



Many-objective differential evolution optimization based on reference points: NSDE-R

Sohail R. Reddy¹ · George S. Dulikravich¹

Received: 17 July 2018 / Revised: 8 January 2019 / Accepted: 31 March 2019 / Published online: 30 April 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

Design methodologies of today require the solution of several many-objective optimization problems. The last two decades have seen a surge in several algorithms capable of solving multi-objective optimization problems. It was only in the past 5 years that new algorithms capable of coping with a large number of objectives have been introduced. This work presents a new differential evolution algorithm (NSDE-R) capable of efficiently solving many-objective optimization problems. The algorithms make use of reference points evenly distributed through the objective function space to preserve diversity and aid in multi-criteria-decision-making. The proposed NSDE-R was applied to test problems from the DTLZ and WFG suite, having three to 15 objectives. Two mutation donor operators were investigated for their ability to converge to the analytical Pareto front while maintaining diversity. The ability of NSDE-R to converge to a user-specified region of the Pareto front is also investigated. The proposed NSDE-R algorithm has shown to have a higher rate of convergence and better convergence to the analytical Pareto front.

Keywords Evolutionary computation · Differential evolution · Many-objective optimization · Non-dominated sorting · NSDE-R · Reference points

1 Introduction

Evolutionary and heuristic optimization methods have been extensively exploited for their ability to explore and exploit vast design and objective function space. The last two decades have given rise to several multi-objective optimization algorithms capable of finding a well-converged and well-diversified Pareto set of a two to three objective optimization problems. The optimization problems faced today require the solution of five to 10+ objectives (Chikumbo et al. 2012; Coello Coello and Lamont 2004) and are typically referred to as many-objective optimization problems. This leads to a

challenging increase in size or the restriction placed on the objective functions space. The current algorithms suffer from solution clustering in certain regions of the objective function space and are not able to preserve diversity in the Pareto solutions. The need for the solution of these large scale optimization problems has led to the development of several many-objective optimization algorithms in the last 5 years.

A typical multi-objective optimization problem can be stated as

$$\begin{aligned} \min & \vec{f}(\vec{x}) \\ \vec{f} &= \{f_1(\vec{x}), \dots, f_l(\vec{x})\} \\ \vec{x} &= \{x_1, \dots, x_m\} \\ \text{subject to } & x \in [a_i, b_i] \quad i = 1, \dots, m \end{aligned} \quad (1)$$

subject to a number of inequality and equality constraints. These optimizers typically use various niching techniques to preserve diversity and typically use either decomposition based or indicator based methods. Decomposition based algorithms make use of reference points or reference directions to guide the search to scarce areas of the Pareto fronts. Algorithms using decomposition based approach include Non-dominated Sorting Genetic Algorithm (NSGA-III) (Deb and Jain 2014), Many-objective Optimization Evolutionary Algorithm based on Dominance and

Responsible Editor: Mehmet Polat Saka

✉ Sohail R. Reddy
sredd001@fiu.edu

George S. Dulikravich
dulikrav@fiu.edu

¹ Department of Mechanical and Materials Engineering and Multidisciplinary Analysis, Inverse Design, Robust Optimization and Control - MAIDROC Laboratory, Florida International University, Miami, FL 33174, USA

Decomposition (MOEA-DD) (Li et al. 2015), and Strength Pareto Evolutionary Algorithm (SPEA-R) (Jiang and Yang 2017). Indicator-based algorithms use certain performance metrics to quantify the superiority of one solution over another and control the recombination and selection operators. Some indicator-based algorithms include Indicator Based Evolutionary Algorithm (IBEA) (Zitzler and Künzli 2004) and Hypervolume-Based Estimation Algorithm (HypE) (Bader and Zitzler 2011a). The indicator-based algorithms are heavily influenced by the accuracy of the indicators used and by the computational times required to compute the indicators. Decomposition-based algorithm does not make use of indicators in the search process, thereby reducing the number of operations required for each generation.

This work presents a Non-dominated Sorting Differential Evolution algorithm based on reference points (NSDE-R). The effects of two different recombination operators on the performance on the NSDE-R algorithm are also investigated. The algorithm uses the recombination operator to create a candidate design, non-dominated sorting to find the non-dominated front, and a reference point-based niching strategy for environmental selection.

The performance of NSDE-R algorithm is evaluated against the performance of NSGA-III and MOEA/DD algorithms. The test problems were taken from the DTLZ and WFG test suites and feature three to 15 objectives. The performance is quantified utilizing the widely used Inverted Generalized Distance (IGD) and Hypervolume (HV) measures. The best, average, and worst IGD and HV metrics are presented for each algorithm.

2 Related work

There have been significant developments in the area of genetic evolutionary algorithms when solving many-objective optimization problems. The previously mentioned NSGA-III, MOEA/DD, and SPEA-R all use simulated binary crossover (Deb and Agrawal 1994) and polynomial mutation (Deb and Goyal 1996) to create offspring/candidate solutions. Therefore, they are closer to genetic algorithms in their algorithm structure. Whereas the development in genetic algorithms has progressed steadily over the decade, there has not been a differential evolution (DE) algorithm that is able to cope with large numbers of simultaneous objectives.

The differential evolution algorithm was first introduced by Storn and Price (Storn and Price 1997) in 1997 for single-objective problems. Das and Suganthan (Das and Suganthan 2011) recently performed a comprehensive review of the advances in differential evolution algorithm. They have shown that there did not exist a differential evolution algorithm for many-objective optimization. Robic and Filipic (Robic and Filipic 2005) presented the most popular differential evolution

algorithm for multi-objective optimization (DEMO) (Robic and Filipic 2005). This algorithm uses the same Pareto ranking and crowding distance ranking scheme as NSGA-II (Deb et al. 2002) and, therefore, also underperforms when solving many-objective optimization problems. Recent work of He et al. (He et al. 2014) proposed a DE algorithm for many-objective optimization based on ranking (MODER). However, the MODER was shown to consistently perform worse than the NSGA-III algorithm. Denysiuk et al. (Denysiuk et al. 2013) also presented a many-objective version of the DE algorithm, but only compared its performance to multi-objective optimization algorithms.

To minimize the computing time in many-objective optimization problems, the algorithms should utilize the users' preferences within their search procedures rather than insisting on uniform accuracy and details of the search. The proposed NSDE-R algorithm is able to incorporate the users' preferences when performing the search and performs better for many-objective problems than the aforementioned algorithms.

3 Proposed NSDE-R algorithm

The basic framework of the proposed algorithm is presented in Algorithm 1. The NSDE-R starts with an initial population set P_i , and reference points set, R . The reference points can be uniformly distributed through objective function space or can be biased towards particular regions. At each generation, the NSDE-R algorithm then selects three members and performs its mutation operator. The parent population and offspring population, O , are then combined and normalized. Each individual in the combined population is then associated with the closest reference points. Then, an environmental selection procedure is applied to select individuals for the next generation that can aid convergence and preserve diversity. The following sections present each operation in the NSDE-R algorithm.

Algorithm 1: Framework for NSDE – R

Input: Parent population P_i , reference points R

Output: P_{i+1}

- 1: p = Unique selection (P_i)
 - 2: O = Mutation (p)
 - 3: $C = p \cup O$
 - 4: Normalize (C)
 - 5: Associate (C, R)
 - 6: S_i = Non-dominated Sorting (C) such that $|S_i| \leq |P_i|$
 - 6: $P_{i+1} = \text{Niching}(C \cap S_i) \cup S_i$
-

3.1 Generation of reference points

The proposed NSDE-R algorithm uses reference points to guide the search and preserve diversity. The reference points can be uniformly distributed through the objective function space or can be biased towards a particular region of interest. Results from both approaches are presented in this work. Two approaches can be used to uniformly distribute reference points through the objective function space. The first approach uses Das and Dennis’ (Das and Dennis 1998) systematic approach to place points on a normalized unit hyperplane. This is shown in Fig. 1a. This method guarantees uniform distribution of reference points, but suffers from the curse of dimensionality. Using this approach, for d divisions along each objective axis, the total number of reference points N in M -objective space is given by (2).

$$N = \binom{M + d - 1}{d} = \frac{(M + d - 1)!}{(M - 1)!d!} \tag{2}$$

For example, given eight divisions ($d = 8$) along each objective of a 10-objective problem ($M = 10$), the total number of points produced by Das and Dennis’ approach is 24,310. This drastically increases the computational cost when associating the designs with the reference points. The number of points can be significantly reduced using a two-step approach where an outer layer is first created followed by an inner layer as shown in Fig. 1b. Consider a 3-objective case with two divisions along the outer layer and one division along the inner layer. Then, the two-step systematic approach will yield a total of 9 points.

An alternative method for uniformly distributing reference points in high dimensional space is to utilize a random number generator. A pseudo-random sequence that uniformly

distributes points in M -dimensional space can be used. One can utilize Uniform Latin Hypercube Sampling (Wyss and Jorgensen 1998). Figure 2 shows such distribution of 20 points for a 3-objective problem.

3.2 Selection and recombination

The traditional DE algorithm selects three random, unique parents and combines them to create an offspring. At the end of each generation, each individual in the current population is compared to remaining members of the population to identify its uniqueness. An individual I is said to be unique, in design variable space, to a set A of size N_A , if

$$\left\| \vec{T} - \vec{A}_i \right\|_{i \in \{1, \dots, N_A\}} \geq \varepsilon \tag{3}$$

This means that the L^2 norm between the variables defining individual \vec{T} and the variables of every individual in \vec{A} must be greater than some specified tolerance, ε . All required parent population that must satisfy uniqueness is selected from this unique archive, U . A new archive U is constructed every generation. For each individual i in parent population P , three parents are randomly selected from the unique archive (Alg. 1: Step 1) to perform mutation and crossover. Two different mutation operators are investigated for the NSDE-R algorithm (Alg. 1: Step 2). The most common used mutation operator in DE algorithms is the “rand/1/bin” (R1B) (Storn and Price 1997) given by

$$\vec{V}_i = \vec{X}_{r_1} + F(\vec{X}_{r_2} - \vec{X}_{r_3}) \tag{4}$$

where $i = 1, \dots, N_P$, $\vec{V}_i = [v_{i,1}, \dots, v_{i,N}]$ and $\{r_j\}_{j=1, \dots, 3} \in U \neq i$. The factor F determines the scaling of the difference

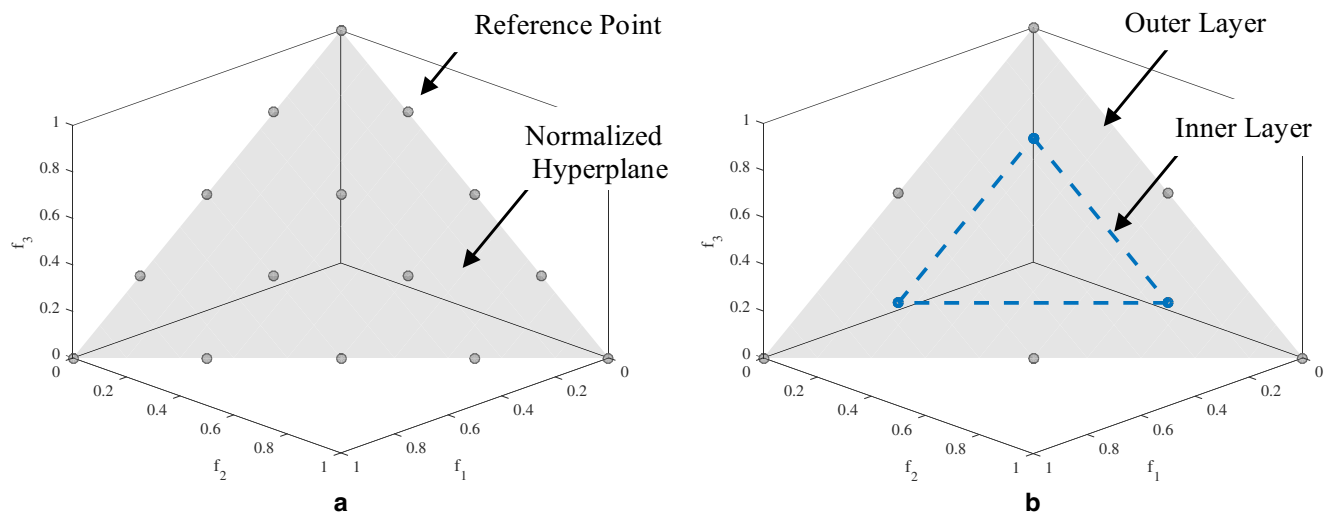


Fig. 1 Das and Dennis’ method applied using a standard approach and b two-layer approach

