X}}$ for all $\bar{X} \in A$.

6. Simulation results

This section provides the relative performance of the individual extensions of DEMO introduced before through computer simulations. A comparative analysis of the performance of DENMO and the performance of a few standard algorithms is also undertaken in this section.

6.1. Effectiveness of different extensions of DENMO

In Section 4, we extended DEMO by introducing three strategies—namely, adaptive selection of sample size (ASSS), SDFE, and PS—to suit its application in noisy MOO problems. The ASSS, SDFE, and PS strategies are mutually independent as they deal with three different issues in the context of noisy optimization. To be more specific, while the ASSS strategy is concerned with the sample size adaptation, the SDFE strategy deals with expected fitness of trial solutions based on the distribution of fitness samples, and the PS strategy takes care of relative comparison of trial solutions of the same rank to ensure diversity as well as quality of the solutions in the population. Naturally, incorporation of any one of the three strategies into DEMO does not necessarily require exclusion of the rest. The following groups of strategies are considered below for the experiments:

- (a) **DEMO-ASSS:** Traditional DEMO is extended here with the ASSS strategy only. The mean value of the fitness samples is used as the fitness estimate of a given trial solution. The crowding-distance-induced selection scheme is employed for the truncation of the extended population.
- (b) **DEMO-SDFE:** Here, the SDFE strategy is integrated with the traditional DEMO with equal sample size (size 20) being assigned to all the trial solutions for their fitness reevaluation. Here also the crowding-distance-induced selection scheme is employed for selection of quality solutions from the same-rank front.
- (c) **DEMO-PS:** The performance of traditional DEMO is improved here by the PS approach. Here also the sample size of all the trial solutions is fixed at 20 to estimate their mean fitness.
- (d) **DEMO-ASSS-SDFE:** Traditional DEMO here utilizes the benefits of both the ASSS strategy and the SDFE strategy. Truncation of the extended population using crowding distance sorting of traditional DEMO remains unchanged.
- (e) **DEMO-ASSS-PS:** The extension here encompasses the ASSS and PS strategies to govern the diversity and quality of the solutions in the population.
- (f) **DEMO-SDFE-PS:** Traditional DEMO is extended here with SDFE and PS. The same sample size (size 20) is assigned to each trial solution.
- (g) **DENMO:** All three extensions are integrated into traditional DEMO.

The mean and the standard deviation of the best-of-run IGD metric values for 50 independent runs of each of the seven variants considered in this section, along with their traditional counterpart DEMO, are presented in Table 3 for 23 30-dimensional CEC'2009-recommended multiobjective benchmark functions [30], each contaminated with random noise of restricted amplitude. To obtain the results shown in Table 3, all algorithms were run from the same initial population in every run. The relative ranking of different extensions was also examined with respect to the remaining performance metrics with different noise settings. The results obtained follow a trend similar to those reported in Table 3; however, they are not given here because of space restrictions. Friedman two-way analysis of variances by ranks [43], a nonparametric statistical test, was also performed on the mean of the IGD metric values for 50 independent runs of each variant as reported in Table 3. The null hypothesis here states that all the algorithms are equivalent, so their individual ranks should be equal. The last row in Table 3 summarizes the rankings obtained by the Friedman procedure. It is evident that DENMO outperforms the other extended versions of the traditional DEMO in handling noise in fitness landscapes. With a level of significance $\alpha = 0.05$, the Friedman statistic exhibits significant differences in the performance of the competing algorithms with a test value of 161. The results highlight DENMO as the best algorithm, so post hoc analysis [44] was applied with DENMO as the control method. Although we provide the results for the IGD metric only, the trend remains the same for each of the

Table 3

Ranking of different extensions introduced in traditional differential evolution for multiobjective optimization (DEMO) with respect to the mean inverted generational distances (with the standard deviation in parentheses) after contaminating the benchmark functions with random noise of amplitude within $\pm 25\%$ of the true objective function values for UF1–UF8.

Functions	Groups							
	DEMO	DEMO-ASSS	DEMO-SDFE	DEMO-PS	DEMO-ASSS-SDFE	DEMO-ASSS-PS	DEMO-SDFE-PS	DENMO
UF1	0.7420 (0.358)	0.6573 (0.238)	0.3269 (0.181)	0.4537 (0.223)	0.0937 (0.143)	0.2330 (0.171)	0.0216 (0.134)	0.0176 (0.119)
UF2	0.7622 (0.405)	0.6289 (0.374)	0.3913 (0.261)	0.4639 (0.305)	0.1239 (0.096)	0.3122 (0.183)	0.0827 (0.049)	0.0379 (0.009)
UF3	0.7631 (0.320)	0.6912 (0.303)	0.5347 (0.296)	0.5727 (0.297)	0.2901 (0.070)	0.4948 (0.272)	0.2014 (0.042)	0.0701 (0.039)
UF4	0.6940 (0.389)	0.6336 (0.357)	0.3279 (0.271)	0.3653 (0.281)	0.1246 (0.065)	0.2517 (0.266)	0.1229 (0.061)	0.0396 (0.014)
UF5	0.6252 (0.425)	0.5218 (0.299)	0.4180 (0.210)	0.5153 (0.218)	0.1289 (0.102)	0.3933 (0.166)	0.0938 (0.072)	0.0728 (0.048)
UF6	0.6812 (0.476)	0.6280 (0.369)	0.2151 (0.317)	0.3092 (0.368)	0.0500 (0.103)	0.1697 (0.261)	0.0376 (0.065)	0.0092 (0.008)
UF7	0.7887 (0.375)	0.6967 (0.306)	0.3192 (0.286)	0.3537 (0.300)	0.0711 (0.156)	0.2324 (0.268)	0.0434 (0.091)	0.0324 (0.025)
UF8	0.7731 (0.354)	0.6335 (0.346)	0.4417 (0.325)	0.5440 (0.340)	0.0764 (0.177)	0.3736 (0.315)	0.0629 (0.126)	0.0566 (0.024)
UF9	0.6856 (0.301)	0.5224 (0.260)	0.4950 (0.241)	0.5107 (0.252)	0.2015 (0.089)	0.4675 (0.168)	0.0339 (0.056)	0.0072 (0.010)
UF10	0.6257 (0.420)	0.5461 (0.369)	0.4255 (0.293)	0.4847 (0.336)	0.0895 (0.106)	0.3896 (0.290)	0.0733 (0.073)	0.0431 (0.070)
UF11	0.7527 (0.271)	0.6601 (0.264)	0.5588 (0.153)	0.5592 (0.256)	0.0619 (0.072)	0.5140 (0.149)	0.0510 (0.035)	0.0359 (0.018)
UF12	0.6228 (0.468)	0.5059 (0.325)	0.4576 (0.298)	0.4786 (0.320)	0.0924 (0.212)	0.4485 (0.250)	0.0822 (0.153)	0.0772 (0.008)
UF13	0.7555 (0.412)	0.6794 (0.321)	0.5144 (0.214)	0.5180 (0.320)	0.2565 (0.091)	0.4795 (0.180)	0.1643 (0.034)	0.0603 (0.011)
CF1	0.6497 (0.416)	0.5525 (0.375)	0.3862 (0.354)	0.4493 (0.359)	0.1442 (0.213)	0.2717 (0.313)	0.1439 (0.026)	0.0606 (0.007)
CF2	0.6436 (0.389)	0.5473 (0.375)	0.2837 (0.164)	0.3391 (0.246)	0.0733 (0.055)	0.2058 (0.141)	0.0704 (0.042)	0.0640 (0.016)
CF3	0.7329 (0.387)	0.5490 (0.378)	0.4863 (0.272)	0.5306 (0.306)	0.3027 (0.134)	0.4588 (0.270)	0.2040 (0.118)	0.0785 (0.097)
CF4	0.8125 (0.307)	0.6536 (0.300)	0.3413 (0.241)	0.5382 (0.287)	0.1312 (0.204)	0.2772 (0.233)	0.0768 (0.164)	0.0260 (0.033)
CF5	0.5903 (0.374)	0.4875 (0.354)	0.4269 (0.181)	0.4713 (0.290)	0.0911 (0.123)	0.3162 (0.165)	0.0877 (0.087)	0.0415 (0.050)
CF6	0.5854 (0.373)	0.5060 (0.337)	0.3718 (0.308)	0.4422 (0.313)	0.0761 (0.151)	0.2961 (0.291)	0.0646 (0.017)	0.0054 (0.003)
CF7	0.5632 (0.381)	0.4922 (0.331)	0.2215 (0.269)	0.2222 (0.306)	0.1177 (0.094)	0.1757 (0.249)	0.0994 (0.089)	0.0690 (0.070)
CF8	0.5344 (0.373)	0.4947 (0.308)	0.3904 (0.195)	0.4357 (0.222)	0.1291 (0.039)	0.3890 (0.077)	0.1163 (0.039)	0.0541 (0.017)
CF9	0.7763 (0.386)	0.6915 (0.251)	0.4344 (0.156)	0.4788 (0.204)	0.1804 (0.071)	0.2816 (0.135)	0.1193 (0.040)	0.0517 (0.021)
CF10	0.6036 (0.416)	0.5197 (0.370)	0.4798 (0.357)	0.5306 (0.366)	0.2059 (0.172)	0.3714 (0.324)	0.1127 (0.156)	0.1080 (0.055)
Friedman ranking	8	7	5	6	3	4	2	1

The best metric value obtained in each case is shown in bold.

ASSS, adaptive selection of sample size; DENMO, differential evolution for noisy multiobjective optimization; PS, probabilistic selection; SDFE, sample-distribution-based fitness estimation.

remaining performance metrics—that is, Δ , ER, and HVR—and for all five variants of the stochastic noise. In the post hoc analysis, the Holm test [44] was employed on the results of the Friedman procedure with DENMO as the control algorithm, and the results are given in Table 4. The null hypothesis here considers that the performance of DENMO and any of the seven competitor algorithms is equally good. However, the outcome of the analysis indicates that only for DEMO-SDFE-PS the null hypothesis cannot be rejected with any of the benchmark functions. The performance of DENMO, however, may be considered as significantly better than that of the other six variants in the present context.

