Identifying Stochastically Non-dominated Solutions Using Evolutionary Computation

Hemant Kumar Singh[1,*] and Juergen Branke[2]

[1] The University of New South Wales, Canberra, Australia. h.singh@adfa.edu.au
[2] University of Warwick, Coventry, UK. juergen.branke@wbs.ac.uk
*Corresponding author

Abstract. We consider the problem of finding a solution robust to disturbances of its decision variables, and explain why this should be framed as problem to identify all stochastically non-dominated solutions. Then we show how this can be formulated as an unconventional multi-objective optimization problem and solved using evolutionary computation. Because evaluating stochastic dominance in a black-box setting is computationally very expensive, we also propose more efficient algorithm variants that utilize surrogate models and re-use historical data. Empirical results on several test problems demonstrate that the algorithm indeed finds the stochastically non-dominated solutions, and that the proposed efficiency enhancements are able to drastically cut the number of required function evaluations while maintaining good solution quality.

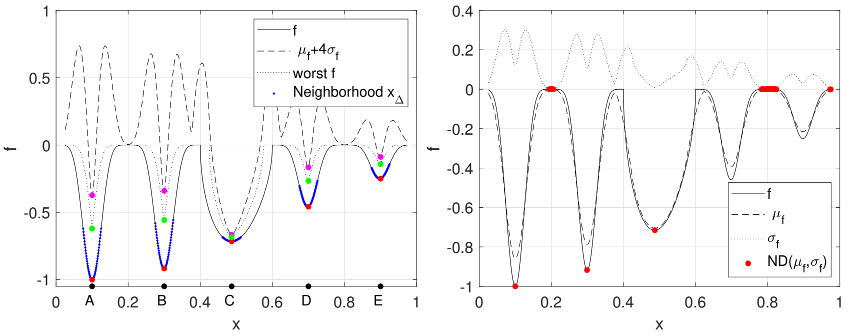
Keywords: Robust optimization · stochastic dominance · evolutionary algorithm

1 Background and Motivation

In some real-world environments, the decision variables are subject to disturbances before implementation, e.g., due to manufacturing tolerances [2]. In such cases, it is desirable that the solution is not only good, but also robust. Different definitions of robustness have been proposed in the previous literature:

1. the solution with the best expected performance despite the disturbances. This corresponds to a risk neutral decision maker. [13]
2. the solution with the best worst-case performance given the possible range of disturbances. This corresponds to a highly risk sensitive decision maker, willing to sacrifice expected performance for protection from risk. [2, 12]
3. the solution with the best weighted combination of expected performance plus w times the standard deviation σ. The larger the weight w on the standard deviation, the more risk averse this choice becomes. It has also been suggested to treat this as a multi-objective problem [10, 1, 14].

The above definitions can be intuitively understood by the illustration of a single-variable function shown in Fig. 1(a). The objective function is that of the TP3 problem in [13]. The deterministic function is shown with a solid black line.



(a) (b)

Fig. 1. Illustration of the existing robustness measures. The image of the points $A - E$ in the fitness landscape of deterministic, 4σ robustness and worst-case robustness are marked in red, magenta and green dots, respectively.

It is assumed that any given design x has an uncertainty uniformly distributed in $x_{\Delta} = [x - \Delta, x + \Delta]$ with $\Delta = 0.025$. Five points of interest (local optima) A, B, C, D and E have been marked in the design space. For each of these points, we show the region x_{Δ} by 51 uniformly sampled points, shown as blue dots. The resulting landscape of the robust formulation based on the worst case is shown as dotted line, the landscape for a mean plus 4σ robust formulation is shown as dashed line. Both robust formulations result in the solution C being identified as the robust optimum design. However, it can be seen that the distribution of objective values around design A yields (significantly) better performance under the given variations for some values of $x \in x_{\Delta}$. Even though the $\mu + 4\sigma$ value and the worst value obtained by the design C is better than that of A, design A yields a better or equal performance compared to C with an 88.23% probability (based on the uniform sampling shown)[3].