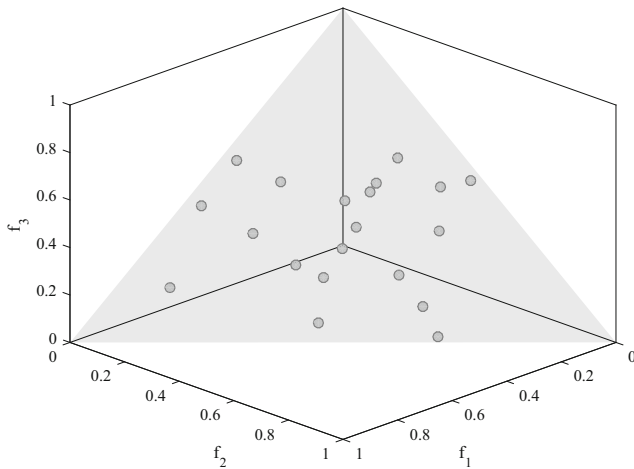


Fig. 2 Randomly distributed reference points on a unit hyperplane

between two random parents. In the above mutation, parent r_3 is taken as a donor parent. The R1B approach selects the donor vector randomly from the unique archive, U . Fan et al. (Fan et al. 2010) proposed using a convex sum of the three parents to create a donor vector. The combination is given as

$$donor = \sum_{k=1}^3 \mu_k \vec{X}_{r_k} \tag{5}$$

where μ is a random number. Different distributions of the parameter μ result in different mutation scheme. Here, μ is assumed to be a uniformly distributed random number $\mu \in [0,1]$ which leads to the Donor3 (D3) scheme given by

$$donor3 = \frac{\sum_{k=1}^3 \mu_k \vec{X}_{r_k}}{\sum_{j=1}^3 \mu_j} \tag{6}$$

The complete D3 mutation scheme can then be given by

$$\vec{V}_i = \sum_{k=1}^3 \frac{\mu_k \vec{X}_{r_k}}{\sum_{j=1}^3 \mu_j} + F(\vec{X}_{r_2} - \vec{X}_{r_3}) \tag{7}$$

The crossover operation is then performed using the mutated vector and the i th individual. The offspring o is then given by

$$o_j = \begin{cases} v_j & \text{if } \text{rand} \leq C_r \vee j = k \\ x_{i,j} & \text{otherwise} \end{cases} \Big|_{j=1, \dots, N} \tag{8}$$

where C_r is the crossover probability. An offspring is created for each individual i in the parent population, P . The parent population and offspring population are then combined (Alg. 1: Step 3) to create the combined population, C , and the NSDE-R algorithm proceeds to objective normalization. The complete selection and recombination procedure is given in Algorithm 2.

Algorithm 2: Recombination

Input: P (parent population), U (unique archive)
Output: C (combined population)
 1: $C = P$
 2: **for** each p in P **do**
 3: Select random parents $\{r\}$ from U
 4: $v = \text{mutate}(r)$ (Rand/1/Bin or Donor3)
 5: $o = \text{crossover}(v, p)$
 6: $C = C \cup o$
 7: **end do**

3.3 Objective normalization

Normalization of the population members at each generation allows for diversity preservation and solution of differently scaled optimization problems. The normalization procedure is given in Algorithm 3.

Algorithm 3: Population Normalization

Input: C (combined population)
Output: C^n (normalized combined population)
 1: **For** $j = 1$ to M **do**
 2: Compute ideal point: $f_j^{\min} = \min_{i \in C} f_j(x_i)$
 3: Translate objectives: $C' = C - f_j^{\min}$
 4: Compute extreme points of C'
 5: **end do**
 6: Compute hyperplane intercepts a_i
 7: Normalize objective using Eq. (9): $C^n = C' / a_i$

The first step is to find the ideal point of the combined population, C , such that $|C| = N_P$. The approximate ideal point, $\mathbf{z}^{\min} = (f_1^{\min}, \dots, f_M^{\min})$ is calculated by

$$f_j^{\min} = \min_{i \in C} f_j(x_i), j = 1, \dots, M \tag{9}$$

Each member in C is then translated by \mathbf{z}^{\min} as $f' = f - \mathbf{z}^{\min}$, such that the ideal point is now the zero vector. The extreme solutions from the translated population are then found. A solution i in C is extreme for axis i if the solution i makes the achievement scalarizing function (ASF) minimum (Figueiredo et al. 2016). The extreme solutions are then used to build a hyperplane, whose intercepts a_i are then used to normalize the population C as

$$f_i^n = \frac{f'_i(x)}{a_i}, i = 1, \dots, M \tag{10}$$

This normalized objective function is then used to perform the association procedure.

3.4 Member association

The association procedure associates each individual in the set to a single reference point. A member is associated with the reference point that has the shortest perpendicular distance between that individual and a line passing through the reference point and the origin. All distances are computed in the normalized objective space. This can be seen as a measure of density in the vicinity of the reference point. That is, the more individuals that are associated with a reference point, the denser (more crowded) that region will become. Figure 3 illustrates this association procedure, where the black dots are the reference points, the red squares are the individuals in the normalized objective functions' space, and the blue lines are the rays connecting the reference points and the origin. It should be noted that a reference point may have one or more individuals associated with it or it may have none. The algorithm for association is given in Algorithm 4.

3.5 Non-dominated sorting

The combined population C is then sorted into its non-dominated front levels using the usual non-dominated sorting approach (Chankong and Haimes 1983). Members of the first l front levels are added to S_l such that $|S_l| \leq N$. If $|S_l| = N$, then no further operations are needed and the next generation is started with $P_{i+1} = S_l$. If $|S_l| > N$, then members from the first $l - 1$ fronts are added to S_l . The current population for the next generation is given by $P_{i+1} = \cup_{l=1}^{l-1} F_l$. The remaining ($K = N_p - |S_l|$) population is selected from the l th front, F_l , using the niching procedure.

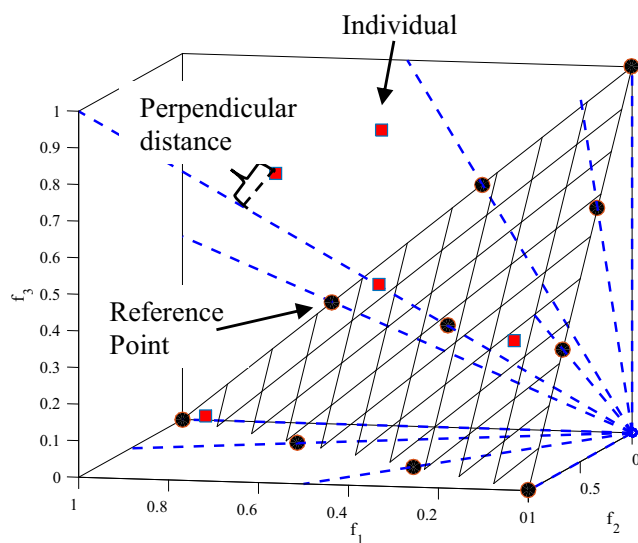


Fig. 3 Association of population members with reference points

Algorithm 4: Association Procedure

Input: R (reference points), C (combined population)
Output: π, d
 1: **for** each reference point $r \in R$ **do**
 2: compute reference line $\mathbf{w} = \mathbf{r}$
 3: **end do**
 4: **for** each c in C **do**
 5: Compute $d^{\perp}(\mathbf{c}, \mathbf{w}) = \|(\mathbf{c} - \mathbf{w}^T \mathbf{c} \mathbf{w}) / \|\mathbf{w}\|^2\|$
 6: Assign $\pi(c) = \mathbf{w} : \operatorname{argmin} d^{\perp}(\mathbf{c}, \mathbf{w})$
 7: Assign $d(c) = d^{\perp}(\mathbf{c}, \pi(c))$
 8: **end do**

3.6 Niching procedure

The niching procedure aims to select a diverse population to carry into the next generation. Non-dominated sorting and niching are used together to carefully select a total of $|N_p|$ individuals from a combined population C of size $2|N_p|$. The niching procedure is given in Algorithm 5.

Algorithm 5: Niching Procedure

Input: K (pop. needed), $\pi(\mathbf{s} \in S_l)$ (associated ref pt.), $d(\mathbf{s} \in S_l)$ (dist. to ref. point), ρ_j (niche count), R (Ref. Pt.)
Output: P_{i+1}
 1: $k = 1$
 2: **while** $k \leq K$ **do**
 3: $J = \operatorname{random}(\{j : \operatorname{argmin}_{j \in R} \rho_j\})$
 4: $I_J = \{\mathbf{s} : \pi(\mathbf{s}) = J, \mathbf{s} \in F_l\}$
 5: **if** $I_J \neq \emptyset$ **then**
 6: **if** $\rho_J = 0$ **then**
 7: $p = \{\mathbf{s} : \operatorname{argmin}_{j \in F_l} d(\mathbf{s})\}$
 8: **else**
 9: $p = \operatorname{random}(I_J)$
 10: **end if**
 11: $P_{i+1} = P_{i+1} \cup p$
 12: $\rho_J = \rho_J + 1, F_l = F_l \setminus p$
 13: $k = k + 1$
 14: **else**
 15: $R = R \setminus J$
 16: **end if**
 17: **end while**

As previously stated, a reference point can have one or more members associated with it or it can have no individuals associated with it. Let us denote the niche count, ρ_j , of the j th reference point as the number of individuals associated with that reference point. The reference point with the highest niche count is the most crowded reference point.

The points to carry over into the next generation are selected as follows. The least used reference point is first identified. This will be the reference point with the lowest niche count. In the case of multiple such reference points, one is selected at random.

If $\rho_j = 0$, two scenarios are possible. First, there exists one or more solutions in F_l associated with reference point j . For this scenario, the individual in F_l with the shortest perpendicular distance to reference point j is added to P_{i+1} . The niche count for reference point j is then incremented by 1 and the previously selected individual from F_l is excluded from further consideration. The second scenario is that there are no individuals in F_l associated with reference point j . In this case, reference point j is excluded from further consideration.

If $\rho_j \geq 1$ and there exists one or more solutions in F_l associated with reference point j , then one is selected at random from F_l and added to P_{i+1} . The niche count of the reference point j is incremented by 1. If there are no individuals in F_l associated with the reference point j , then this reference point is excluded from further consideration. The procedure is repeated until the population set has been filled.

4 Experiments on many-objective optimization

The ability of NSDE-R to solve many-objective optimization problems is investigated on a total of 14 test problems with three, five, eight, 10, and 15 objectives. This section presents optimization problem formulations and algorithm setup used to validate the NSDE-R algorithm.