6.2. Performance analysis of DENMO

The comparative analysis of the relative performance of the proposed DENMO algorithm and its competitors is discussed in this section. Although all the experiments were performed for all five variants of noise with noise variance $\sigma^2 \in [0, 1]$ and

Table 4

Holm test applied on results in Table 3 with differential evolution for noisy multiobjective optimization (DENMO) as the control algorithm.

$k - i$	Variants of DEMO	z	p	$\alpha/(k - i), \alpha = 0.05$	Acceptance/rejection of null hypothesis
7	Traditional DEMO	9.69106	0.000010	0.00714	Reject
6	DEMO-ASSS	8.30662	0.000010	0.00833	Reject
5	DEMO-PS	6.92218	0.000010	0.01000	Reject
4	DEMO-SDFE	5.53774	0.000010	0.01250	Reject
3	DEMO-ASSS-PS	4.15331	0.000033	0.01666	Reject
2	DEMO-ASSS-SDF	2.76887	0.005626	0.02500	Reject
1	DEMO-SDFE-PS	1.38443	0.166236	0.05000	Accept

k is the number of extended versions of DEMO = 8 and i is the rank of each extended version in descending (or in ascending) order of z values (or p values).

DEMO, differential evolution for multiobjective optimization; ASSS, adaptive selection of sample size; PS, probabilistic selection; SDFE, sample-distribution-based fitness estimation.

also for 10-, 30-, and 50-dimensional problems, we report here the results for finite values of σ^2 , and for specific problem dimensions to save space. The results omitted follow a trend similar to that in the tables and figures in this section.

6.2.1. Comparative performance of algorithms with respect to the IGD metric

The mean and standard deviation of the IGD metric for 50 independent runs (each with 300,000 FEs for a 30-dimensional problem) of each of the ten algorithms are presented in Table 5 with η_k as the Poisson noise (mean 0.25 and variance 0.25).

Since all the algorithms commence with the same initial population for each problem instance, we use paired t tests to compare the means of the results produced by the best and the second-best algorithms [45]. The statistical significance level for the difference of the means of the best two algorithms is presented in the 12th column in Tables 5–8. Here a plus sign designates that the t value of 49 degrees of freedom is significant at the 0.05 level of significance by a two-tailed test, a minus sign denotes that the difference of means is not statistically significant, and “not applicable” refers to the cases for which two or more algorithms achieve the same best accuracy results. The statistical significance hence refers to the comparison of DENMO with the best of the remaining nine algorithms.

The simulation results in Table 5 show that DENMO outperformed its competitors for 21 of the 23 benchmark functions. Of these 21, for 20 functions the difference between the mean of the IGDs of DENMO and its nearest competitor is statistically significant. DEMON, which remains the second-best algorithm, achieves the best average IGD, outperforming DENMO in one case (UF7).

Plots of the final approximation sets with the smallest IGD in the objective spaces for some test instances with two- and three-objectives are shown in Fig. 4. It is evident from Fig. 4 that the approximate Pareto front found by the DENMO algorithm is closer to the optimal Pareto front in comparison with the other competing algorithms.

6.2.2. Comparative performance of algorithms with respect to the Δ metric

The mean and standard deviation of the Δ metric for 50 independent runs (each with 500,000 FEs for a 50-dimensional problem) of each of the ten algorithms are presented in Table 6 with η_k as the Rayleigh noise (mean 0.3 and variance 0.025). Table 6 indicates that DENMO outperformed all the contender algorithms in a statistically significant fashion for 20 functions with respect to the Δ metric. It remains the second-best algorithm for the benchmark function UF11, being outperformed by DEMON alone. In the case of the benchmark functions UF13 and CF7, the performance of DENMO remains comparable to that of DEMON. However, it is noteworthy that for UF13 DENMO achieves the lowest standard deviation once again.

6.2.3. Comparative performance of algorithms with respect to the ER metric

The mean and standard deviation of the ER metric for 50 independent runs (each with 300,000 FEs for a 30-dimensional problem) of each of the ten algorithms are presented in Table 7 with η_k as exponential noise (mean 0.86 and variance 0.75). Close scrutiny of Table 7 reveals that DENMO outperformed all nine evolutionary/swarm noisy MOO algorithms in a statistically significant fashion for 19 of the 23 test functions in achieving near optimal values of the ER metric. It yielded results statistically equivalent to those of DEMON in the case of benchmark functions UF3 and CF2. It achieved the second-best rank among the contender algorithms for benchmarks UF7 and UF13.

6.2.4. Comparative performance of algorithms with respect to the HVR metric

The mean and standard deviation of the HVR metric for 50 independent runs (each with 300,000 FEs for a 30-dimensional problem) of each of the ten algorithms are presented in Table 8 with η_k as Gaussian noise (mean 0 and variance 0.4). Close inspection of Table 8 indicates that the performance of the proposed DENMO algorithm remained consistently superior to that of the other noisy MOO algorithms with respect to the HVR metric. Of the 23 benchmark functions, in 20 cases DENMO outperforms its nearest-neighbor competitor in a statistically significant fashion.

6.2.5. Effect of varying the noise variance

In this subsection, we scrutinize the effect of varying the noise variance on the performance of all the algorithms. Fig. 5 shows the evolution of the average Δ and ER metric values of the population with the noise variance for all ten algorithms

Table 5

Mean inverted generational distances and standard deviation (in parentheses) for 50 independent runs with Poisson noise with mean and variance of 0.25.

Function	DENMO	DEMON	NSGA-II-A	CDR	SA	eEMAS	MOEA-RF	Mod. NSGA-II	NT-SPEA	ParEGO	Statistical significance
UF1	0.0957 (0.044)	0.1102 (0.091)	0.1202 (0.139)	0.1522 (0.206)	0.2138 (0.265)	0.2968 (0.279)	0.3807 (0.312)	0.4524 (0.314)	0.4629 (0.331)	0.5751 (0.371)	+
UF2	0.0795 (0.124)	0.1391 (0.169)	0.1573 (0.201)	0.2998 (0.262)	0.3056 (0.282)	0.4045 (0.297)	0.4235 (0.321)	0.4523 (0.373)	0.4725 (0.403)	0.5296 (0.408)	+
UF3	0.0780 (0.030)	0.1732 (0.051)	0.2081 (0.053)	0.313 (0.106)	0.3564 (0.177)	0.4183 (0.196)	0.4224 (0.217)	0.4235 (0.253)	0.5692 (0.349)	0.6150 (0.481)	+
UF4	0.0612 (0.072)	0.0906 (0.127)	0.1254 (0.168)	0.3431 (0.190)	0.4100 (0.232)	0.4549 (0.252)	0.4985 (0.284)	0.5041 (0.325)	0.6091 (0.359)	0.6352 (0.417)	+
UF5	0.0286 (0.009)	0.1564 (0.095)	0.2314 (0.131)	0.2542 (0.160)	0.2850 (0.189)	0.4402 (0.205)	0.4430 (0.208)	0.4535 (0.373)	0.4743 (0.398)	0.4863 (0.430)	+
UF6	0.1476 (0.075)	0.2238 (0.081)	0.2578 (0.096)	0.2711 (0.197)	0.2997 (0.249)	0.4587 (0.285)	0.5159 (0.372)	0.5401 (0.418)	0.5524 (0.428)	0.6487 (0.480)	+
UF7	0.0691 (0.071)	0.0556 (0.026)	0.0843 (0.240)	0.1971 (0.271)	0.3065 (0.282)	0.3305 (0.337)	0.4200 (0.395)	0.4634 (0.412)	0.4889 (0.426)	0.5110 (0.450)	–
UF8	0.0003 (0.040)	0.1572 (0.119)	0.3309 (0.278)	0.3437 (0.284)	0.3641 (0.303)	0.4227 (0.320)	0.5529 (0.323)	0.5972 (0.353)	0.6824 (0.358)	0.6831 (0.409)	+
UF9	0.0116 (0.015)	0.0827 (0.024)	0.3314 (0.059)	0.3970 (0.093)	0.3979 (0.101)	0.5263 (0.175)	0.5905 (0.237)	0.6097 (0.267)	0.6354 (0.298)	0.6723 (0.305)	+
UF10	0.0274 (0.060)	0.0944 (0.099)	0.1218 (0.101)	0.1324 (0.131)	0.1816 (0.202)	0.3216 (0.237)	0.3453 (0.274)	0.3471 (0.310)	0.4093 (0.413)	0.4480 (0.423)	+
UF11	0.0350 (0.030)	0.0552 (0.064)	0.0712 (0.086)	0.0957 (0.193)	0.1121 (0.248)	0.3865 (0.380)	0.4183 (0.416)	0.4909 (0.417)	0.6274 (0.453)	0.6387 (0.477)	+
UF12	0.0172 (0.026)	0.0211 (0.057)	0.2025 (0.085)	0.2065 (0.160)	0.2281 (0.168)	0.3201 (0.216)	0.4442 (0.321)	0.3831 (0.261)	0.5772 (0.435)	0.6436 (0.479)	+
UF13	0.0607 (0.050)	0.1078 (0.085)	0.1743 (0.101)	0.2178 (0.162)	0.2510 (0.227)	0.2815 (0.253)	0.4281 (0.327)	0.4627 (0.361)	0.4955 (0.438)	0.6053 (0.442)	+
CF1	0.0432 (0.006)	0.0811 (0.045)	0.1115 (0.049)	0.1287 (0.095)	0.1467 (0.149)	0.1815 (0.161)	0.3401 (0.214)	0.3615 (0.221)	0.4457 (0.435)	0.5417 (0.482)	+
CF2	0.0428 (0.023)	0.0428 (0.023)	0.0558 (0.052)	0.2053 (0.163)	0.0620 (0.142)	0.2059 (0.258)	0.3490 (0.292)	0.4357 (0.323)	0.5257 (0.361)	0.5366 (0.400)	NA
CF3	0.0390 (0.024)	0.1825 (0.050)	0.2512 (0.062)	0.3527 (0.266)	0.3334 (0.235)	0.4388 (0.294)	0.4482 (0.356)	0.4624 (0.359)	0.4952 (0.431)	0.4944 (0.387)	+
CF4	0.0461 (0.041)	0.0560 (0.113)	0.0698 (0.177)	0.0743 (0.282)	0.0835 (0.317)	0.4054 (0.356)	0.4327 (0.358)	0.4573 (0.363)	0.5052 (0.420)	0.5968 (0.470)	+
CF5	0.0567 (0.044)	0.1347 (0.048)	0.1349 (0.073)	0.2408 (0.111)	0.2418 (0.115)	0.2543 (0.142)	0.3917 (0.234)	0.4488 (0.257)	0.5057 (0.311)	0.5171 (0.379)	+
CF6	0.0674 (0.053)	0.0722 (0.090)	0.1875 (0.129)	0.2607 (0.132)	0.2782 (0.192)	0.3882 (0.236)	0.4469 (0.317)	0.4876 (0.372)	0.5045 (0.386)	0.6203 (0.452)	+
CF7	0.0228 (0.003)	0.0364 (0.044)	0.0768 (0.052)	0.1935 (0.061)	0.2633 (0.065)	0.2769 (0.205)	0.3738 (0.257)	0.4127 (0.306)	0.4269 (0.317)	0.4736 (0.350)	+
CF8	0.0646 (0.037)	0.0931 (0.080)	0.1286 (0.102)	0.1103 (0.089)	0.1426 (0.152)	0.1645 (0.270)	0.2074 (0.301)	0.2081 (0.342)	0.4610 (0.443)	0.5853 (0.479)	+
CF9	0.0472 (0.078)	0.0985 (0.142)	0.1645 (0.170)	0.1087 (0.162)	0.2532 (0.257)	0.2566 (0.289)	0.2567 (0.332)	0.3960 (0.367)	0.4365 (0.403)	0.6271 (0.422)	+
CF10	0.0253 (0.116)	0.0285 (0.153)	0.2261 (0.317)	0.1882 (0.222)	0.2658 (0.359)	0.2669 (0.360)	0.2700 (0.368)	0.2846 (0.370)	0.3511 (0.453)	0.4670 (0.464)	–