Moreover, the formulation based on mean plus variance may distort the fitness landscape in undesirable ways. If the uncertain region is slightly larger, say $\Delta = 0.05$, the fitness of solution A becomes even worse than the design $x = 0.2$, whereas the objective value around design A is *never* worse than the latter. Increasing the value of w would magnify the penalty associated with the standard deviation and a solution with extremely poor value but very low standard deviation (e.g. $x = 0.2, 0.8$) is considered equivalent to a solution with much better expected values but higher standard deviation (e.g., A, B, C). To remove the sensitivity of the results to the choice of w, some works have suggested optimizing the expected value and standard deviation as a bi-objective problem [10]. However, as shown in Fig. 1(b), the non-domination sorting based on μ and σ (of 951 uniform samples in $[x_{min} + \Delta, x_{max} - \Delta]$) would also yield several undesirable solutions that have poor objective value, on account of their low/zero variations.

[3] Note that these probabilities will change if the uncertainty does not follow a uniform random distribution; a scenario excluded from the scope of this work.

Also to note is that some of these solutions (e.g., again $x = 0.2, 0.8$) which have the worst possible objective value of $f = 0$, are preferred over the local minima D, E since the latter get dominated by another point (C) in the search space. The worst case formulation also masks the information regarding the better performance achieved within the variable uncertainties, as seen between the designs A and C. It also renders many of the designs indistinguishable in terms of their fitness (flat regions in Fig. 1). Optimizing the worst case performance is also a *bilevel optimization* which entails other characteristic challenges [9].

In order to overcome some of the shortcomings above, we propose a new way of defining robustness that does not depend only on expected or extreme values, but rather takes into consideration the distribution of the design performance more comprehensively. In particular, we propose to identify *all* solutions that are stochastically non-dominated. The concept of stochastic dominance is often used to compare or rank probability distributions [11]. For two probability distributions $g_A(x)$ and $g_B(x)$, the corresponding cumulative distribution $G_A(x)$ is said to first-order stochastically dominate $G_B(x)$ $(G_A(x) \preceq_{sd} G_B(x))$ if and only if the following inequality holds:

$$G_A(x) \leq G_B(x) \quad \forall x. \tag{1}$$

For any utility function $u(x)$ that is strictly increasing and piece-wise differentiable (which should be true for any rational DM), if $G_A(x) \preceq_{sd} G_B(x)$

$$G_A(x) \preceq_{sd} G_B(x) \Leftrightarrow \mathbb{E}_A(u(X)) \leq \mathbb{E}_B(u(X), \tag{2}$$

where $\mathbb{E}_A$ and $\mathbb{E}_B$ are the expectations over the probability distributions g_A and g_b, respectively. In other words, if we are able to identify all first-order stochastically non-dominated solutions, then we would be sure that among the identified solutions would be the most preferred solution for any rational decision maker, irrespective of their risk preferences.

Our paper is structured as follows. After formulating the problem in Section 2, we explain our baseline algorithm and strategies to reduce the number of function evaluations in Section 3. Empirical results are reported in Section 4. The paper concludes with a summary and some ideas for future work.

2 Proposed Problem Formulation

The proposed definition for robustness is based on the *quantile function (QF)* of the objective computed within the given uncertain region $\mathbf{x}_{\mathbf{\Delta}}$. This function defines, for each possible probability $p \in [0, 1]$ the fitness value that is obtained at least with that probability. More formally,

$$QF(x, p) = \inf\{y \in \mathbb{R} : p \leq G(f(x))\} \tag{3}$$

where $G(f(x))$ is the cumulative probability density function of the fitness value $f(x)$ of solution x given the uncertainty of the disturbance.