4.1 Test problems

The test problems used to assess the performance of NSDE-R are taken from the DTLZ (Deb et al. 2005) and WFG (Huband et al. 2006) test suites. Table 1 gives the properties of each of the test problems selected. A total of 14 test problems, featuring three, five, eight, 10, and 15 objectives were used. The total number of variables for the DTLZ cases are $N = M + k - 1$,

Table 1 Characteristics of DTLZ and WFG test problems

Test problem	Properties
DTLZ1	Linear, multi-modal
DTLZ2	Concave
DTLZ3	Concave, multi-modal
DTLZ4	Concave, biased
WFG1	Mixed, biased
WFG2	Convex, disconnected, multi-modal, non-separable
WFG3	Linear, degenerate, non-separable
WFG4	Concave, multi-modal
WFG5	Concave, deceptive
WFG6	Concave, non-separable
WFG7	Concave, biased
WFG8	Concave, biased, non-separable
WFG9	Concave, biased, multi-modal, deceptive, non-separable

where M is the number of objectives, and $k = 5$ for DTLZ1, while $k = 10$ for DTLZ2, DTLZ3, and DTLZ4. Then, number of input variables for the WFG suite were $N = k + l$, where $l = 20$ and $k = 2 \times (M - 1)$. The many-objective test functions used in this paper can be scaled to any number of objectives and any number of design variables. They provide a diverse set of objective function topologies that the optimization algorithm can be validated against. Their analytical Pareto fronts are also known, which allows for the convergence analysis of the algorithms.

4.2 Indicator of performance

This work utilizes two metrics of accuracy for two different test suites of problems. The two metrics both measure diversity and convergence of the algorithms.

4.2.1 Inverted generalized distance (Bosman and Thierens 2003)

For each of the previously stated DTLZ test problems, the analytical Pareto front is known. Because the reference point-guided algorithms drive the search towards the reference points, only the analytical Pareto points closest to the reference points, called targeted Pareto points, should be used to construct the accuracy measure. Since the analytical Pareto front is known, one can find the targeted Pareto point, Z_{tar} , by finding the intersection between the analytical Pareto front and the reference line connecting the reference points and the origin. The accuracy measure should then use these targeted Pareto points and the approximate Pareto point obtained from the optimization algorithms. This work makes use of the Inverted Generalized Distance (IGD) measure given by

$$IGD(P, Z_{tar}) = \frac{1}{|Z_{tar}|} \sum_{i=1}^{|Z_{tar}|} \min_{j=1}^{|P|} d(z_i, p_j) \tag{11}$$

where $d(z_i, p_j) = \|z_i - p_j\|_2$. The IGD metric measures both the convergence to the analytical Pareto front and the diversity of the Pareto solutions. The smaller the value of the IGD, the better the approximated Pareto solution.

4.2.2 Hypervolume (Zitzler and Thiele 1999)

For the WFG suite of test cases, the analytical Pareto front is often difficult to obtain. It is even more difficult to uniformly distribute the analytical Pareto solution on this analytical Pareto front. For this reason, a metric that does not require the analytical Pareto front to be known is used for the WFG test suite. Hypervolume measures the size of the objective function space dominated by solutions S and bounded by z^r , where z^r is a reference point dominated by all Pareto-optimal solutions. The hypervolume is computed as

$$HV(S) = Vol(\underset{x \in S}{[f_1(x), z_1^r]} \times \dots [f_M(x), z_M^r]) \tag{12}$$

where Vol(·) is the Lebesgue measure. A larger value of HV indicates a better converged and more diversified solution set.

The reference point used for each test problem is given in Table 2. The HV is normalized to [0,1] by diving by $z = \prod_{i=1}^M z_i^r$. For three to 10-objective test problems, the exact HV is calculated using the WFG algorithm (While et al. 2012). For test problems with the number of objectives greater than 10, the HV is approximated using Monte Carlo sampling (Bader and Zitzler 2011b). The sampling size was held constant, as recommended, at 10,000 (Bader and Zitzler 2011b).

4.3 Compared algorithms

The proposed NSDE-R algorithm is compared to two other state-of-the-art, popular, and powerful many-objective optimization algorithms: NSGA-III (Deb and Jain 2014) and the MOEA/DD (Li et al. 2015). They have been shown to outperform several indicator-based and other decomposition-based many-objective optimization algorithms. The basic principles of operation for these two algorithms are given below.

NSGA-III (Deb and Jain 2014) It is an upgraded version of the popular NSGA-II algorithm. NSGA-III uses the same crossover and mutation operators as NSGA-II, but makes use of reference points to preserve diversity using niche preservation.

MOEA/DD (Li et al. 2015) It is an extension of the MOEA/D algorithm. Unlike the original algorithm that only uses decomposition methods, the MOEA/DD algorithm also makes use of domination principle and was shown to converge faster to the analytical Pareto front for several test functions.

4.4 Parameter settings

The parameters used for each algorithm and each test problem are presented in this section. Table 3 shows the number of generations and the population sizes for the test problems. The number of reference points is equal to the population size. Each algorithm was run 20 times on each test problem.

Table 2 Hypervolume reference point used for each problem

Test problem	Reference point
DTLZ1	$(1.0, \dots, 1.0)^T$
DTLZ2 to DTLZ4	$(2.0, \dots, 2.0)^T$
WFG1 to WFG9	$(3.0, \dots, 2.0 \times M + 1.0)^T$

Table 3 Population size and generations for test problems

M	Population size	Number of generations		
		DTLZ {1,2}	DTLZ {3,4}	WFG
3	92	300	1000	390
5	212	500	1000	650
8	156	800	1000	1040
10	276	1000	1500	1300
15	136	1500	2000	1950

The algorithmic parameter settings for each algorithm are given below.

NSGA-III The crossover probability, p_c , and mutation probability p_m are 1.0 and $1/N$ respectively. The crossover distribution index η_c and mutation distribution index η_m are set to 30 and 20 respectively.

MOEA/DD The crossover and mutation parameters are the same as for the NSGA-III algorithm. The penalty parameter θ in PBI is set to 5.0. The neighborhood size, T , is set to 20 and the probability, δ , used to select in the neighborhood is chosen to be 0.9.

NSDE-R The scaling factor F in (4) is held constant at 0.5 and the crossover probability Cr is also held constant at 0.5. Both the “rand/1/bin” and “donor3” mutation operators will be investigated.

It can be observed that the NSDE-R algorithm requires significantly fewer parameters to be specified. Since the choice of mutation donor does not affect the number of parameter needed, the NSDE-R algorithm requires at most two parameters, as opposed to four required by NSGA-III and seven required by MOEA/DD. In the upcoming sections, the DE algorithm using “rand/1/bin” mutation donor will be referred to as NSDE-R, while the DE algorithm using the “donor3” mutation operator will be referend to as NSDE-D3.

5 Empirical results and discussions

5.1 Performance comparisons using DTLZ test suite

A comparison of the four algorithms on the four DTLZ test problems having a different number of objectives is presented. As previously mentioned, each algorithm was run 20 times on each test problem. The best, average, and worst case IGD values from the 20 runs are presented in Table 4. The shaded, bold cells indicate that the algorithm is superior to others for that particular test problem.

Table 4 shows that the NSDE family of algorithms is consistently on par if not better than the NSGA-III and MOEA/DD algorithms. For certain test cases, such as lower dimensional DTLZ1, the IGD values for the NSDE family differ by an order of magnitude from the NSGA-III and MOEA/DD. Moreover, Table 4 shows that NSDE-R consistently outperforms NSDE-D3 for DTLZ1 and DTLZ3, whereas NSDE-D3 outperforms NSDE-R in DTLZ2 and DTLZ4 test cases. This suggests that the “rand/1/bin” mutation is more robust for a multi-modal test problem and the “donor3” mutation is more robust for a concave test problem. Although NSDE-D3 outperforms NSDE-R for DTLZ2 and DTLZ4, it is not by much, as their performance measures are very similar. It can also be seen that the NSGA-III consistently performs better than the other three algorithms for the DTLZ4 problems which are of the concave, biased type. For the

DTLZ2 and DTLZ4 problems, the IGD values of all algorithms are of the same order of magnitude.

Table 5 shows the percent of total DTLZ test cases for which a particular algorithm was able to outperform the others. It can be seen that the NSDE family performs better than the other algorithms in a larger set of test cases. In the best case, average, and worst case scenario, the NSDE family outperforms the other two in 75%, 60%, and 55% of the test cases, respectively. It can be deduced that for the DTLZ test cases, the “rand/1/bin” mutation is more robust than the “donor3” mutation. If the “donor3” results are omitted, the “rand/1/bin” mutation outperforms the others in 62% of the test runs.

Figure 4 shows the variation of the IGD metrics obtained using the four algorithms for a random 10-objective DTLZ1 test problem run. It can be seen that at the beginning, the

Table 4 Best, average, and worst IGD values for the DTLZ test problems. Best performance is bold-italic

	M	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD		M	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD
DTLZ1	3	<i>2.204E-04</i>	7.516E-04	7.876E-03	1.115E-02	DTLZ3	3	<i>1.858E-03</i>	1.023E+00	6.834E-02	3.993E-02
		<i>3.044E-04</i>	3.253E-01	2.001E-02	1.292E-02			<i>5.210E-02</i>	7.651E+00	3.889E-01	1.408E-01
		<i>4.552E-04</i>	8.697E-01	3.741E-02	2.507E-02			<i>9.995E-01</i>	2.320E+01	1.048E+00	6.774E-01
	5	<i>3.904E-05</i>	1.414E-03	8.043E-03	3.287E-02		5	<i>5.664E-03</i>	2.108E-02	2.899E-01	8.287E-02
		<i>7.744E-05</i>	6.695E-02	2.081E-02	3.357E-02			<i>6.554E-03</i>	6.829E+00	4.677E-01	1.026E-01
		<i>1.529E-04</i>	2.628E-01	5.658E-02	3.456E-02			<i>7.628E-03</i>	2.253E+01	1.113E+00	1.622E-01
	8	<i>4.860E-03</i>	7.157E-03	1.154E-02	7.061E-02		8	<i>1.262E-01</i>	2.024E+00	4.585E-01	1.546E-01
		<i>1.220E-02</i>	2.890E-01	1.768E-02	7.249E-02			1.318E+00	9.243E+00	1.141E+00	<i>2.724E-01</i>
		7.219E-02	1.375E+00	<i>4.207E-02</i>	7.451E-02			5.361E+00	3.495E+01	3.205E+00	<i>1.144E+00</i>
	10	<i>5.684E-03</i>	7.599E-03	8.645E-03	9.255E-02		10	6.956E-02	<i>3.594E-02</i>	1.788E-01	8.240E-02
		<i>8.938E-03</i>	3.058E-01	1.476E-02	9.379E-02			4.096E-01	3.353E+00	2.705E-01	<i>9.451E-02</i>
		6.374E-02	8.825E-01	<i>3.436E-02</i>	9.539E-02			2.388E+00	8.870E+00	3.994E-01	<i>1.071E-01</i>
	15	<i>5.456E-03</i>	4.052E-02	9.922E-03	1.559E-01		15	<i>2.242E-02</i>	2.019E+00	1.943E-01	6.121E-02
		1.841E-01	4.936E-01	<i>2.640E-02</i>	1.629E-01			2.858E-01	8.114E+00	5.680E-01	<i>8.931E-02</i>
		2.850E+00	1.194E+00	<i>1.241E-01</i>	1.698E-01			2.626E+00	1.306E+01	1.430E+00	<i>2.737E-01</i>
DTLZ2	3	2.759E-03	<i>2.202E-03</i>	4.271E-03	3.606E-02	DTLZ4	3	1.360E-03	1.116E-03	<i>8.247E-04</i>	3.578E-02
		3.356E-03	<i>2.694E-03</i>	6.223E-03	3.680E-02			1.681E-03	<i>1.323E-03</i>	2.405E-01	3.617E-02
		3.865E-03	<i>3.085E-03</i>	1.604E-02	3.729E-02			2.052E-03	<i>1.736E-03</i>	5.332E-01	3.687E-02
	5	<i>5.181E-03</i>	6.307E-03	1.360E-02	5.312E-02		5	5.771E-03	4.883E-03	<i>4.740E-03</i>	5.299E-02
		<i>6.147E-03</i>	6.898E-03	1.783E-02	5.372E-02			6.512E-03	<i>5.487E-03</i>	6.902E-03	5.358E-02
		<i>7.251E-03</i>	7.665E-03	3.731E-02	5.485E-02			7.243E-03	<i>6.088E-03</i>	8.678E-03	5.431E-02
	8	2.190E-02	<i>1.735E-02</i>	3.417E-02	3.652E-02		8	4.054E-02	2.670E-02	<i>2.113E-02</i>	3.813E-02
		2.470E-02	<i>2.039E-02</i>	4.181E-02	3.775E-02			4.378E-02	3.027E-02	<i>2.446E-02</i>	4.145E-02
		2.765E-02	<i>2.213E-02</i>	5.358E-02	3.925E-02			4.800E-02	3.414E-02	<i>2.726E-02</i>	4.438E-02
	10	2.622E-02	<i>2.300E-02</i>	4.846E-02	3.543E-02		10	3.890E-02	3.038E-02	<i>2.215E-02</i>	3.834E-02
		2.844E-02	<i>2.512E-02</i>	5.452E-02	3.629E-02			4.123E-02	3.176E-02	<i>2.625E-02</i>	4.022E-02
		3.038E-02	<i>2.694E-02</i>	6.513E-02	3.722E-02			4.338E-02	3.329E-02	<i>2.880E-02</i>	4.188E-02
	15	1.602E-02	1.376E-02	5.761E-02	<i>1.294E-02</i>		15	1.992E-02	<i>1.918E-02</i>	2.056E-02	1.418E-02
		1.770E-02	1.490E-02	6.467E-02	<i>1.485E-02</i>			2.143E-02	1.990E-02	2.309E-02	<i>1.653E-02</i>
		1.952E-02	<i>1.593E-02</i>	8.157E-02	1.625E-02			2.298E-02	2.103E-02	2.510E-02	<i>1.797E-02</i>