The best metric value obtained in each case is shown in bold. A plus sign designates that the t value of 49 degrees of freedom is significant at the 0.05 level of significance by a two-tailed test, and a minus sign denotes that the difference of means is not statistically significant.

CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eEMAS, elitist evolutionary multiagent system; MOEA-RF, multiobjective evolutionary algorithm with robust features; Mod. NSGA-II, modified nondominated sorting genetic algorithm II; NA, not applicable (referring to the cases for which two or more algorithms achieve the same best accuracy results); NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

number when the number of generations is fixed at 300,000 (for problem dimension $D = 30$) for Poisson and exponential noise, respectively. It is evident from Fig. 5 that all the noisy MOO algorithms eventually lose their accuracy in achieving values of Δ and ER close to the ideal value (i.e., zero) as noise of increasing variance creeps into the fitness landscapes. However, DENMO appears to be most effective in achieving an approximate Pareto front with lower values of the Δ and ER metrics, even when noise is a predominant factor and the fitness landscape itself is very complex.

6.2.6. Effect of varying the problem dimension

Plots of the average HVR and IGD metric values against the problem dimension (within [10, 100]) are given in Fig. 6 for Rayleigh and random distribution of noise, respectively. We note from Fig. 6 that the HVR and IGD metrics are nonincreasing and nondecreasing functions, respectively, of the problem dimension D for specific noise settings. An intuitive interpretation of this phenomenon is that with increase in D , a more complex terrain needs to be explored by the population members to obtain the Pareto optimal solutions, thereby decreasing the HVR and increasing the IGD metric values. However, for all

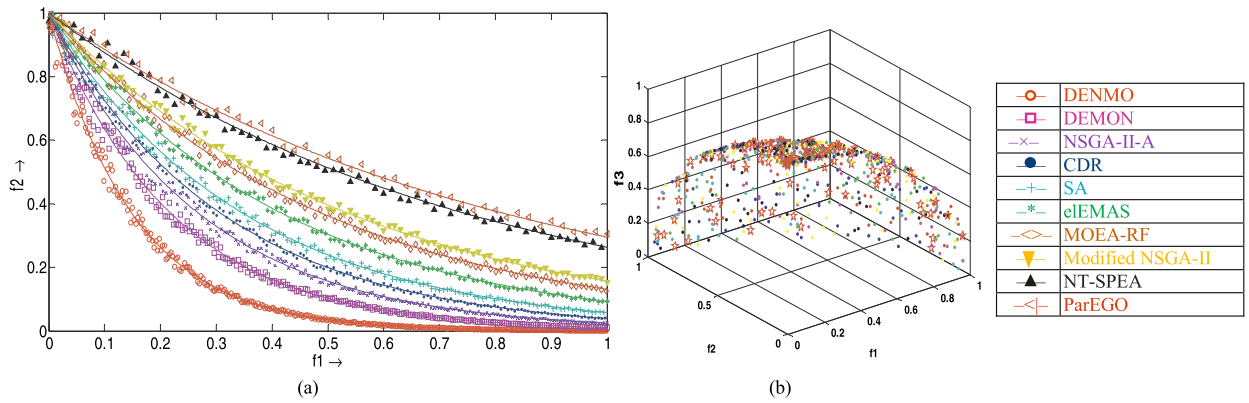


Fig. 4. The final approximation set with the smallest inverted generational distance in the objective space (a) for UF3 contaminated with zero mean Gaussian noise ($\sigma^2 = 0.6$) and (b) for UF10 contaminated with Poisson noise ($\sigma^2 = 1.0$), with the corresponding symbols for the ten algorithms. CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eEMAS, elitist evolutionary multiagent system; MOEA-RF, multiobjective evolutionary algorithm with robust features; NSGA-II, nondominated sorting genetic algorithm II; NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

instances DENMO achieves the best rank among the contestants by providing the largest and smallest values of the HVR and IGD metrics, as substantiated by Fig. 6.

6.2.7. Convergence characteristics

To compare the relative speed of convergence and the quality of the solution of DENMO and the other competitors, the Δ and ER metric values of the median run of each algorithm versus the number of FEs are plotted in Fig. 7. Fig. 7 shows that DENMO outperformed all the contender algorithms.

6.2.8. Robustness and runtime analysis

The latter part of the experiment compared the consistent performance and the speed of different algorithms with respect to the HVR metric. First, a threshold value of the HVR metric was selected corresponding to each benchmark function. Then each of the ten algorithms was run on each benchmark function (for different settings of noise and problem dimension). The algorithm was terminated if either the best value of the HVR metric achieved by the algorithm fell above the predefined threshold or if the maximum number of FEs (300,000 for 30-D problems and 500,000 for 50-D problems) was reached, whichever occurred earlier. The termination of an algorithm because of the occurrence of the first condition indicates that the algorithm succeeds in finding the approximate Pareto front within the prescribed tolerance limit of the HVR metric in that specific run (without reaching the maximum number of FEs) and hence it is said to be a successful run. The number of such successful runs out of a total of 50 runs was recorded for each algorithm with respect to the predefined threshold value of the HVR metric. A lower number of unsuccessful runs (i.e., total runs minus successful runs) of an algorithm corresponds to robustness in its performance. In Fig. 8, we present a plot of number of unsuccessful runs versus the runtime in terms of the expected number of FEs over 50 independent runs, with respect to the HVR tolerance limit of 0.75 for each of the ten algorithms for UF5 (tested for different settings of the problem dimension and stochastic noise). The plot provides a visual means of elucidating the efficacy of all algorithms with respect to both robustness and runtime (in terms of the expected number of FEs). The x and y coordinates are scaled properly to have uniformity in magnitude. The performance of an algorithm was then assessed by measuring the distance of the representative point from the origin. The smaller the measure, the better is the performance of the algorithm in terms of both robustness and expected runtime. The rankings thus obtained for different dimensional problems are given in Table 9. The experiment was repeated for all benchmark functions with different settings of stochastic noise and the problem dimension with respect to all three remaining metrics. The results omitted for economy of space follow a trend similar to that in Fig. 8 and Table 9. It is observed from Fig. 8 that DEMON, MOEA-RF, and simulated annealing in the case of a 30-D problem (with zero mean Gaussian noise of $\sigma^2 = 0.35$) and DENMO, NSGA-II-A, and CDR in the case of a 50-D problem (with Poisson noise of $\sigma^2 = 0.7$) manage to attain a smaller number of unsuccessful runs than DENMO but at the cost of increased expected runtime.

6.2.9. Nonparametric statistical analysis of performance

A series of nonparametric statistical tests were also performed on the mean of all four performance metrics for 50 independent runs of each of the ten competitor algorithms as reported in Tables 5–8. The first test was the Friedman two-way analysis of variances by ranks [43]. Additionally, we used the Iman-Davenport test as a variant of the Friedman test [46]. The objective of the Friedman and Iman-Davenport tests is to show that there is a significant statistical difference between different algorithms. Table 10 summarizes the rankings obtained by the Friedman procedure, highlighting DENMO as the best algorithm. With a level of significance α of 0.05, both the Friedman statistics and the Iman-Davenport statistics

Table 6Mean spacing (Δ) and standard deviation (in parentheses) for 50 independent runs with Rayleigh noise with mean 0.3 and variance 0.025.