To identify all first-order stochastically non-dominated solutions, we are then solving the following optimization problem.

$$\min \qquad QF(x,p) \quad \forall p \tag{4}$$
$$\text{s.t.} \qquad x_i^L \leq x_i \leq x_i^U, \quad i = 1, \dots n_x. \tag{5}$$

Under the proposed definition, a solution x_A is considered better than another solution x_B if $QF(x_A)$ yields a lower or equal value than $QF(x_B)$ (for minimization) for all values of $p \in [0,1]$. This is equivalent to x_A first-order stochastically dominating x_B.

To understand the proposed measure intuitively, let us consider the QF functions of the solutions $A - E$ previously discussed, as shown in Fig. 2. A given point on the curve, say (0.5,-0.4043) of curve D can be interpreted as: 50% of the designs within the $\mathbf{x_\Delta}$ region of solution D have a better (lower) performance value than -0.4043. From the observed QF curves, it can be inferred that A dominates B, D, E, which means that for any quantile of fitness values A yields a lower fitness than either B, D or E. On the other hand, (A, C) and (B, C) are first order stochastically non-dominated pairs, implying that for each of the pair, there exists a monotonic utility function that would lead to this being the preferred solution. Thus, the set of first-order stochastically non-dominated solutions out of these points is identified as A and C.

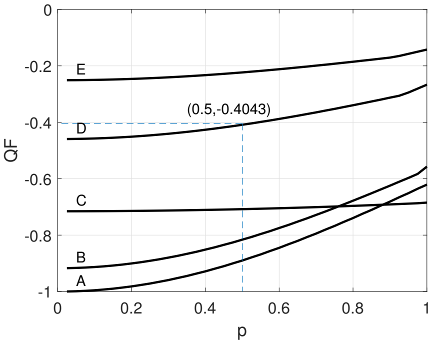


Fig. 2. QF function of solutions A, B, C, D, E in Fig.1

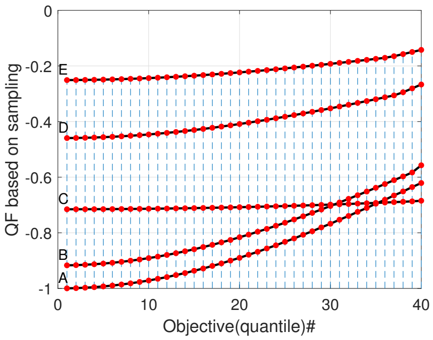


Fig. 3. Evaluating the quantiles (objectives)

Interestingly, the above formulation can be regarded as a multi-objective problem with *infinite* number of objectives. Regardless of the nature of the objective function $f(\mathbf{x})$, QF is mathematically a continuous function. Thus, first-order stochastic non-dominance is simply the non-domination criterion applied to compare two continuous functions. For practical implementation of the idea, we approximate QF by a finite (but large) set of objectives M, see below for more details. At the same time, it should also be noted that QF is a strictly non-decreasing function. This characteristic can be used to circumvent some of the scalability issues normally associated with non-domination based sorting for problems with large number of objectives [8].

3 Solution using an Evolutionary Algorithm

The basic framework of our proposed algorithm is quite similar to a canonical EA used to solve deterministic problems, but its components have been customized to deal with the proposed robust problem formulation. The algorithm assumes no prior information about the nature of the function, considering it as a black-box. For brevity, we refer to the first-order stochastic domination as FOS-domination in the following.

3.1 Discretization and evaluation of objectives

In many practical problems, the analytical form of the objective function is unknown. A viable method to *approximate* the quantile function would then be by sampling a finite number (say N_s) of designs within the uncertain region. Furthermore, to practically compare between different solutions and to represent them in a way that can be handled by EAs, a discretization of the quantile function itself is needed. We propose to do so by using M uniformly sampled values of p between 0 and 1. In order to evaluate a solution's performance, the quantile function value corresponding to the i^{th} value of p is assigned as its i^{th} quantile, where $i \in [1, M]$. This is illustrated in Fig. 3, where we chose $M = 40$ objectives and $N_s = 1000$ samples to construct the quantile function. Each vertical dotted line in the figure corresponds to an objective (denoted on the x-axis), and the red dots represent the corresponding robust objective values for a solution (read from the y-axis).