Table 5 Percent of DTLZ cases that each algorithm outperforms others (best performance is bold-italic)

	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD
Best	50	25	20	5
Average	35	25	15	25
Worst	25	30	25	20

convergence rate of the NSDE-R is comparable to that of MOEA/DD. However, the NSDE-R algorithm not only maintains its higher convergence rate, but also arrives at a much lower IGD value than the other three algorithms. The plateau in IGD metric experienced by the MOEA/DD and the NSDE-D3 algorithms seen here and also in (Deb and Jain 2014) indicates convergence to a local Pareto front, whereas the NSDE-R and NSGA-III algorithms are able to escape this local Pareto front.

Figure 4 b shows the IGD convergence for the DTLZ2 test problem. It can be seen that the IGD values of all four algorithms continually decreases. DTLZ2 is a concave problem with no local Pareto front. Again the IGD values of NSGA-III and MOEA/DD follow similar pattern as the one presented in (Deb and Jain 2014). Figure 4 b shows that performances of NSDE-R and NSDE-D3 are almost indistinguishable from each other for this test case.

Figure 4 c shows the IGD convergence for a 10-objective DTLZ4 run. It can be seen that the IGD of all four algorithms is monotonically decreasing as it did for the DTLZ2 test problem. It should be noted that both DTLZ2 and DTLZ4 are concave test problems, and that DTLZ3 would have similar behavior to DTLZ1 test problem as they are both multi-modal. It can be seen that the NSDE family of algorithms consistently leads to similar IGD values if not lower. They also feature a higher convergence rate than the NSGA-III and MOEA/DD algorithms.

Having assessed the convergence to the analytical Pareto front, the diversity of the Pareto solutions obtained using the four algorithms is now investigated.

Figure 5 shows the parallel value plot obtained using the four algorithms on a random run of a 10-objective DTLZ1 test problem. It can be seen that the NSDE-R and NSGA-III are both able to find the minimum and maximum values of each objective. The NSDE-D3 and MOEA/DD, however, were not able to find the limits of each objective function. Comparing Fig. 5a and Fig. 5b it can be seen that the “donor3” mutation does not perform well for a multi-modal test function. Similar conclusion can be made about the MOEA/DD algorithm by comparing Fig. 5b and Fig. 5d. Figure 5 shows that NSDE-R and NSGA-III are able to converge well to the analytical Pareto front while maintaining diversity. This is also evident in the IGD values given in Table 5.

The parallel value plot obtained for a random run of a 15-objective DTZL3 test problem is shown in Fig. 6. The DTZL3 problem combines the multi-modal feature of DTLZ1 and the concave property of DTLZ2.

Figure 6 a shows that the NSDE-R algorithm is again able to both converge well and maintain solution diversity for a high dimensional, multi-modal, concave problem. The NSDE-D3, NSGA-III, and MOEA/DD all fail to fully converge and preserve diversity for this challenging test problem. This reaffirms that NSDE-R can outperform the other algorithms in a broad range of test problems that have various properties, whereas the NSDE-D3 performs well only on concave problems.

Figure 7 shows the parallel value plot obtained using the four algorithms on a random run of a 10-objective DTLZ4 test problem. All four algorithms were able to find the maximum and minimum values of each objective. It can be seen that both the “donor3” mutation and MOEA/DD perform well for concave test problems. For the DTLZ4 problem, all four algorithms produced evenly distributed solutions with MOEA/DD having the most uniformly distributed solutions.

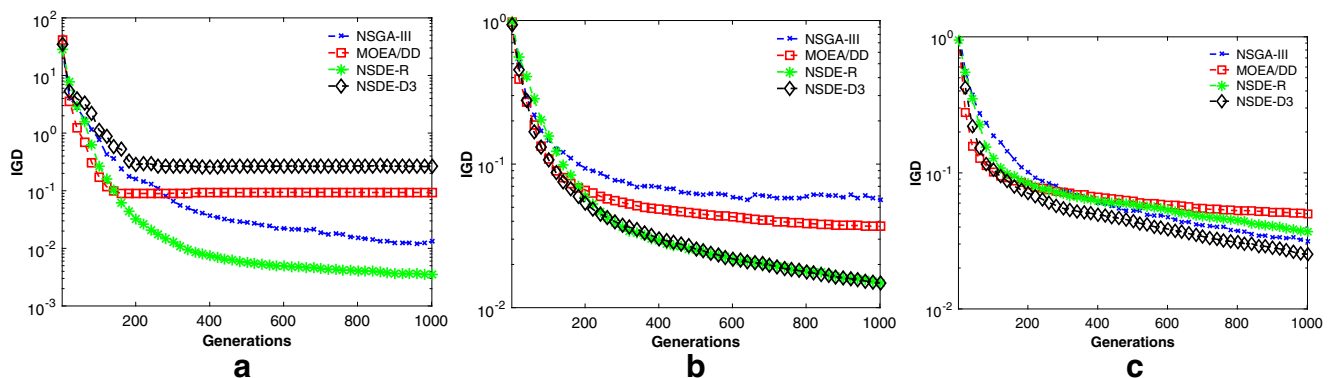


Fig. 4 Convergence histories: variations of IGD with the four algorithms for a random 10-objective run of **a** DTLZ1 test problem, **b** DTLZ2 test problem, and **c** DTLZ4 test problems

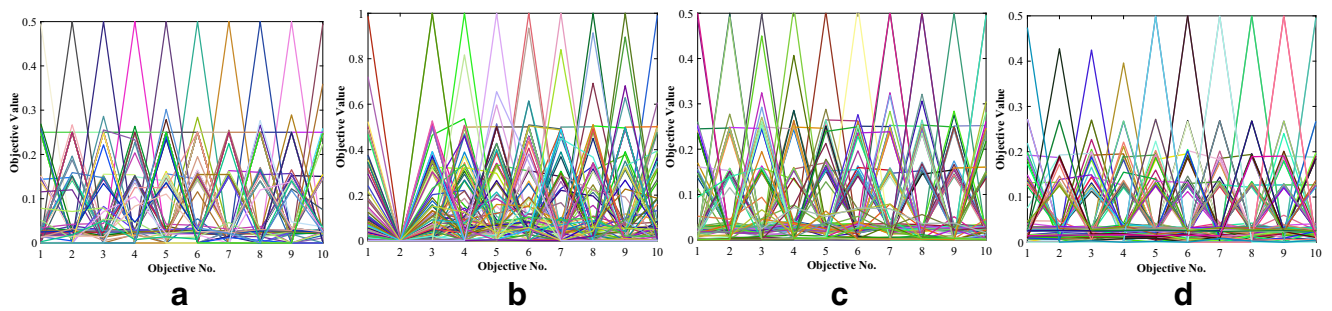


Fig. 5 Parallel value plots obtained for a random run of a 10-objective DTLZ1 test problem using **a** NSDE-R, **b** NSDE-D3, **c** NSGA-III, and **d** MOEA/DD

5.2 Performance comparisons on the WFG test suite

The WFG test suite combines several properties of objective function space (non-separability, multi-modality, biasing) into a single test problem, making it very difficult for algorithms to obtain well-converged and well-distributed Pareto solutions.

A comparison of the four algorithms tested on the nine WFG test problems having a different number of objectives is presented. As previously mentioned, each algorithm was run 20 times on each test problem. The best, average, and worst case HV values from the 20 runs are presented in Table 6. The shaded, bold cells indicate that the algorithm is superior to others for that particular test problem.

WFG1 test case investigates the algorithms' performance for biased and mixed Pareto fronts. It can be seen that for lower dimensional WFG1 problems, the NSGA-III algorithm, performs best, but for higher dimensional problems, the NSDE-D3 algorithm performs better. It should be noted that although NSGA-III performs well for lower dimensional problems, the performance of the NSDE-D3 is also just as good. The NSDE-R, however, is not able to cope with the difficulties of the WFG1 test function. It should be mentioned however, that the other algorithms are only slightly better.

WFG2 test case investigates the algorithms' ability to cope with convex, disconnected, multi-modal, and non-separable problems. The two DE algorithms perform best for the WFG2 problem. For lower dimensional problems, the performance of NSDE-R is comparable to that of NSDE-D3, but for higher dimensional problems, the NSDE-R performs significantly better.

WFG3 test case represents a connected version of WFG2. Its features are linear, degenerate, and non-separable. For this problem, the NSDE-D3 performs better for lower dimensional problems, while NSGA-III performs better for higher dimensional problems. The NSGA-III algorithm is more robust for the WFG3 test problem. The NSDE-R algorithm is the least robust for the WFG3 test case. This suggests that the "rand/1/bin" mutation is not ideal for degenerate type problems.

WFG4 to WFG9 test cases all have a hyperelliptic Pareto front with radii $r_i = 2i$, where $i \in \{1, \dots, M\}$. They are all concave problems featuring additional properties given in Table 1. The NSDE-D3 algorithm performs best on the WFG4 test problem, which features a concave, multi-modal space. The HV values for most algorithms are slightly below that of NSDE-D3 algorithm, but are still comparable.