Function	DENMO	DEMON	NSGA-II-A	CDR	SA	eEMAS	MOEA-RF	Mod. NSGA-II	NT-SPEA	ParEGO	Statistical significance
UF1	0.0622 (0.019)	0.0669 (0.063)	0.1033 (0.080)	0.1105 (0.276)	0.1340 (0.284)	0.2926 (0.357)	0.4206 (0.369)	0.4305 (0.394)	0.5202 (0.426)	0.5649 (0.501)	+
UF2	0.0122 (0.033)	0.1077 (0.120)	0.1523 (0.135)	0.1943 (0.231)	0.2378 (0.303)	0.2475 (0.353)	0.3258 (0.395)	0.3898 (0.403)	0.5508 (0.418)	0.5561 (0.484)	+
UF3	0.2042 (0.031)	0.2514 (0.118)	0.2696 (0.186)	0.3254 (0.233)	0.4132 (0.318)	0.4602 (0.332)	0.5078 (0.343)	0.5242 (0.456)	0.5252 (0.500)	0.5484 (0.524)	+
UF4	0.0150 (0.023)	0.0849 (0.046)	0.1100 (0.095)	0.2213 (0.117)	0.2314 (0.139)	0.2516 (0.196)	0.3412 (0.337)	0.4337 (0.380)	0.4388 (0.486)	0.5029 (0.504)	+
UF5	0.0138 (0.015)	0.2229 (0.119)	0.2459 (0.130)	0.2463 (0.144)	0.4025 (0.237)	0.4786 (0.282)	0.4239 (0.279)	0.5137 (0.302)	0.5217 (0.423)	0.5843 (0.519)	+
UF6	0.0121 (0.022)	0.0448 (0.056)	0.0587 (0.072)	0.3578 (0.095)	0.4581 (0.114)	0.5250 (0.186)	0.5931 (0.324)	0.5373 (0.216)	0.6293 (0.329)	0.6626 (0.494)	+
UF7	0.1545 (0.043)	0.1869 (0.216)	0.2130 (0.269)	0.4191 (0.290)	0.4281 (0.307)	0.4852 (0.317)	0.4304 (0.307)	0.5983 (0.378)	0.6228 (0.395)	0.6253 (0.524)	+
UF8	0.0827 (0.005)	0.1433 (0.022)	0.2033 (0.118)	0.2280 (0.199)	0.2333 (0.365)	0.2721 (0.370)	0.4779 (0.405)	0.2993 (0.384)	0.5836 (0.443)	0.6393 (0.485)	+
UF9	0.0645 (0.023)	0.1059 (0.052)	0.1414 (0.052)	0.1609 (0.094)	0.2856 (0.101)	0.3731 (0.164)	0.4587 (0.293)	0.4026 (0.257)	0.4963 (0.406)	0.5590 (0.471)	+
UF10	0.0371 (0.072)	0.0690 (0.180)	0.1428 (0.205)	0.2304 (0.255)	0.2656 (0.325)	0.2934 (0.375)	0.3407 (0.427)	0.3467 (0.470)	0.4275 (0.483)	0.6161 (0.488)	+
UF11	0.2186 (0.172)	0.0845 (0.173)	0.2700 (0.189)	0.3743 (0.217)	0.4952 (0.289)	0.4382 (0.259)	0.5083 (0.326)	0.5648 (0.365)	0.5662 (0.398)	0.6345 (0.512)	–
UF12	0.0204 (0.025)	0.1472 (0.072)	0.1798 (0.126)	0.2280 (0.148)	0.3991 (0.183)	0.4234 (0.237)	0.4544 (0.347)	0.4723 (0.376)	0.5354 (0.384)	0.6097 (0.450)	+
UF13	0.2388 (0.010)	0.2388 (0.022)	0.2476 (0.099)	0.3039 (0.190)	0.3311 (0.308)	0.3407 (0.326)	0.3978 (0.351)	0.5456 (0.355)	0.5659 (0.427)	0.5931 (0.512)	NA
CF1	0.0515 (0.024)	0.0750 (0.049)	0.0935 (0.064)	0.1421 (0.180)	0.2039 (0.387)	0.1650 (0.286)	0.1585 (0.210)	0.2480 (0.418)	0.4004 (0.465)	0.5993 (0.489)	+
CF2	0.0271 (0.150)	0.0321 (0.228)	0.1147 (0.247)	0.3419 (0.295)	0.1782 (0.286)	0.4028 (0.376)	0.4227 (0.421)	0.4237 (0.441)	0.5153 (0.485)	0.5246 (0.518)	+
CF3	0.0897 (0.011)	0.1740 (0.070)	0.1990 (0.150)	0.2344 (0.158)	0.3076 (0.294)	0.3259 (0.314)	0.4183 (0.421)	0.4845 (0.471)	0.5047 (0.494)	0.5097 (0.516)	+
CF4	0.0174 (0.037)	0.2508 (0.108)	0.3008 (0.196)	0.3511 (0.312)	0.3699 (0.344)	0.3885 (0.351)	0.4354 (0.436)	0.4907 (0.447)	0.5955 (0.459)	0.6358 (0.491)	+
CF5	0.2517 (0.078)	0.2581 (0.129)	0.2998 (0.237)	0.4127 (0.239)	0.4683 (0.293)	0.5018 (0.315)	0.5348 (0.412)	0.5778 (0.464)	0.6014 (0.473)	0.6114 (0.481)	+
CF6	0.0800 (0.149)	0.1340 (0.171)	0.1793 (0.205)	0.1856 (0.262)	0.2654 (0.302)	0.4328 (0.320)	0.4387 (0.365)	0.4969 (0.435)	0.5750 (0.439)	0.5862 (0.471)	+
CF7	0.0114 (0.129)	0.0114 (0.129)	0.2681 (0.197)	0.2685 (0.208)	0.2795 (0.209)	0.4132 (0.271)	0.4371 (0.287)	0.4415 (0.295)	0.4989 (0.346)	0.5416 (0.500)	NA
CF8	0.0074 (0.050)	0.0383 (0.077)	0.0534 (0.174)	0.0851 (0.271)	0.1039 (0.291)	0.1912 (0.300)	0.2055 (0.332)	0.2536 (0.452)	0.4579 (0.512)	0.5273 (0.524)	+
CF9	0.0432 (0.100)	0.0474 (0.106)	0.2637 (0.136)	0.2875 (0.265)	0.2916 (0.280)	0.3288 (0.312)	0.4101 (0.322)	0.5473 (0.431)	0.5527 (0.472)	0.5935 (0.490)	+
CF10	0.0995 (0.189)	0.2206 (0.203)	0.2722 (0.224)	0.2887 (0.226)	0.3289 (0.239)	0.3541 (0.365)	0.3944 (0.385)	0.4424 (0.386)	0.4581 (0.408)	0.6146 (0.402)	+

The best metric value obtained in each case is shown in bold. A plus sign designates that the t value of 49 degrees of freedom is significant at the 0.05 level of significance by a two-tailed test, and a minus sign denotes that the difference of means is not statistically significant.

CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eEMAS, elitist evolutionary multiagent system; MOEA-RF, multiobjective evolutionary algorithm with robust features; Mod. NSGA-II, modified nondominated sorting genetic algorithm II; NA, not applicable (referring to the cases for which two or more algorithms achieve the same best accuracy results); NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

show significant differences in the performance of the contender algorithms, with the test values as presented in Table 10 and $p < 0.001$. So the post hoc analysis was performed with DENMO as the control method.

In the post hoc analysis, we applied the Bonferroni–Dunn test [47] to the ranking results of the Friedman procedure. The analysis indicates the level of significance of the superiority of the control algorithm over each of the remaining nine algorithms. For the Bonferroni–Dunn test, a critical difference [44] was calculated, which for these data was 2.2677. The interpretation of this measure is that the performance of the two algorithms is significantly different only if their corresponding average Friedman ranks differ by at least a critical difference, which is depicted in Fig. 9. Fig. 9 shows a bar graph, the height of each bar being proportional to the average Friedman ranking obtained for the representative algorithm. We chose the smallest of them (corresponding to the best/control algorithm, i.e., here DENMO), and we summed its height and the critical difference obtained by the Bonferroni–Dunn test. The result is represented by a cut-line going through all the bars. Then we can conclude that the behaviors of the algorithms characterized by the bars above the cut-line are sig-

Table 7

Mean error ratio and standard deviation (in parentheses) for 50 independent runs with exponential noise with mean 0.86 and variance 0.75.

Function	DENMO	DEMON	NSGA-II-A	CDR	SA	eEMAS	MOEA-RF	Mod. NSGA-II	NT-SPEA	ParEGO	Statistical significance
UF1	0.0363 (0.010)	0.0789 (0.052)	0.1650 (0.081)	0.2814 (0.146)	0.3315 (0.335)	0.4264 (0.382)	0.5006 (0.401)	0.5092 (0.448)	0.5659 (0.449)	0.6821 (0.578)	+
UF2	0.0142 (0.003)	0.0406 (0.072)	0.0408 (0.095)	0.0708 (0.338)	0.1106 (0.346)	0.3156 (0.374)	0.3773 (0.513)	0.4478 (0.568)	0.6036 (0.614)	0.6266 (0.661)	+
UF3	0.0588 (0.160)	0.0588 (0.166)	0.1213 (0.295)	0.1925 (0.302)	0.3603 (0.356)	0.3668 (0.371)	0.4155 (0.439)	0.4694 (0.455)	0.5412 (0.562)	0.6659 (0.631)	NA
UF4	0.0786 (0.058)	0.2095 (0.079)	0.2723 (0.093)	0.2801 (0.172)	0.2817 (0.174)	0.2839 (0.199)	0.3098 (0.284)	0.4041 (0.287)	0.5817 (0.331)	0.6051 (0.473)	+
UF5	0.0008 (0.045)	0.0023 (0.092)	0.0510 (0.148)	0.0981 (0.177)	0.1302 (0.251)	0.1519 (0.323)	0.1677 (0.350)	0.1846 (0.401)	0.2712 (0.603)	0.5407 (0.629)	+
UF6	0.0052 (0.146)	0.0185 (0.152)	0.1238 (0.213)	0.3066 (0.248)	0.3107 (0.269)	0.4843 (0.370)	0.5028 (0.407)	0.5430 (0.517)	0.6799 (0.573)	0.6848 (0.596)	+
UF7	0.0228 (0.032)	0.0241 (0.125)	0.3047 (0.161)	0.3668 (0.162)	0.4075 (0.184)	0.4936 (0.215)	0.5877 (0.225)	0.6230 (0.270)	0.6263 (0.367)	0.6270 (0.370)	–
UF8	0.0018 (0.192)	0.1102 (0.294)	0.1257 (0.336)	0.1349 (0.363)	0.2262 (0.385)	0.2576 (0.418)	0.5206 (0.432)	0.5853 (0.500)	0.6838 (0.510)	0.6689 (0.510)	+
UF9	0.0627 (0.016)	0.0838 (0.038)	0.1006 (0.187)	0.1747 (0.356)	0.1348 (0.263)	0.2363 (0.449)	0.2936 (0.560)	0.3432 (0.607)	0.4725 (0.650)	0.4574 (0.633)	+
UF10	0.0420 (0.101)	0.0846 (0.124)	0.2725 (0.129)	0.3282 (0.183)	0.4239 (0.190)	0.4246 (0.228)	0.4738 (0.261)	0.5005 (0.398)	0.5244 (0.625)	0.5299 (0.644)	+
UF11	0.0077 (0.041)	0.0642 (0.148)	0.0920 (0.186)	0.1653 (0.332)	0.2632 (0.339)	0.2789 (0.381)	0.3055 (0.430)	0.4026 (0.560)	0.4317 (0.567)	0.5547 (0.650)	+
UF12	0.0645 (0.040)	0.1234 (0.060)	0.1694 (0.154)	0.1825 (0.215)	0.4291 (0.270)	0.4434 (0.342)	0.5338 (0.385)	0.6138 (0.485)	0.6513 (0.543)	0.6844 (0.662)	+
UF13	0.2178 (0.180)	0.0753 (0.171)	0.3099 (0.195)	0.3628 (0.208)	0.3815 (0.351)	0.4974 (0.428)	0.5156 (0.475)	0.5408 (0.569)	0.5689 (0.610)	0.5886 (0.651)	–
CF1	0.0427 (0.089)	0.0620 (0.098)	0.1336 (0.113)	0.1994 (0.139)	0.3096 (0.213)	0.5791 (0.449)	0.4091 (0.382)	0.4090 (0.318)	0.5865 (0.453)	0.6633 (0.553)	+
CF2	0.0235 (0.176)	0.0235 (0.176)	0.2606 (0.207)	0.2738 (0.220)	0.3229 (0.277)	0.3809 (0.393)	0.3951 (0.435)	0.5650 (0.508)	0.6498 (0.597)	0.5760 (0.583)	NA
CF3	0.1301 (0.026)	0.1366 (0.049)	0.2960 (0.129)	0.5451 (0.395)	0.5086 (0.374)	0.3477 (0.289)	0.5598 (0.501)	0.5645 (0.511)	0.6944 (0.634)	0.6924 (0.609)	+
CF4	0.1271 (0.034)	0.2790 (0.077)	0.3316 (0.265)	0.3443 (0.358)	0.3581 (0.374)	0.4695 (0.472)	0.5912 (0.506)	0.6466 (0.644)	0.6655 (0.645)	0.6876 (0.666)	+
CF5	0.0114 (0.113)	0.2091 (0.130)	0.2968 (0.132)	0.3648 (0.172)	0.3714 (0.186)	0.3728 (0.218)	0.3986 (0.247)	0.4895 (0.320)	0.5503 (0.443)	0.6192 (0.589)	+
CF6	0.1260 (0.224)	0.2648 (0.252)	0.2840 (0.287)	0.2865 (0.399)	0.3313 (0.470)	0.4327 (0.492)	0.5519 (0.505)	0.5938 (0.551)	0.6168 (0.574)	0.6813 (0.593)	+
CF7	0.0063 (0.005)	0.0484 (0.020)	0.0632 (0.085)	0.1237 (0.139)	0.2115 (0.237)	0.3152 (0.292)	0.3673 (0.292)	0.3691 (0.304)	0.4371 (0.487)	0.6222 (0.522)	+
CF8	0.0340 (0.108)	0.0343 (0.232)	0.0629 (0.290)	0.1523 (0.309)	0.1666 (0.359)	0.3379 (0.387)	0.4106 (0.428)	0.5816 (0.479)	0.5926 (0.510)	0.6661 (0.614)	+
CF9	0.0878 (0.015)	0.1237 (0.074)	0.2268 (0.139)	0.2718 (0.247)	0.4315 (0.272)	0.4380 (0.295)	0.5553 (0.367)	0.6115 (0.406)	0.6784 (0.492)	0.6912 (0.540)	+
CF10	0.0308 (0.104)	0.0881 (0.129)	0.0950 (0.231)	0.2012 (0.280)	0.3344 (0.418)	0.4869 (0.494)	0.4914 (0.505)	0.6088 (0.539)	0.6462 (0.548)	0.6956 (0.602)	+

The best metric value obtained in each case is shown in bold. A plus sign designates that the t value of 49 degrees of freedom is significant at the 0.05 level of significance by a two-tailed test, and a minus sign denotes that the difference of means is not statistically significant.

CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eEMAS, elitist evolutionary multiagent system; MOEA-RF, multiobjective evolutionary algorithm with robust features; Mod. NSGA-II, modified nondominated sorting genetic algorithm II; NA, not applicable (referring to the cases for which two or more algorithms achieve the same best accuracy results); NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

nificantly inferior to the behavior of the control algorithm. Only for DEMON and NSGA-II-A the null hypothesis cannot be rejected with any of the tests for $\alpha = 0.05$ for all four metrics. However, the other seven algorithms may be regarded as significantly poorer than DENMO with a level of significance $\alpha = 0.05$.

7. Conclusion

We have proposed a novel approach to management of uncertainty during ranking the trial solutions due to possible creeping of noise in the fitness landscapes of a MOO problem. Although most of the traditional evolutionary MOO algorithms can be extended with the principles of uncertainty management introduced herein, we realized the strategies in the DEMO algorithm for its quality performance with respect to both runtime accuracy and computational complexity.

Table 8

Mean hypervolume ratio and standard deviation (in parentheses) for 50 independent runs with Gaussian noise with mean 0 and variance 0.4.

Function	DENMO	DEMON	NSGA-II-A	CDR	SA	eEMAS	MOEA-RF	Mod. NSGA-II	NT-SPEA	ParEGO	Statistical significance
UF1	0.8935 (0.050)	0.8362 (0.103)	0.8309 (0.147)	0.7943 (0.158)	0.7515 (0.234)	0.7091 (0.300)	0.6777 (0.316)	0.6613 (0.354)	0.6562 (0.375)	0.6488 (0.421)	+
UF2	0.9024 (0.141)	0.8269 (0.192)	0.8172 (0.228)	0.8072 (0.252)	0.8034 (0.337)	0.7943 (0.364)	0.7440 (0.364)	0.6755 (0.423)	0.6668 (0.458)	0.6382 (0.463)	+
UF3	0.9007 (0.034)	0.8783 (0.058)	0.8065 (0.060)	0.8045 (0.120)	0.7863 (0.287)	0.7730 (0.299)	0.7534 (0.299)	0.7017 (0.356)	0.6847 (0.396)	0.6381 (0.546)	+
UF4	0.9112 (0.051)	0.8985 (0.216)	0.8470 (0.266)	0.8443 (0.286)	0.8009 (0.288)	0.7434 (0.291)	0.7294 (0.369)	0.7170 (0.382)	0.6614 (0.407)	0.6444 (0.473)	+
UF5	0.8415 (0.010)	0.8200 (0.108)	0.7875 (0.148)	0.7415 (0.181)	0.7149 (0.217)	0.7112 (0.233)	0.6838 (0.243)	0.6776 (0.375)	0.6479 (0.387)	0.6142 (0.404)	+
UF6	0.8840 (0.091)	0.8629 (0.180)	0.8587 (0.241)	0.8461 (0.369)	0.8416 (0.422)	0.7375 (0.434)	0.7269 (0.458)	0.7251 (0.474)	0.7207 (0.522)	0.6691 (0.544)	+
UF7	0.8829 (0.030)	0.8641 (0.049)	0.8527 (0.061)	0.8395 (0.074)	0.7881 (0.084)	0.7646 (0.104)	0.7302 (0.272)	0.7019 (0.272)	0.6639 (0.293)	0.6495 (0.373)	+
UF8	0.9164 (0.125)	0.8974 (0.263)	0.8812 (0.273)	0.8769 (0.290)	0.8722 (0.316)	0.8609 (0.336)	0.8229 (0.440)	0.7864 (0.464)	0.7793 (0.495)	0.6001 (0.544)	+
UF9	0.9143 (0.009)	0.8850 (0.066)	0.8363 (0.115)	0.8335 (0.266)	0.8132 (0.319)	0.8115 (0.319)	0.8003 (0.422)	0.7875 (0.464)	0.6789 (0.474)	0.6267 (0.510)	+
UF10	0.8015 (0.023)	0.7967 (0.080)	0.7767 (0.103)	0.7566 (0.155)	0.7159 (0.244)	0.6657 (0.274)	0.6616 (0.294)	0.6393 (0.296)	0.6158 (0.349)	0.6102 (0.382)	+
UF11	0.8729 (0.045)	0.8796 (0.024)	0.8048 (0.058)	0.7807 (0.078)	0.7569 (0.091)	0.7334 (0.315)	0.6864 (0.341)	0.6667 (0.400)	0.6659 (0.512)	0.6396 (0.521)	–
UF12	0.9150 (0.013)	0.8990 (0.016)	0.8753 (0.073)	0.8748 (0.162)	0.8513 (0.165)	0.8361 (0.183)	0.7642 (0.256)	0.7275 (0.356)	0.6568 (0.463)	0.6202 (0.516)	+
UF13	0.8875 (0.055)	0.8733 (0.158)	0.8120 (0.227)	0.7788 (0.255)	0.7427 (0.296)	0.7113 (0.388)	0.7058 (0.420)	0.6566 (0.450)	0.6378 (0.543)	0.6173 (0.549)	+
CF1	0.8896 (0.057)	0.8161 (0.073)	0.7670 (0.101)	0.7499 (0.116)	0.7150 (0.164)	0.7074 (0.309)	0.6668 (0.328)	0.6569 (0.404)	0.6562 (0.410)	0.6181 (0.501)	+
CF2	0.8871 (0.025)	0.8863 (0.034)	0.7413 (0.055)	0.6983 (0.163)	0.6775 (0.164)	0.6629 (0.182)	0.6298 (0.250)	0.6228 (0.277)	0.6172 (0.418)	0.6042 (0.547)	+
CF3	0.8896 (0.005)	0.8642 (0.026)	0.8487 (0.158)	0.8019 (0.289)	0.7708 (0.331)	0.7081 (0.367)	0.6940 (0.388)	0.6349 (0.401)	0.6287 (0.410)	0.6258 (0.429)	+
CF4	0.8496 (0.028)	0.8372 (0.040)	0.8216 (0.048)	0.7945 (0.266)	0.7781 (0.302)	0.7268 (0.403)	0.7080 (0.439)	0.6409 (0.439)	0.6335 (0.489)	0.6197 (0.518)	+
CF5	0.8397 (0.047)	0.8312 (0.129)	0.8187 (0.201)	0.8092 (0.203)	0.8050 (0.328)	0.7866 (0.376)	0.7052 (0.404)	0.6422 (0.407)	0.6375 (0.476)	0.6353 (0.533)	+
CF6	0.8525 (0.050)	0.8470 (0.055)	0.8054 (0.057)	0.7765 (0.083)	0.7242 (0.130)	0.7176 (0.161)	0.6732 (0.222)	0.6659 (0.266)	0.6658 (0.291)	0.6277 (0.430)	+
CF7	0.9086 (0.020)	0.8510 (0.060)	0.8425 (0.103)	0.8223 (0.146)	0.8097 (0.150)	0.7931 (0.217)	0.7384 (0.268)	0.7383 (0.422)	0.6933 (0.438)	0.6359 (0.513)	+
CF8	0.8228 (0.050)	0.8472 (0.004)	0.8154 (0.059)	0.7951 (0.069)	0.7445 (0.073)	0.7374 (0.232)	0.7010 (0.292)	0.6416 (0.347)	0.6401 (0.360)	0.6190 (0.397)	–
CF9	0.8857 (0.042)	0.8250 (0.091)	0.7015 (0.101)	0.7012 (0.116)	0.6803 (0.172)	0.6696 (0.305)	0.6627 (0.306)	0.6538 (0.342)	0.6454 (0.388)	0.6315 (0.502)	+
CF10	0.9161 (0.084)	0.9161 (0.088)	0.8189 (0.161)	0.7986 (0.192)	0.7287 (0.209)	0.7286 (0.291)	0.7269 (0.377)	0.6824 (0.416)	0.6545 (0.457)	0.6236 (0.479)	NA