The WFG5 test problem features a concave, deceptive landscape. The NSGA-III performs significantly better than

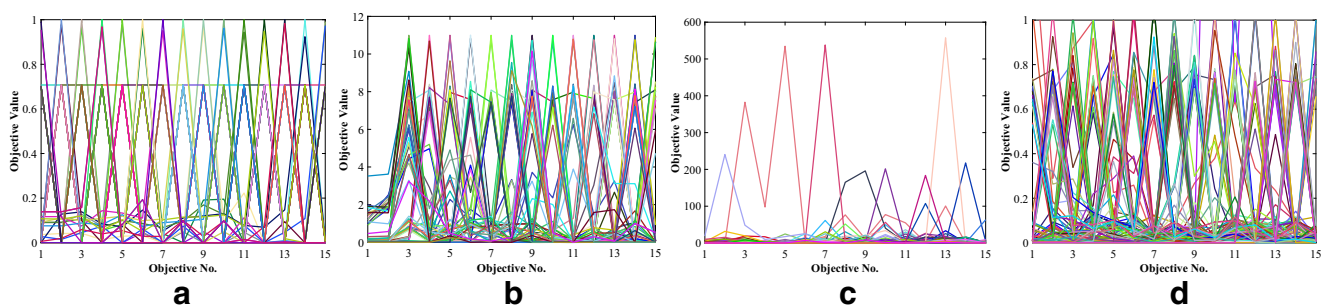


Fig. 6 Parallel value plots obtained for a random run of a 15-objective DTLZ3 test problem using **a** NSDE-R, **b** NSDE-D3, **c** NSGA-III, and **d** MOEA/DD

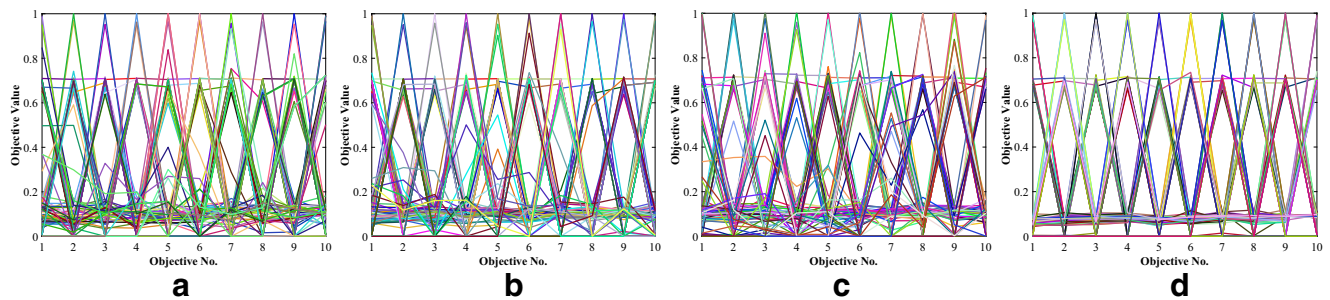


Fig. 7 Parallel value plots obtained for a random run of a 10-objective DTLZ4 test problem using **a** NSDE-R, **b** NSDE-D3, **c** NSGA-III, and **d** MOEA/DD

the other three algorithms for the WFG5 test function. The HV values for the NSGA-III are significantly and consistently higher.

WFG6 test case investigates the algorithms' ability to deal with concave and non-separable problem. The NSDE-D3 algorithm consistently performs better than the other three algorithms. The HV values of NSDE-D3 algorithm are significantly higher than those of the other three algorithms. The NSDE-R algorithm performs slightly worse than the NSDE-D3 algorithm. It performs slightly, but consistently better than the NSGA-III and MOEA/DD algorithms, especially for higher dimensional problems. This suggests both DE mutations are able to cope with non-separable problems.

WFG7 test case investigates the algorithms' abilities to deal with concave and biased problems. NSDE-D3 again performed better for most objectives. For five- and eight-objective problems, where the NSGA-III was better, the HV values of NSDE-D3 are on par with those for the NSGA-III algorithm. This shows that for the WFG7 test functions, the NSDE-D3 is significantly more robust. The NSDE-R performed consistently better than the MOEA/DD algorithm, but was less robust than NSGA-III for higher dimensional problems. This suggests that the "rand/1/bin" mutation is not able to cope with biased problems. This was also seen in the case of the DTLZ4 test case.

WFG8 test case investigates algorithms' ability to cope with concave, biased, non-separable problems. The NSGA-III algorithm performed consistently better than the other three algorithms. The HV values of the NSGA-III were also consistently and significantly higher than those of the other algorithms. The HV values of NSDE-D3 algorithm were only slightly lower than those of the NSGA-III algorithm and consistently higher than those of the MOEA/DD algorithm. The NSDE-R algorithm was again less robust than NSDE-D3 due to the biased nature of the test case.

In WFG9 test case, the NSDE-D3 performed consistently better. However, the HV values of the NSGA-III were also on par with NSDE-D3 for most objectives cases that were run. The HV values of the NSDE-R were also comparable to those of the NSDE-D3, but only for lower dimensional problems.

The NSDE-R algorithm performed consistently better than the MOEA/DD algorithm, but slightly worse than the NSGA-III algorithm. This is again due to the biased characteristic of the WFG9 test case.

Table 7 shows by how many percent of the total WFG test problems was a particular algorithm able to outperform the others. For each case, "best-case," "averaged-case," and "worst-cases," there were a total of 45 test instances (# test problems \times # of objectives). Meaning, for each of the nine WFG test cases, the percent was calculated over 45 samples. It can be seen that the "best-case scenario", the NSDE-D3 algorithm performs best in 56% of the test cases. In the "averaged-case" scenario, the NSDE-D3 is successful in 56% of the test cases. In the "best-case" and "averaged-case", the success rate of NSDE-D3 is significantly higher than that of the other three algorithms. In the "worst-case" scenario, the NSDE-D3 is again the best performing with a success rate of 44%.

It is interesting to note that, unlike in the DTLZ comparison (Table 5), the NSDE-R algorithm consistently underperforms for the WFG suite. This is primarily because the "rand/1/bin" mutation is unable to cope with the biased characteristic present in the large number of WFG test cases. The "donor3" mutation however performs significantly better in a large number of WFG test cases.

5.3 Performance comparisons of optimization algorithms on the DTLZ and WFG test suites

In the previous sections, Tables 4 and 6 showcased the performance of each algorithm on each test problem. This section presents a more condensed comparison between algorithms over all test cases. It presents the success rate of one algorithm over another over all test suites. This method of comparison provides a more conclusive result when comparing algorithms.

Table 8 shows the success rate for the four algorithms on the DTLZ test suite. It shows that the NSDE-R is comparable to NSDE-D3, with both having a success rate of almost 50% over each other. The NSDE-R algorithm is better in 72% and 65% of the DTLZ family of test problems over the NSGA-III and MOEA/DD algorithms, respectively. It can be deduced that the NSDE-R algorithm is more robust for the DTLZ test

Table 6 Best, average, and worst HV values for the WFG test problems. Best performance is bold-italic

	<i>M</i>	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD	<i>M</i>	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD	<i>M</i>	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD	
WFG1	3	0.452219	0.468852	0.468032	0.457662	WFG4	3	0.685769	0.696223	0.695180	0.696973	WFG7	3	0.712082	0.725241	0.703642
		0.448817	0.465777	0.465277	0.453368			0.684098	0.692618	0.690607	0.692694			0.708498	0.721509	0.702097
		0.446124	0.461096	0.461650	0.448408			0.680072	0.688292	0.686611	0.681842			0.706263	0.716682	0.700901
	5	0.389792	0.400385	0.413669	0.407385		5	0.806563	0.828379	0.818419	0.815723		5	0.814635	0.853795	0.841765
		0.387939	0.398334	0.409981	0.405991			0.799895	0.823489	0.813538	0.809527			0.805741	0.840524	0.838946
		0.386527	0.395960	0.407630	0.404935			0.792892	0.815560	0.806783	0.804778			0.792449	0.830897	0.835923
	8	0.319102	0.368551	0.345378	0.336671		8	0.825135	0.869355	0.841991	0.778617		8	0.803177	0.904297	0.882251
		0.317054	0.351522	0.342516	0.332115			0.813378	0.852439	0.833469	0.763290			0.778917	0.878615	0.879082
		0.315236	0.335205	0.339182	0.328509			0.796430	0.818828	0.819638	0.746090			0.753859	0.857435	0.870957
	10	0.291193	0.330705	0.316374	0.325462		10	0.836362	0.886244	0.862630	0.769994		10	0.806616	0.928749	0.905092
		0.287712	0.322897	0.310890	0.318779			0.820205	0.875114	0.846037	0.754488			0.793054	0.912277	0.899592
		0.284884	0.313551	0.306208	0.312417			0.807270	0.856113	0.818566	0.744195			0.766691	0.900557	0.893187
	15	0.247224	0.312288	0.280296	0.270620		15	0.866386	0.879724	0.833486	0.669051		15	0.895333	0.949736	0.918416
		0.243842	0.293603	0.275800	0.262069			0.849044	0.860790	0.811611	0.580780			0.855012	0.936655	0.910528
		0.240113	0.278297	0.271296	0.256539			0.831795	0.828119	0.790643	0.429049			0.830685	0.921597	0.903723
WFG2	3	0.941011	0.950158	0.935680	0.927358	WFG5	3	0.679108	0.675167	0.679728	0.682045	WFG8	3	0.656430	0.665253	0.656337
		0.936407	0.942485	0.903059	0.915260			0.673513	0.666707	0.670113	0.666402			0.647277	0.659027	0.654106
		0.931765	0.931832	0.790975	0.777228			0.665807	0.656797	0.664833	0.626379			0.641700	0.653242	0.651581
	5	0.963424	0.982961	0.971267	0.922731		5	0.769813	0.761009	0.664833	0.781172		5	0.721195	0.754740	0.767523
		0.958969	0.974097	0.967454	0.918439			0.758242	0.727831	0.789640	0.776863			0.711423	0.741073	0.763312
		0.951908	0.962936	0.964658	0.913757			0.749508	0.706141	0.783963	0.768613			0.703842	0.730634	0.759300
	8	0.998827	0.965251	0.965774	0.940915		8	0.640077	0.650971	0.810512	0.703702		8	0.667313	0.745036	0.775683
		0.996824	0.945081	0.951246	0.919159			0.610570	0.595382	0.800499	0.679063			0.646544	0.713551	0.760874
		0.992222	0.790901	0.784355	0.777765			0.570938	0.552502	0.785940	0.662105			0.634328	0.688835	0.738869
	10	0.998643	0.984896	0.966711	0.951029		10	0.685900	0.627387	0.810574	0.667883		10	0.648964	0.757165	0.786862
		0.995627	0.967669	0.958379	0.942205			0.631352	0.580525	0.790145	0.645806			0.629842	0.738185	0.777392
		0.984116	0.805015	0.943400	0.935832			0.587933	0.548867	0.772162	0.611509			0.616772	0.716261	0.759196
	15	0.998679	0.966573	0.968033	0.938723		15	0.692861	0.643415	0.752379	0.502300		15	0.667003	0.755032	0.807141
		0.951815	0.841819	0.926075	0.921900			0.654515	0.600068	0.730119	0.420527			0.626959	0.715426	0.782751
		0.899467	0.765856	0.753725	0.786708			0.628793	0.558090	0.700447	0.353484			0.473204	0.678126	0.752667

Table 6 (continued)