The best metric value obtained in each case is shown in bold. A plus sign designates that the t value of 49 degrees of freedom is significant at the 0.05 level of significance by a two-tailed test, and a minus sign denotes that the difference of means is not statistically significant.

CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eEMAS, elitist evolutionary multiagent system; MOEA-RF, multiobjective evolutionary algorithm with robust features; Mod. NSGA-II, modified nondominated sorting genetic algorithm II; NA, not applicable (referring to the cases for which two or more algorithms achieve the same best accuracy results); NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

The integrity of the work lies in the following: (1) adaptive selection of the sample size for the periodic fitness evaluation of a trial solution based on the fitness variance in the subpopulation surrounding it; (2) evaluation of the expected value of the fitness samples of a trial solution based on the nonuniform fitness sample distribution in the entire sample space; and (3) enhancing the robustness in selecting trial solutions from the same-rank candidate pool based on the selection probability influenced jointly by the crowding distance metric and the probability of nonoccurrence of rare (probably noisy) fitness samples.

The adaptive selection of the sample size for fitness reevaluation of a trial solution is significant for the efficient trade-off between runtime complexity and computational accuracy. A nonlinear functional form, induced by the function $1 - \exp(-v)$, is used to proficiently capture the variation of the sample size with the fitness variance v (which is an indirect implication of the extent of noise contamination) in the local neighborhood of a given trial solution. The second noise handling strategy provides a unique fitness estimate of a trial solution based on the expectation of the measured noisy fitness samples.

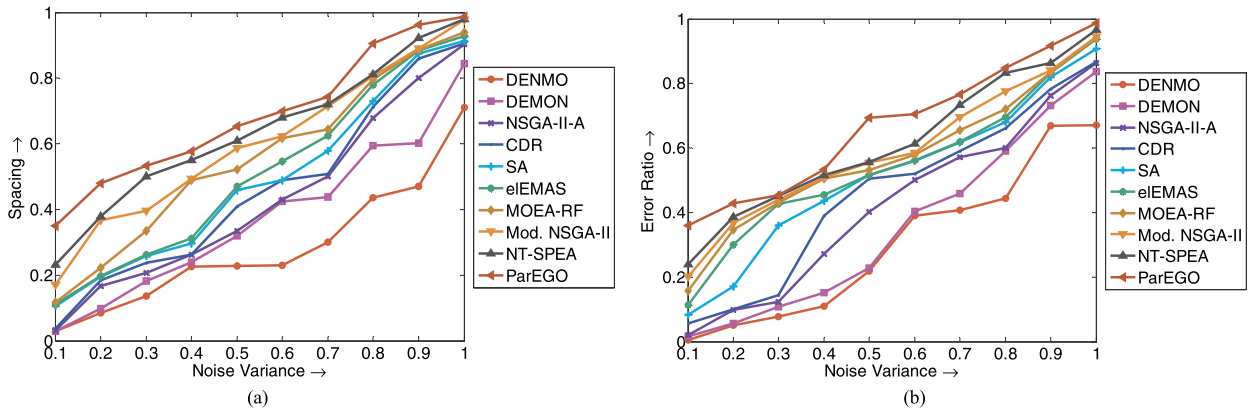


Fig. 5. Average (a) spacing versus Poisson noise variance σ^2 for UF4 and (b) error ratio versus exponential noise variance σ^2 for CF8 for 300,000 function evaluations (30-D problem). CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eIEMAS, elitist evolutionary multiagent system; Mod. NSGA-II, modified nondominated sorting genetic algorithm II; MOEA-RF, multiobjective evolutionary algorithm with robust features; NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

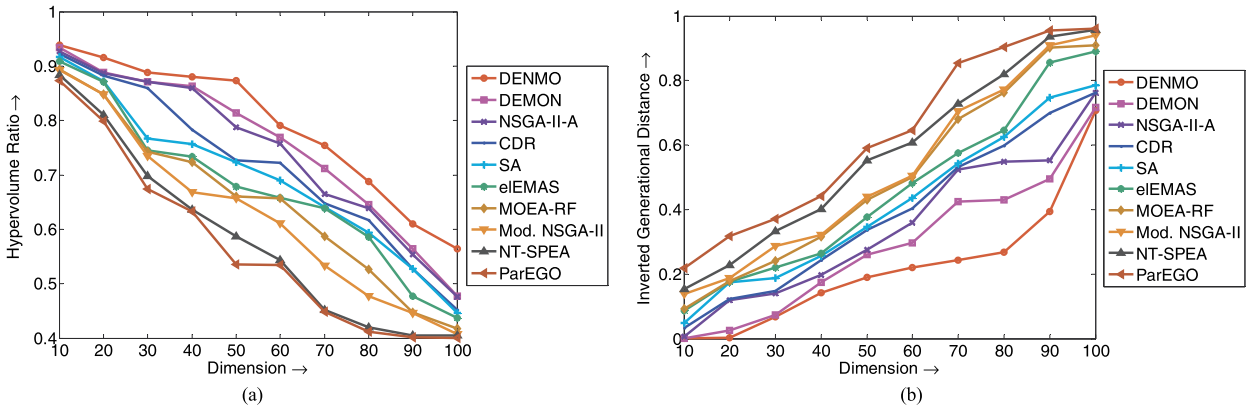


Fig. 6. Average (a) hypervolume ratio versus problem dimension for Rayleigh ($\sigma^2 = 0.25$) noise contaminating CF5 and (b) inverted generational distance versus problem dimension for random (limited-amplitude) noise contaminating UF10. CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eIEMAS, elitist evolutionary multiagent system; Mod. NSGA-II, modified nondominated sorting genetic algorithm II; MOEA-RF, multiobjective evolutionary algorithm with robust features; NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

It differs from the conventional averaging approach which considers equal probability of occurrence for all the fitness samples. Moreover, the proposed strategy recommends a novel method of prioritizing the fitness samples in more crowded regions in the sample space, while reducing the impact of the noisy rare fitness samples, during evaluation of the expected fitness estimate. The PS strategy, adopted as the third extension of the traditional DEMO algorithm, paves a way for the decisive selection of quality solutions in the next-generation population, simultaneously ensuring population diversity in the objective spaces. The stratagem circumvents the dismissal of quality solutions over generations by limiting the impact of misleading individuals in the noisy environment.

We undertook a comparative study of the proposed DENMO algorithm with nine state-of-the-art noisy MOO algorithms. The efficacy of all the contender algorithms to handle uncertainty was scrutinized with respect to the noisy version of a test suite of 23 CEC'2009 benchmark functions. The performance of all the algorithms was compared on the basis of four performance metrics; *IGD*, Δ , *ER*, and *HVR*. Statistical significance of the results was judged with the nonparametric Friedman test, the Iman-Davenport statistic, the Holm test, and Bonferroni–Dunn post hoc analysis. The experimental study clearly reveals that DENMO outperforms its competitor algorithms in a statistically significant manner with respect to four standard metrics in the presence of five different stochastic noise distributions (Gaussian, Poisson, Rayleigh, exponential, and random noise of limited amplitude). In addition, one more fundamental claim of this paper is that DENMO outperforms its competitors both in consistency of quality performance and in expected runtime.

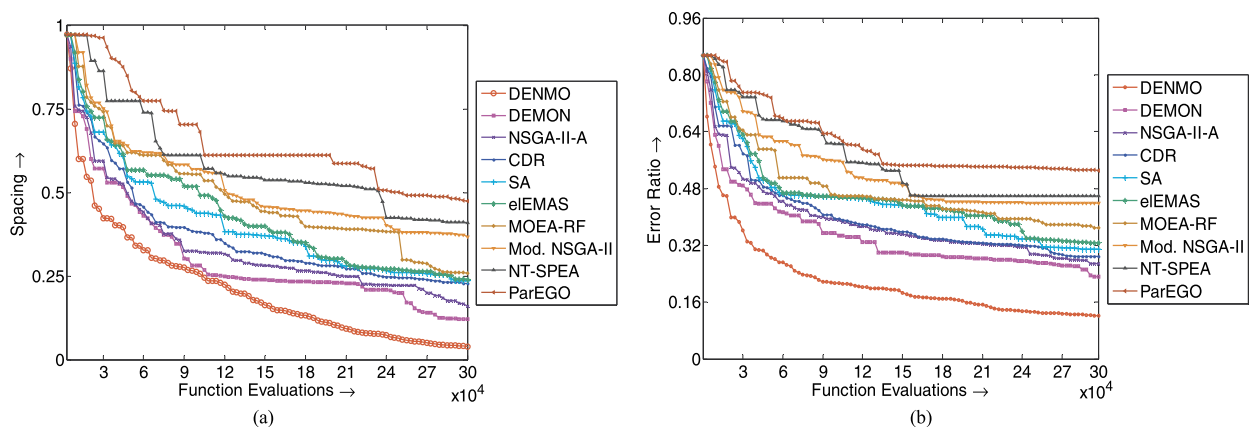


Fig. 7. Average (a) spacing versus number of function evaluations for Poisson ($\sigma^2 = 0.65$) noise contaminating CF4 and (b) error ratio versus number of function evaluations for exponential ($\sigma^2 = 0.5$) noise contaminating UF8. CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eIEMAS, elitist evolutionary multiagent system; Mod. NSGA-II, modified nondominated sorting genetic algorithm II; MOEA-RF, multiobjective evolutionary algorithm with robust features; NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

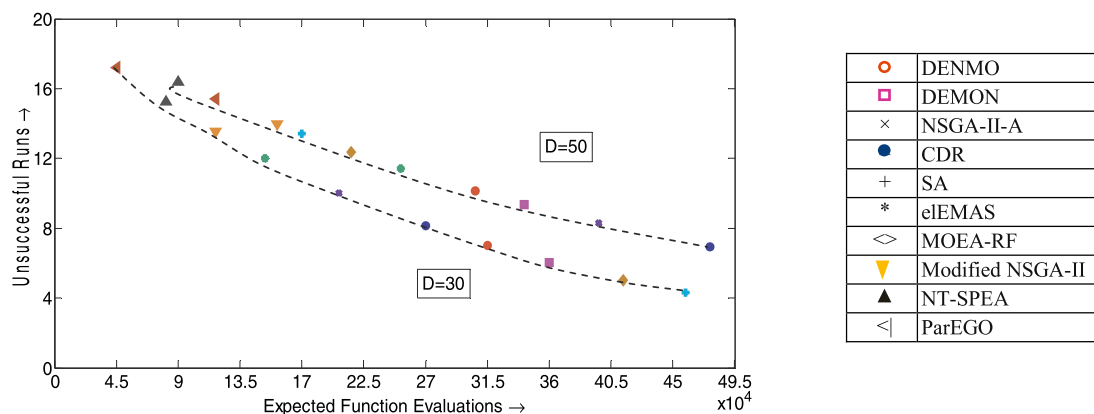


Fig. 8. Relative performance in terms of unsuccessful runs versus expected runtime for UF5 with different settings of noise and problem dimension D . CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eIEMAS, elitist evolutionary multiagent system; MOEA-RF, multiobjective evolutionary algorithm with robust features; NSGA-II, nondominated sorting genetic algorithm II; NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

Table 9

Rankings obtained from Fig. 8.

Ranking	30-D UF5 with zero mean Gaussian noise ($\sigma^2 = 0.35$)	50-D UF5 with Poisson noise ($\sigma^2 = 0.7$)
1	DENMO	DENMO
2	DEMON	DEMON
3	CDR	NSGA-II-A
4	MOEA-RF	CDR
5	NSGA-II-A	eIEMAS
6	SA	MOEA-RF
7	eIEMAS	SA
8	Modified NSGA-II	Modified NSGA-II
9	NT-SPEA	ParEGO
10	ParEGO	NT-SPEA

CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eIEMAS, elitist evolutionary multiagent system; MOEA-RF, multiobjective evolutionary algorithm with robust features; NSGA-II, nondominated sorting genetic algorithm II; NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

Table 10

Average rankings obtained through Friedman's test.

Algorithm	Friedman ranking obtained from			
	Table 5	Table 6	Table 7	Table 8
DENMO	1.0652	1.0869	1.0869	1.1087
DEMON	1.9347	1.9130	1.9130	1.8913
NSGA-II-A	3.1304	3.0000	3.0000	3.0000
CDR	3.9565	4.0434	4.1304	4.0000
SA	4.9130	5.0869	4.9565	5.0000
eEMAS	6.0000	6.0434	6.0000	6.0000
MOEA-RF	7.0434	6.9565	7.0000	7.0000
Modified NSGA-II	7.9565	7.8695	7.9130	8.0000
NT-SPEA	9.0434	9.0000	9.1739	9.0000
ParEGO	9.9565	10.0000	9.8260	10.0000
Friedman statistic	205.2410	204.2680	203.9748	206.5138
Iman-Davenport statistic	2566.971	1644.910	1483.353	9344.515
Critical difference for $\alpha = 0.05$	2.2677			

CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eEMAS, elitist evolutionary multiagent system; MOEA-RF, multiobjective evolutionary algorithm with robust features; NSGA-II, nondominated sorting genetic algorithm II; NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

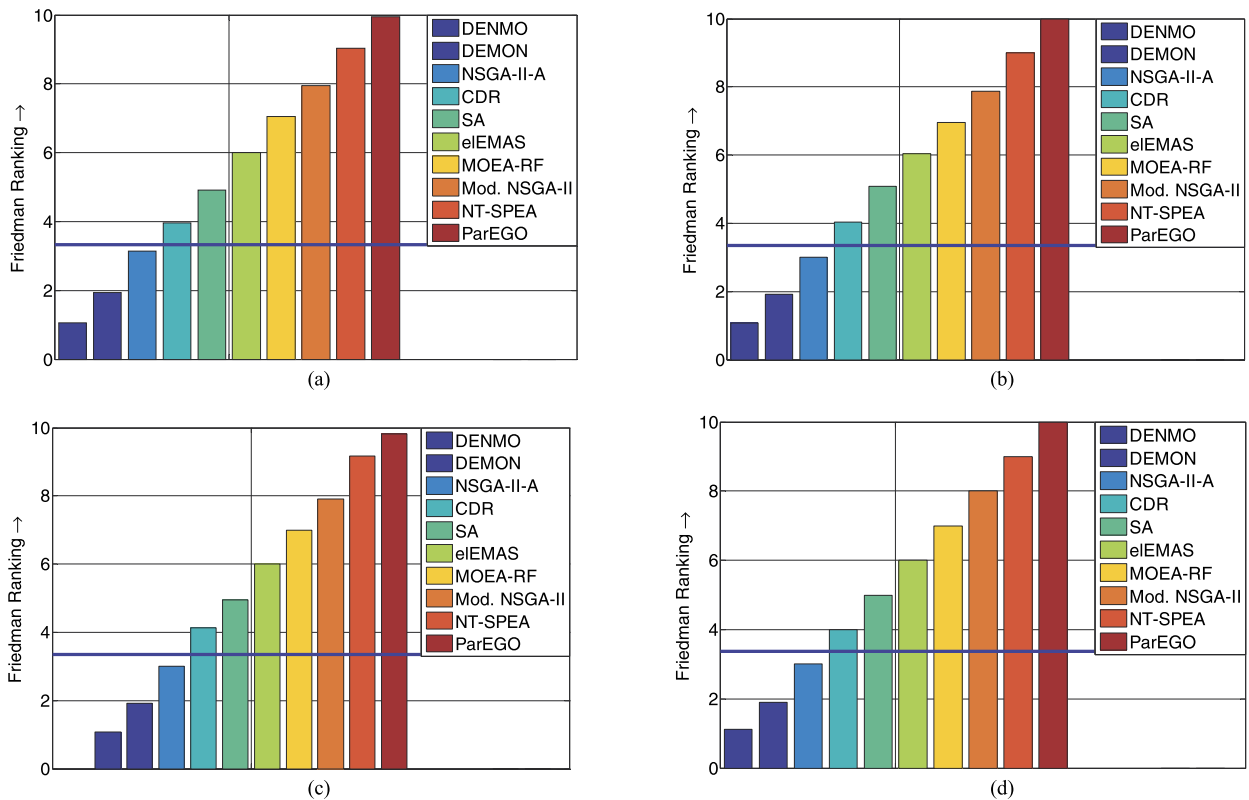


Fig. 9. Graphical representation of Bonferroni–Dunn's procedure with DENMO as the control method using the results given in Table 10 corresponding to the values for the (a) inverted generational distance in Table 5, (b) spacing (Δ) in Table 6, (c) error ratio in Table 7, and (d) hypervolume ratio in Table 8. CDR, confidence-based dynamic resampling; DEMON, differential evolution for multiobjective optimization with noise; DENMO, differential evolution for noisy multiobjective optimization; eEMAS, elitist evolutionary multiagent system; Mod. NSGA-II, modified nondominated sorting genetic algorithm II; MOEA-RF, multiobjective evolutionary algorithm with robust features; NSGA-II-A, nondominated sorting genetic algorithm II with α -dominance operator; NT-SPEA, noise-tolerant strength Pareto evolutionary algorithm; ParEGO, Pareto front-efficient global optimization; SA, simulated annealing.

Acknowledgements

Funding by the Council of Scientific and Industrial Research (for the award of a Senior Research Fellowship to Pratyusha Rakshit with Acknowledgment No. 143392/2K12/1 and FILE NO: 09/096(781)/2013-EMR-I) and UGC-UPE-II, Cognitive Science Program, Jadavpur University, is gratefully acknowledged.