	M	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD	WFG6	M	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD	WFG9	M	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD
WFG3	3	0.659500	0.707537	0.682666	0.681734		3	0.671145	0.718970	0.672973	0.663672		3	0.637276	0.686780	0.671947	0.670948
		0.652899	0.694292	0.679276	0.641425			0.647020	0.714054	0.658651	0.650143			0.634849	0.678994	0.654048	0.663289
		0.643160	0.680793	0.673400	0.627905			0.607929	0.710584	0.653405	0.569467			0.632747	0.638538	0.635923	0.629125
	5	0.576762	0.661561	0.646745	0.657104		5	0.800502	0.860913	0.789874	0.782119		5	0.738342	0.784764	0.763974	0.751332
		0.565451	0.651947	0.633761	0.649620			0.775845	0.846799	0.783355	0.768886			0.734697	0.747623	0.738439	0.743008
		0.552606	0.637258	0.617894	0.641002			0.647369	0.826234	0.775551	0.760472			0.728665	0.732602	0.720227	0.707286
	8	0.518065	0.581892	0.598018	0.575689		8	0.847089	0.908025	0.851149	0.739190		8	0.688575	0.738082	0.737084	0.639353
		0.483127	0.526631	0.547223	0.561891			0.842058	0.892929	0.826540	0.713018			0.675645	0.715734	0.715325	0.620527
		0.418636	0.295833	0.509021	0.551503			0.827882	0.868289	0.807703	0.687609			0.656189	0.693662	0.688842	0.571713
	10	0.550847	0.610048	0.618992	0.555823		10	0.860765	0.930200	0.853504	0.713763		10	0.706871	0.738506	0.747845	0.642828
		0.519403	0.550100	0.597712	0.548848			0.857093	0.920632	0.842502	0.696791			0.692179	0.722071	0.719549	0.616941
		0.492628	0.429930	0.555996	0.540403			0.851317	0.911472	0.833301	0.667941			0.676816	0.710280	0.701043	0.571569
	15	0.490015	0.525096	0.590452	0.219023		15	0.869762	0.952737	0.871725	0.549089		15	0.657392	0.716236	0.704363	0.479993
		0.458003	0.464023	0.557694	0.201414			0.867450	0.925610	0.850793	0.512737			0.639989	0.669889	0.675020	0.434122
		0.419618	0.366236	0.499510	0.184514			0.863619	0.881951	0.826170	0.406139			0.617690	0.644441	0.648289	0.330144

Table 7 Percent of nine WFG test cases that each algorithm performs best in (best performance is bold-italic)

	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD
Best	7	56	27	11
Average	9	56	29	7
Worst	11	44	40	4

suite as it is consistently better in more than 50% of the test problems over the other algorithms.

Table 9 shows the success rate of the four algorithms on the WFG test suite. The previously robust NSDE-R algorithm is the least robust on the WFG suite. The NSDE-D3 is better than NSDE-R in 80% of the test functions. The NSDE-D3 performs better in 61% and 79% of the WFG test problems over the NSGA-III and MOEA/DD algorithms, respectively. For the WFG family of test problems, NSDE-D3 has shown to be the most robust algorithm.

5.4 Reference points to aid multi-criteria-decision-making

Often in multi-criteria-decision-making (MCDM), the user is interested in finding solutions in a particular region of the Pareto front. It was previously mentioned that the reference point based approach can be used to guide the search to particular regions of interest. This section presents the results of reference point assisted MCDM on a three-objective and 15-objective DTLZ1 and DTLZ4 test problems obtained using the NSDE-R algorithm.

Figure 8 a shows the distribution of the reference points for three-objective problems. These reference points are biased towards the extreme values of each objective. Figure 8 b and c show the converged Pareto front obtained using these reference points and the NSDE-R algorithm. It can be seen that the search for both the multi-modal and concave problem is driven to the region of the reference points.

Figure 9 shows results of the reference point guided approach on 15-objective test problems. Figure 9 a shows

Table 8 Success rate of each algorithm over another for the four DTLZ test problems

	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD
NSDE-R	–	52	72	65
NSDE-D3	48	–	47	53
NSGA-III	28	53	–	48
MOEA/DD	35	47	52	–

Table 9 Success rate of each algorithm over another for the nine WFG test problems

	NSDE-R	NSDE-D3	NSGA-III	MOEA/DD
NSDE-R	–	20	23	55
NSDE-D3	80	–	61	79
NSGA-III	77	39	–	85
MOEA/DD	45	21	15	–

the distribution of the reference points on a unit hyper-plane. The reference points for this problem were biased

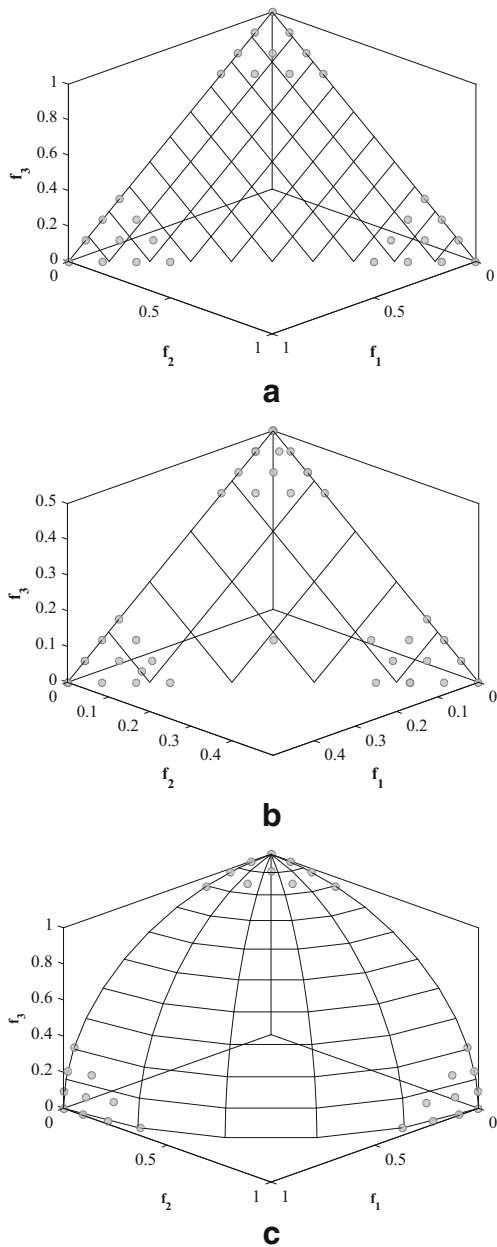


Fig. 8 Pareto fronts obtained using biased placement of reference points **a** for a three-objective test problem: **b** DTLZ1 problem and **c** DTLZ4 problem

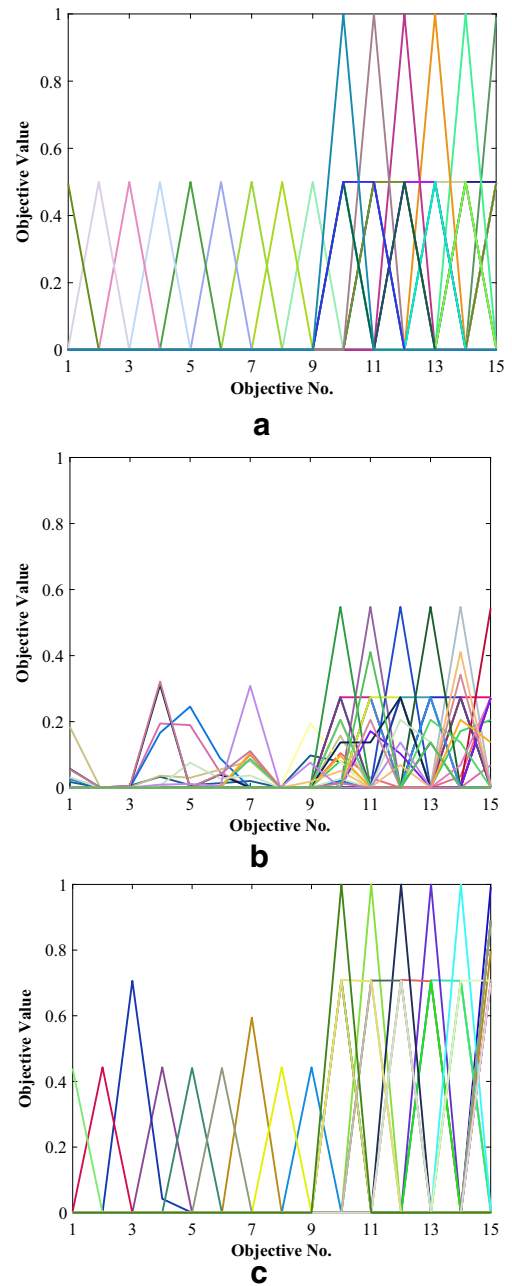


Fig. 9 Parallel value plot of Pareto fronts obtained using biased placement of reference points **a** for 15-objective test problem: **b** DTLZ1 problem and **c** DTLZ4 problem

towards the 15th objective. Figure 9 b and c show the distribution obtained on the DTLZ1 and DTLZ4 test problems. It can be seen that both algorithms were able to find the minimum and maximum values for each objective. For the DTLZ1 problem, the NSDE-R algorithm was able to drive the search in the region of the reference points and was also able to converge in that particular region. For the DTLZ4 problem, NSDE-R was again able to drive the search towards the reference points. It can also be seen

from Fig. 9c that the algorithm was able to converge not only to the region, but also to the analytical Pareto front.

Table 10 shows the IGD values obtained using NSDE-R on the two test problems for two different numbers of objectives. As previously stated, the IGD metric measures both convergence and diversity of a solution. It can be seen that the IGD values are relatively low, indicating that well-converged and well-diversified Pareto sets were reached in these test cases.

5.5 Randomly varying algorithm properties

All evolutionary algorithms are heavily influenced by the parameters used by them. Some of these properties are given in Section 4, D. It is very well known that the values of F and Cr greatly impact the search process of differential evolution algorithms. Often, users of the algorithm select these parameters either based on their experience or by trial-and-error (Storn and Price 1997). Others (Pedersen 2012) have proposed using meta-optimization to determine the values for these parameters. Some works (Das et al. 2008; Konar et al. n.d.) have proposed randomly varying the algorithm parameters within a specified bound. It was previously shown (Inclan and Dulikravich 2017) that for single-objective optimization, randomly varying these parameters significantly improves convergence. The effects of this adaption on the NSDE-R and NSDE-D3 for many-objective problems are investigated here.

The scaling factor F and crossover probability Cr will both vary uniformly within $[0.3, 0.7]$ and $[0.3, 0.7]$, respectively. The effects of random variation of the DE parameters were investigated on both the “rand/1/bin” and “donor3” mutation schemes. The algorithms were again run on the DTLZ test suite. The algorithms’ number of runs, number of generations,

population size, and number of reference points were kept the same as for the previous DTLZ runs.

Table 11 shows the best, average, and worst case IGD values for the DE algorithms with randomly varying algorithm properties. The IGD values of NSDE-R and NSDE-D3 refer to the previous case where F and Cr were both set at 0.5, while the NSDE-R* and NSDE-D3* are runs with randomly varying parameters. It can be seen that the NSDE-R and NSDE-D3 outperform their stochastic counterparts for the DTLZ1 and DTLZ3 problems, suggesting that the chaotic search pattern caused by random values of the parameter should be avoided for multi-modal problems. The values of NSDE-D3* performs well for the concave problems such as DTLZ2 and DTLZ4. That being said, the original NSDE-R and NSDE-D3 both produce similar results to their stochastic versions for concave problems.

It can again be seen that the NSDE-R algorithm is more robust than the other three algorithms since its performance is on par, if not better, for all test problems studied here. It is recommended that for many-objective problems, the DE parameters not be varied randomly each generation, especially for multi-modal functions, as this leads to an overly chaotic search. The cases with static values of algorithm parameters produce well-converged results in average and in worst case scenarios.