References

- [1] A. Chowdhury, A. Konar, P. Rakshit, A.K. Nagar, A multi-objective evolutionary approach to evaluate the designing perspective of protein–protein interaction network, *J. Netw. Innov. Comput.* 1 (2013) 270–289.
- [2] A.N. Aizawa, B.W. Wah, Scheduling of genetic algorithms in a noisy environment, *Evol. Comput.* 2 (2) (1994) 97–122.
- [3] J. Branke, C. Schmidt, Selection in the presence of noise, in: *Genetic and Evolutionary Computation Conference*, in: *Lecture Notes in Computer Science*, vol. 2723, Springer-Verlag, Berlin, June 2003, pp. 766–777.
- [4] B.L. Miller, Noise, sampling, and efficient genetic algorithms, Doctor of Philosophy thesis, University of Illinois, Urbana-Champaign, 1997.
- [5] S. Markon, D. Arnold, T. Back, T. Beislstein, H.G. Beyer, Thresholding—a selection operator for noisy ES, in: *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1, May 2001, pp. 465–472.
- [6] B.L. Miller, D.E. Goldberg, Genetic algorithms, selection schemes, and the varying effects of noise, *Evol. Comput.* 4 (2) (1996) 113–131.
- [7] M. Babbar, A. Lakshmikantha, D.E. Goldberg, A modified NSGA-II to solve noisy multi-objective problems, in: *Genetic and Evolutionary Computation Conference*, AAAI, Chicago, 2003, pp. 21–27, Late-Breaking Papers.
- [8] T. Robic, B. Philipic, DEMO: differential evolution for multiobjective optimization, in: *The Third International Conference on Evolutionary Multi-Criterion Optimization*, in: *Lecture Notes in Computer Science*, vol. 3410, Springer-Verlag, Berlin, 2005, pp. 520–533.
- [9] P. Rakshit, A. Konar, S. Das, L.C. Jain, A.K. Nagar, Uncertainty management in differential evolution induced multi-objective optimization in presence of measurement noise, *IEEE Trans. Syst. Man Cybern., Syst.* 44 (7) (July 2014) 922–937.
- [10] P. Boonma, J. Suzuki, A confidence-based dominance operator in evolutionary algorithms for noisy multiobjective optimization problems, in: *Proceedings of 21st IEEE International Conference on Tools with Artificial Intelligence*, November 2009, pp. 387–394.
- [11] A. Syberfeldt, A. Ng, R.I. John, P. Moore, Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling, *Eur. J. Oper. Res.* 204 (3) (2010) 533–544.
- [12] V. Mattila, K. Virtanen, R.P. Hämmäläinen, A simulated annealing algorithm for noisy multiobjective optimization, *J. Multi-Criteria Decis. Anal.* 20 (5–6) (2013) 255–276.
- [13] L. Sivik, S. Natanek, Elitist evolutionary multi-agent system in solving noisy multi-objective optimization problems, in: *Proceedings of IEEE Congress on Evolutionary Computation*, June 2008, pp. 3319–3326.
- [14] C.K. Goh, K.C. Tan, An investigation on noisy environments in evolutionary multiobjective optimization, *IEEE Trans. Evol. Comput.* 11 (3) (2007) 354–381.
- [15] D. Buche, P. Stall, R. Dornberger, P. Koumoutsakos, Multiobjective evolutionary algorithm for the optimization of noisy combustion processes, *IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev.* 32 (4) (November 2002) 460–473.
- [16] J. Knowles, E.J. Hughes, Multiobjective optimization on a budget of 250 evaluations, in: *Evolutionary Multi-Criterion Optimization*, in: *Lecture Notes in Computer Science*, vol. 3410, Springer-Verlag, Berlin, 2005, pp. 176–190.
- [17] P. Staggie, Averaging efficiently in the presence of noise, in: *The Fifth International Conference on Parallel Problem Solving from Nature*, in: *Lecture Notes in Computer Science*, vol. 1498, Springer-Verlag, 1998, pp. 188–197.
- [18] S. Das, A. Konar, U.K. Chakraborty, Improved differential evolution algorithms for handling noisy optimization problems, in: *Proceedings of IEEE Congress of Evolutionary Computation*, vol. 2, September 2005, pp. 1691–1698.
- [19] E.J. Hughes, Evolutionary multi-objective ranking with uncertainty and noise, in: *Evolutionary Multi-Criterion Optimization*, in: *Lecture Notes in Computer Science*, vol. 1993, Springer-Verlag, Berlin Heidelberg, July 2001, pp. 329–343.
- [20] A. Singh, B. Minsker, Uncertainty based multi-objective optimization of groundwater remediation at the umatilla chemical depot, in: *Critical Transitions in Water and Environmental Resources Management*, Environmental and Water Resources, Ground Water Issues Symposium, 2004, pp. 1–10.
- [21] J. Knowles, D. Corne, A. Reynolds, Noisy multiobjective optimization on a budget of 250 evaluations, in: *Evolutionary Multi-Criterion Optimization*, in: *Lecture Notes in Computer Science*, vol. 5467, Springer, Berlin, April 2009, pp. 36–50.
- [22] P. Rakshit, A. Konar, P. Bhowmik, I. Goswami, S. Das, L.C. Jain, A.K. Nagar, Realization of an adaptive memetic algorithm using differential evolution and Q-learning: a case study in multirobot path planning, *IEEE Trans. Syst. Man Cybern., Syst.* 43 (4) (July 2013) 814–831.
- [23] R. Storn, K.V. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *J. Glob. Optim.* 11 (4) (1997) 341–359.
- [24] S. Das, A. Abraham, U.K. Chakraborty, A. Konar, Differential evolution using a neighborhood-based mutation operator, *IEEE Trans. Evol. Comput.* 13 (3) (June 2009) 526–553.
- [25] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA II, *IEEE Trans. Evol. Comput.* 6 (2) (April 2002) 182–197.
- [26] J.M. Fitzpatrick, J.J. Greffensette, Genetic algorithms in noisy environments, *Mach. Learn.* 3 (2–3) (October 1988) 101–120.
- [27] E. Mendel, R.A. Krohling, M. Campos, Swarm algorithms with chaotic jumps applied to noisy optimization problems, *Inf. Sci.* 181 (20) (2011) 4494–4514.
- [28] M.K. Gupta, A.M. Gun, B. Dasgupta, Measures of Central Tendency, *Fundam. Stat.*, vol. 1, World Press Pvt Ltd., 2008.
- [29] S. Heymann, M. Latapy, C. Magnien, Outskewer: using skewness to spot outliers in samples and time series, in: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining*, IEEE Computer Society, August 2012, pp. 527–534.
- [30] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective optimization test instances for the CEC 2009 special session and competition, Working Report, CES-887, School of Computer Science and Electrical Engineering, University of Essex, Colchester, UK and Nanyang Technological University, Singapore, April 20, 2009, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report 2008.
- [31] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: *Proceedings of IEEE Congress of Evolutionary Computation*, vol. 1, May 2002, pp. 825–830.
- [32] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, *Evol. Comput.* 8 (2) (June 2000) 173–195.
- [33] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, *IEEE Trans. Evol. Comput.* 10 (5) (October 2006) 477–506.
- [34] G.E.P. Box, M.E. Muller, A note on the generation of random deviates, *Ann. Math. Stat.* 29 (2) (1958) 610–611.
- [35] D.E. Knuth, *The Art of Computer Programming*, vol. 2, *Seminumerical Algorithms*, 3rd ed., Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

- [36] W. Hörmann, J. Leydold, G. Derflinger, General Principles in Random Variate Generation, Automatic Nonuniform Random Variate Generation, Springer, Berlin, 2004, pp. 13–41.
- [37] G. Marsaglia, W.W. Tsang, The Ziggurat method for generating random variables, *J. Stat. Softw.* 5 (8) (2000) 1–7.
- [38] S. Tezuka, Uniform Random Numbers: Theory and Practice, The Springer International Series in Engineering and Computer Science, vol. 315, 1995, XII, p. 209.
- [39] J.R. Schott, Fault tolerant design using single and multi-criteria genetic algorithm optimization, Master of Science thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.
- [40] D.A. van Veldhuizen, G.B. Lamont, Multiobjective evolutionary algorithms: analyzing the state-of-the-art, *Evol. Comput.* 8 (2) (2000) 125–147.
- [41] C.A. Coello Coello, G.B. Lamont, D.A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed., Genetic and Evolutionary Computation Series, 2007.
- [42] M. Fleischer, The measure of pareto optima. Applications to multi-objective metaheuristics, in: *Second International Conference on Evolutionary Multi-Criterion Optimization*, in: Springer. Lecture Notes in Computer Science, vol. 2632, Springer-Verlag, Berlin, April 2003, pp. 519–533.
- [43] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, 4th edition, Chapman and Hall/CRC, 2007.
- [44] S. Picek, M. Golub, D. Jakobovic, Evaluation of crossover operator performance in genetic algorithms with binary representation, in: *The Seventh International Conference on Intelligent Computing: Bio-Inspired Computing and Applications*, in: Lecture Notes in Computer Science, vol. 6840, Springer-Verlag, Berlin, 2012, pp. 223–230.
- [45] B. Flury, *A First Course in Multivariate Statistics*, Springer Texts in Statistics Series, 1997.
- [46] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.* 1 (1) (March 2011) 3–18.
- [47] S.O. García, A.G. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, *Inf. Sci.* 180 (10) (May 2010) 2044–2064.
- [48] L.V. Santana-Quintero, C.A. Coello Coello, An algorithm based on differential evolution for multi-objective problems, *Int. J. Comput. Intell. Res.* 1 (1) (2005) 151–169.
- [49] P. Rakshit, A.K. Sadhu, A. Halder, A. Konar, R. Janarthanan, Multi-robot box-pushing using differential evolution algorithm for multiobjective optimization, in: *Proceedings of the International Conference on Soft Computing for Problem Solving, Advances in Intelligent and Soft Computing*, vol. 130, 2012, pp. 355–365.