5.6 Performance of NSDE-R on problems with more objectives

It was previously shown that the NSDE-R algorithm is able to converge and preserve diversity in most of the test problems analyzed. The ability of NSDE-R to converge for problems with more objectives was then investigated on a 20-objective DTLZ1, DTLZ2, DTLZ3, and WFG4 test problems. These problems feature linearity, concavity, and multi-modality in their objective functions. The NSDE-R algorithm was run for 1500 generation of each of these test problems with a population size and reference point set size of 230.

The obtained Pareto front in parallel value plot format is shown in Fig. 10. It should be noted that the Pareto points plotted are not normalized and that they represent the actual value of the objective functions. It can be seen in Fig. 10a that for the linear, multi-modal DTLZ1 test case, the Pareto set obtained using NSDE-R are not only diverse, but are also well converged. The convergence is seen in the value of the objective functions, as the Pareto front of the DTLZ1 is a half-unit hyperplane.

The NSDE-R algorithm was also able to preserve diversity and converge well for the concave DTLZ2 test problem. The Pareto set of the DTLZ2 problem is a unit hypersphere. Figure 10 b shows that the NSDE-R algorithm is able to find the minimum and maximum values of each objective.

Table 10 Best, average, and worst IGD values obtained using biased reference points

	M	Max. gen	NSDE-R
DTLZ1	3	800	1.350E-03
			1.490E-03
	15	1500	6.700E-02
			1.883E-01
DTLZ4	3	300	2.420E-01
			4.060E-01
	15	1500	1.850E-03
			1.890E-03
			4.080E-03
			4.850E-01
			5.110E-01
			5.590E-01

Table 11 Best, average, and worst IGD values for the DTLZ test problems. Best performance is bold-italic

	<i>M</i>	NSDE-R	NSDE-D3	NSDE-R*	NSDE-D3*		<i>M</i>	NSDE-R	NSDE-D3	NSDE-R*	NSDE-D3*
DTLZ1	3	<i>2.204E-04</i>	7.516E-04	7.605E-04	7.758E-04	DTLZ3	3	<i>1.858E-03</i>	1.023E+00	3.069E-02	1.999E+00
		<i>3.044E-04</i>	3.253E-01	1.731E-01	3.206E-01			<i>5.210E-02</i>	7.651E+00	2.728E+00	8.088E+00
		<i>4.552E-04</i>	8.697E-01	5.810E-01	8.698E-01			<i>9.995E-01</i>	2.320E+01	6.749E+00	1.751E+01
	5	<i>3.904E-05</i>	1.414E-03	5.926E-04	1.565E-03		5	<i>5.664E-03</i>	2.108E-02	7.385E-03	1.999E+00
		<i>7.744E-05</i>	6.695E-02	1.389E-02	1.376E-01			<i>6.554E-03</i>	6.829E+00	1.113E-01	3.376E+00
		<i>1.529E-04</i>	2.628E-01	2.634E-01	4.728E-01			<i>7.628E-03</i>	2.253E+01	1.001E+00	6.275E+00
	8	<i>4.860E-03</i>	7.157E-03	5.124E-03	6.036E-03		8	<i>1.262E-01</i>	2.024E+00	1.506E-01	3.160E+00
		<i>1.220E-02</i>	2.890E-01	1.747E-02	3.991E-01			<i>1.318E+00</i>	9.243E+00	1.191E+01	1.289E+01
		<i>7.219E-02</i>	1.375E+00	2.389E-01	1.419E+00			<i>5.361E+00</i>	3.495E+01	2.673E+01	3.931E+01
	10	<i>5.684E-03</i>	7.599E-03	5.799E-03	6.823E-03		10	6.956E-02	<i>3.594E-02</i>	1.050E+00	8.475E-02
		<i>8.938E-03</i>	3.058E-01	1.495E-02	2.522E-01			<i>4.096E-01</i>	3.353E+00	8.661E+00	3.695E+00
		<i>6.374E-02</i>	8.825E-01	7.095E-02	8.456E-01			<i>2.388E+00</i>	8.870E+00	1.782E+01	1.623E+01
	15	5.456E-03	4.052E-02	5.625E-03	<i>5.426E-03</i>		15	<i>2.242E-02</i>	2.019E+00	2.157E+00	3.401E+00
		<i>1.841E-01</i>	4.936E-01	6.346E-02	3.918E-01			<i>2.858E-01</i>	8.114E+00	1.046E+01	1.025E+01
		2.850E+00	1.194E+00	<i>3.317E-01</i>	8.395E-01			<i>2.626E+00</i>	1.306E+01	3.519E+01	2.015E+01
DTLZ2	3	2.759E-03	<i>2.202E-03</i>	3.008E-03	2.447E-03	DTLZ4	3	1.360E-03	<i>1.116E-03</i>	1.498E-03	1.138E-03
		3.356E-03	<i>2.694E-03</i>	3.343E-03	2.805E-03			1.681E-03	<i>1.323E-03</i>	1.707E-03	1.394E-03
		3.865E-03	<i>3.085E-03</i>	3.899E-03	3.278E-03			2.052E-03	1.736E-03	2.098E-03	<i>1.547E-03</i>
	5	<i>5.181E-03</i>	6.307E-03	5.809E-03	6.694E-03		5	5.771E-03	<i>4.883E-03</i>	5.506E-03	5.179E-03
		<i>6.147E-03</i>	6.898E-03	6.722E-03	7.075E-03			6.512E-03	<i>5.487E-03</i>	6.828E-03	5.730E-03
		<i>7.251E-03</i>	7.665E-03	7.430E-03	7.813E-03			7.243E-03	<i>6.088E-03</i>	7.832E-03	6.469E-03
	8	2.190E-02	<i>1.735E-02</i>	2.029E-02	1.756E-02		8	4.054E-02	2.670E-02	3.809E-02	<i>2.548E-02</i>
		2.470E-02	2.039E-02	2.207E-02	<i>1.963E-02</i>			4.378E-02	3.027E-02	4.176E-02	<i>2.862E-02</i>
		2.765E-02	2.213E-02	2.407E-02	<i>2.207E-02</i>			4.800E-02	3.414E-02	4.514E-02	<i>3.103E-02</i>
	10	2.622E-02	2.300E-02	2.375E-02	<i>2.214E-02</i>		10	3.890E-02	3.038E-02	3.717E-02	<i>2.866E-02</i>
		2.844E-02	2.512E-02	2.566E-02	<i>2.334E-02</i>			4.123E-02	3.176E-02	3.910E-02	<i>2.967E-02</i>
		3.038E-02	2.694E-02	2.778E-02	<i>2.479E-02</i>			4.338E-02	3.329E-02	4.044E-02	<i>3.077E-02</i>
	15	1.602E-02	1.376E-02	1.562E-02	<i>1.215E-02</i>		15	1.992E-02	1.918E-02	1.948E-02	<i>1.801E-02</i>
		1.770E-02	1.490E-02	1.763E-02	<i>1.365E-02</i>			2.143E-02	1.990E-02	2.056E-02	<i>1.926E-02</i>
		1.952E-02	1.593E-02	1.964E-02	<i>1.508E-02</i>			2.298E-02	2.103E-02	2.217E-02	<i>2.050E-02</i>

Figure 10 c shows the Pareto front obtained for the concave, multi-modal DTLZ3 test problem. The Pareto sets for the DTLZ2 and DTLZ3 test cases were the same.

It can be seen that even in this difficult test problem, the NSDE-R is not only able to converge to the analytical Pareto front but also preserved population diversity. The proposed algorithm was able to find minimum and maximum values of each objective for all three test problems and therefore can be used for the estimation of the ideal point and the Nadir point.

The WFG4 test case also has the same properties as the DTLZ3 test function. The Pareto set of this problem is a hyperellipse. Figure 10 d shows that the NSDE-R algorithm was again able to converge while preserving diversity for this challenging problem, and that it performed well for 20 simultaneous objectives over a range of problem types. It can be seen that the maximum and minimum values of the Pareto

front found by NSDE-R define the hyperelliptic Pareto front of the WFG4 problem.

6 Crashworthiness design of vehicles for complete trade-off front

The crashworthiness problem was first described in (Liao and Li 2008) and aims at the optimization of the frontal structure of a vehicle. The design variables are the reinforced members around the frontal structure and the mass of the vehicle, deceleration during frontal crash, and toe-board intrusion in the offset-frontal crash were the three objective functions, all of which were to be minimized. The deceleration is directly proportional to biomechanical injuries cause to the occupants, while the toe-board intrusion accounts for the structural integrity of the vehicle. In

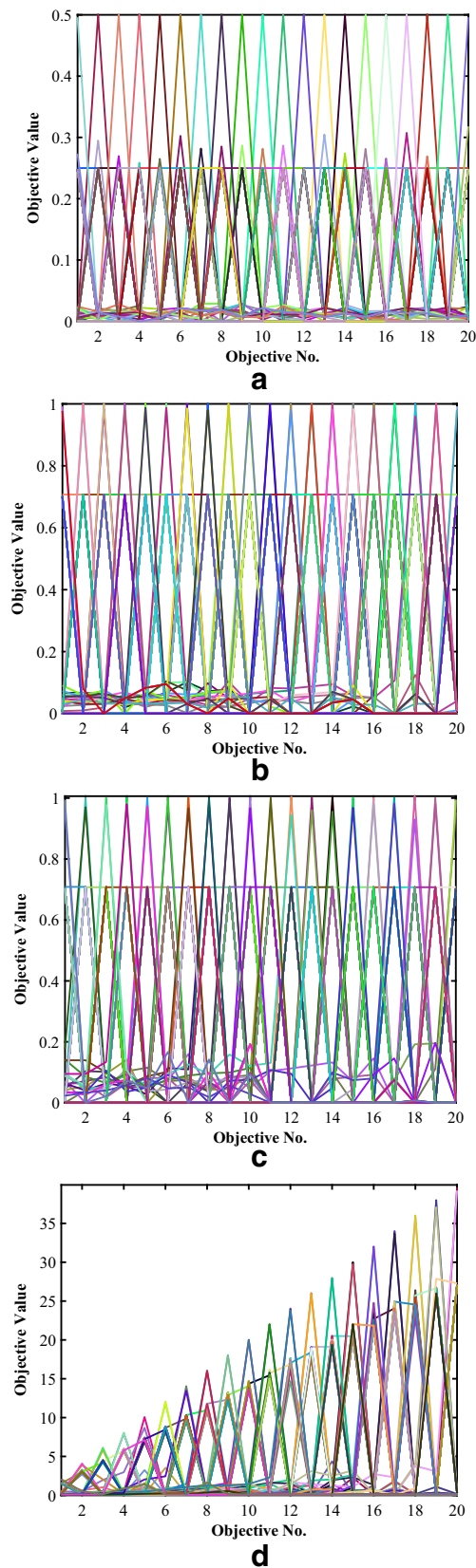


Fig. 10 Parallel value plots of Pareto fronts obtained using NSDE-R on a random run of a 20-objective problems: **a** DTLZ1, **b** DTLZ2, **c** DTLZ3, and **d** WFG4

(Liao and Li 2008), a design of experiments using Latin Hypercube and a full non-linear FEA analysis was performed and a stepwise regression model was fitted to the data. The optimization in this work and in (Deb and Jain 2014) was performed using this stepwise regression surrogate model. For this three-objective test problem, the same algorithm settings were used as for other three-objective problems. The total number of generations was set at 1000.

Figure 11 shows the Pareto front obtained using the four algorithms after 1000 generations. It can be seen that the NSDE-R and NSDE-D3 algorithms both produce uniformly distributed Pareto fronts, while the MOEA/DD produces the least diverse Pareto set. Comparing NSDE algorithms and the NSGA-III, it can be seen that both NSDE algorithms outperform the NSGA-III, since in Fig. 11c, a region of the Pareto front is shared by more than one design.

7 Conclusion

This work introduced two differential evolution algorithms for many-objective optimization. The proposed DE algorithms have been tested on problems from the DTLZ and WFG test suites with the number of simultaneous objectives ranging from three to 20. Both differential evolution algorithms use reference point based niching to preserve diversity among Pareto solutions. The proposed algorithms were compared against two state-of-the-art algorithms, NSGA-III and MOEA/DD. The following conclusions can be drawn from the performances on each test suite.

DTLZ It was seen that the differential evolution using “rand/1/bin” mutation performed best on the DTLZ test suite. The NSDE-R has shown to have a higher convergence rate than the other algorithms, converged well to the analytical Pareto front and was able to maintain diversity in the Pareto solution. The NSDE-D3 algorithm was able to converge well for concave, uni-modal problems, but failed for multi-modal test functions. The NSDE-R, however, performed well for both uni-modal and multi-modal problems. The NSDE-R algorithm performed better than NSGA-III in 72% of the test problems and better than MOEA/DD in 65% of the test problems. The NSDE-R algorithm was shown to be more robust when applied to the DTLZ suite of test problems.

WFG For this set of test problems, NSDE-D3 was shown to be more robust. It was shown that the NSDE-D3 algorithm performs better for higher dimensional problems, while its performance for lower dimensional problems is on par with the other three algorithms. It was again observed that the NSDE-D3 performed well for the concave, non-separable problems, but did not work well on the concave, biased, non-separable WFG8 test

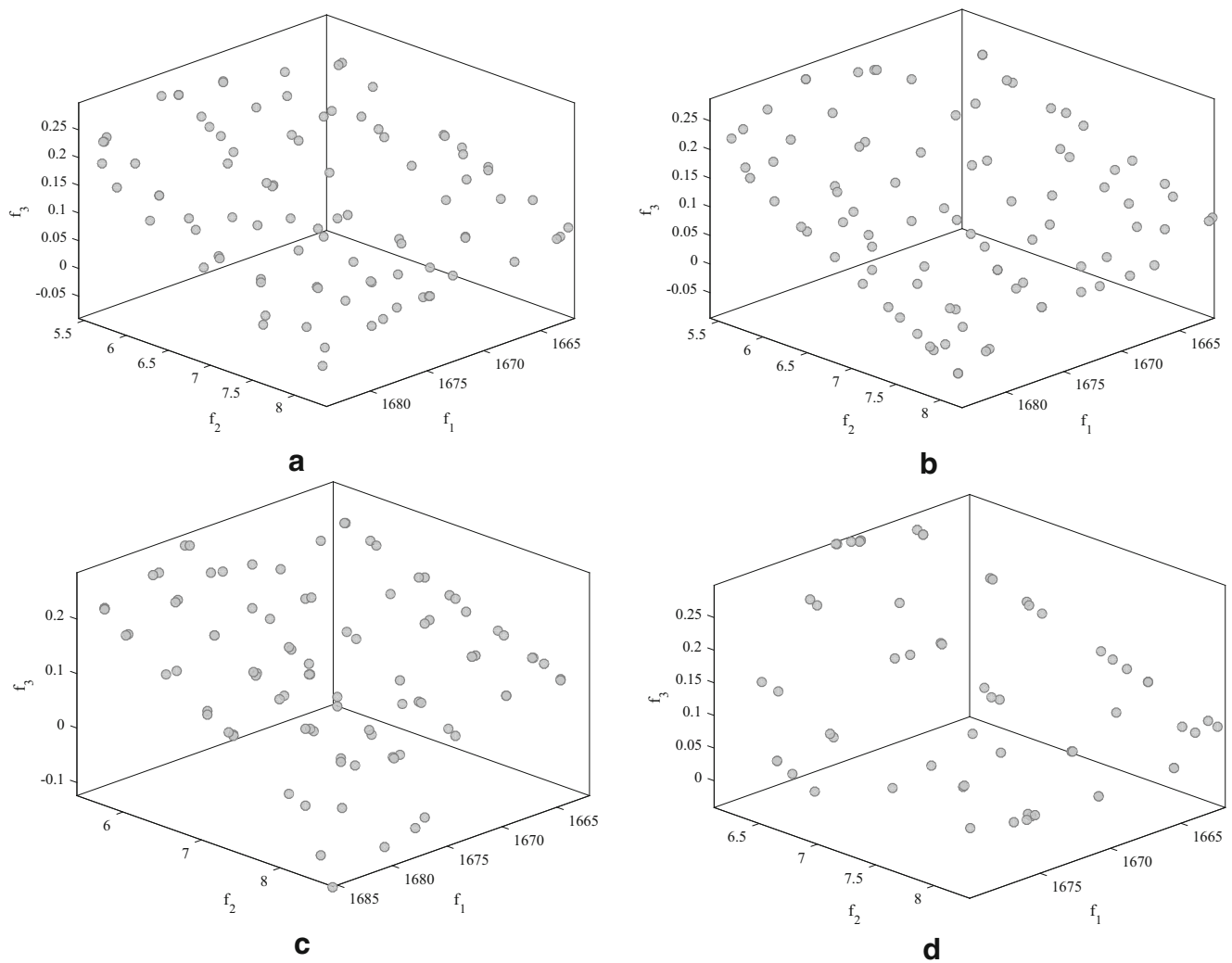


Fig. 11 Pareto front found for the crashworthiness problem using **a** NSDE-R, **b** NSDE-D3, **c** NSGA-III, and **d** MOEA/DD

problem. The NSDE-R algorithm was only able to solve the WFG2 test problem and lower dimensional WFG5 problem. It was observed that for the WFG test suite, the NSDE-R is the least robust algorithm. The NSDE-D3 outperformed NSDE-R in 80% of the WFG test cases. The NSDE-D3 algorithm also outperformed the NSGA-III and MOEA/DD algorithms in 61% and 79% of the test cases respectively.

It was shown that the DE algorithms can aid in multi-criteria-decision-making. The ability of the proposed NSDE-R algorithms to find a preferred part of the Pareto front was demonstrated. It was shown that the NSDE-R is able to guide the search towards the reference point even for high dimensional problems. This ability was showcased on both uni-modal and multi-modal, linear and concave test problems.

The effects of randomly varying algorithm parameters on the performance of the DE algorithms were also investigated. It was shown that the performance of the DE algorithm deteriorates for multi-modal problems when the parameters are

randomly varied. It was also noticed that the solution of the concave DTLZ2 and DTLZ4 problems are less sensitive to the variation in the algorithm parameters.

Finally, the performance of NSDE-R was investigated for high dimensional problems. Its performance on multi-modal, uni-modal, biased and concave test problems were investigated. Results showed that the NSDE-R algorithm is not only able to converge well for 20-objective problems but was also able to preserve diversity. The NSDR-R algorithm was also tested on a practical design problem for vehicle crashworthiness.

The two proposed DE algorithms have shown to be robust for a variety of test problems. They consistently perform better than the state-of-the-art NSGA-III and MOEA/DD algorithms.

Funding The first author gratefully acknowledges the financial support from Florida International University in the form of an FIU Presidential Fellowship and FIU Dissertation Year Fellowship. The authors are grateful for the research funding provided by the grant NSF CBET-1642253 monitored by Dr. Jose Lage, U.S. Army for the equipment grant W911NF-16-1-0494, and NASA grant NNX17AJ96A via TAMU.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Bader J, Zitzler E (2011a) HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evol Comput* 19(1):45–76
- Bader J, Zitzler E (2011b) HyPE: an algorithm for fast hypervolume-based many-objective optimization. *Evol Comput* 19(1):45–76
- Bosman P, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans Evol Comput* 7(2):174–188
- Chankong V, Haimes YY (1983) Multiobjective decision making theory and methodology. North-Holland, New York
- Chikumbo O, Goodman E, Deb K (2012) Approximating a multi-dimensional Pareto front for a land use management problem: a modified MOEA with an epigenetic silencing metaphor. *Proc CEC*:1–8
- Coello Coello CA, Lamont GB (2004) Applications of multi-objective evolutionary algorithms. World Scientific, Singapore
- Das S, Dennis JE (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J Optim* 8:631–657
- Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evol Comput* 15(1):4–31
- Das S, Abraham A, Konar A, Liu Y, Sun A, Loh H et al (2008) Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. In: *Advances of Computational Intelligence in Industrial Systems*, vol 116. Springer, Berlin / Heidelberg, pp 1–38
- Deb K, Agrawal RB (1994) Simulated binary crossover for continuous search space. *Complex Syst* 9:1–34
- Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. *Comput Sci Inform* 26:30–45
- Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: solving problems with box constraints. *IEEE Trans Evol Comput* 18(4):577–601
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: Abraham A, Jain L, Goldberg R (eds) *Evolutionary Multiobjective Optimization, series on advanced information and knowledge processing*. Springer, London, pp 105–145
- Denysiuk R, Costa L, Santo IE (2013) Many-objective optimization using differential evolution with variable-wise mutation restriction. In: *Proceedings of the GECCO'13*, pp 591–598
- Fan H, Liu J, Lampinen J (2010) Some improvement to the mutation donor of differential evolution. *Eng Comput* 27:225–242
- Figueiredo EMN, Luderemir TB, Bastos-Filho CJA (2016) Many objective particle swarm optimization. *Inf Sci* 374:115–134
- He X, Dai C, Chen Z (2014) Many-objective optimization using adaptive differential evolution with a new ranking method. *Math Probl Eng* 2014:1–8
- Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans Evol Comput* 10(5):477–506
- Inclan EJ, Dulikravich GS (2017) Demonstration of effective global optimization techniques via comparative analysis on a large analytical problem set. *Struct Multidiscip Optim* 55(1):179–204
- Jiang S, Yang S (2017) A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. *IEEE Trans Evol Comput* 21(3):329–346
- Konar A, Chakraborty UK, Das S Two improved differential evolution schemes for faster global search. In: Beyer H-G (ed) *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation (GECCO '05)*. ACM, New York, pp 991–998
- Li K, Deb K, Zhang Q, Kwong S (2015) An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans Evol Comput* 19(5):694–716
- X. Liao, Q. Li, Q. Zhang and X. Yang, “Multiobjective optimization for crash safety design of vehicle using stepwise regression model,” *Struct Multidiscip Optim*, vol. 35, No. 6, pp. 561–569, 2008
- Pedersen ME (2012) “Good parameters for differential evolution,” *Magnus Erik Hvass Pedersen*. Retrieved 2 2012, from Hvass Labs: <http://www.hvass-labs.org/people/magnus/publications.php>. Accessed 26 March 2018
- Robic T, Filipic B (2005) DEMO: differential evolution for multi-objective optimization. In: *Proc. 3rd Int. Conf. Evol. Multi-Criterion Optimization*, LNCS 3410, pp 520–533
- Storn R, Price K (1997) Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim*:341–359
- While RL, Bradstreet L, Barone L (2012) A fast way of calculating exact hypervolumes. *IEEE Trans Evol Comput* 16(1):86–95
- Wyss GD, Jorgensen KH (1998) “A user’s guide to LHS- Sandia’s latin hypercube sampling software,” Sandia National Labs, Rept. SAND98–0210, Albuquerque, NM
- Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: *Proc. 8th Int Conf Parallel Problem Solving from Nature*, pp 832–842
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions Evol Comput* 3(4):257–271

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.