3.2 Parent selection and evolution operators

For evolving offspring, the widely used crossover and mutation operators, simulated binary crossover (SBX) and polynomial mutation (PM) [5] are used. Parents are selected from the current population by pairwise tournament selection. These mechanisms have been selected due to their widespread use in literature, but can be easily substituted with other evolutionary operators.

3.3 Dominance calculation and ranking

The process of FOS-domination ranking for a given set S containing N solutions and M objectives (quantiles) is outlined in Algo. 1 and the key steps are briefly described below.

Firstly, a distance matrix $\mathbf{d}$ is computed. Each element of the matrix d_{ij} denotes the minimum amount that needs to be added to *all* QF values of the solution i for it to be dominated by solution j (Line 3 in Algo. 1).

$$d_{ij} = \max\{\max_q\{f_q(j) - f_q(i)\}, 0\} \tag{6}$$

This quantity will correspond to the quantile in which solution i is better than j by the maximum amount. For example when comparing solution A with B in Fig. 3, $d_{AB} = f_1(B) - f_1(A)$. When comparing C and D, the maximum difference occurs in the 40^{th} objective, so $d_{CD} = f_{40}(D) - f_{40}(C)$. Note that this measure is structurally similar to additive ϵ indicator [16], but applied in quantile space instead of objective space.

Next, for each solution, $dMin$, the minimum of its distance values w.r.t. all other solutions is identified (Line 5). This is the minimum value that needs to be added to each objective of this solution to get dominated by *any* other solution in the set S, i.e., $dMin(i) = \min_{j \in S} d_{ij}$. Thus, in the example above, $dMin(A) = d_{AB}$ and $dMin(C) = d_{CA}$. Note that $dMin$ will be 0 for any solution that is dominated by another solution (B, D, E in this case).

The sequence of elimination is then determined in the Lines 7-17. The solution with the lowest $dMin$ represents the solution that can be dominated most easily, and is therefore added first to the elimination set. Then, the solution is removed, and all corresponding d values (row and column) are set to ∞. Thereafter, $dMin$ is updated, based on the updated $\mathbf{d}$ matrix. The solution with the lowest $dMin$ is again identified as the next solution to be added to the elimination list, and so on. Once all solutions have been added to the list, the order is reversed (Line 18), so as to rank the solutions from best to worst.

Note in the above ranking process that the dominated solutions are indistinguishable from each other, since all of them will have a $dMin = 0$. In order to obtain a full ordering, the FOS-domination ranking can be repeated only on solutions that achieved $dMin = 0$ in the first pass. The solutions that get $dMin = 0$ in the second pass can then be further segregated and ranked; until all solutions have obtained a distinct ranking. Equivalently, one can first do a non-domination sorting of the given solution set, and then apply FOS-domination ranking front-by-front.

3.4 Strategies to reduce computational effort

For the above algorithm, an adequate number of samples needs to be sampled in $\mathbf{x}_{\boldsymbol{\Delta}}$ to replicate the quantile function accurately. If the population size is N, the number of generations N_G and the number of samples evaluated in the vicinity of each solution $\mathbf{x}$ is N_s, then the total number of function evaluations (calls to the original function $f(\mathbf{x})$) can be calculated as $NFE = N \times N_G \times N_s$. In order to reduce the NFE, we propose two strategies below.

Use of approximation models: The use of surrogate models is prevalent in the literature for solving computationally expensive problems with stringent limits on NFE [15]. The basic idea is that based on a few available or prudently sampled designs, a surrogate model can be built and used to partially guide the

Algorithm 1 FOS-domination ranking

Input: Solution set $S = N \times M$ matrix, where N = No. of solutions to be ranked, M = No. of quantiles considered

```
for $i = 1$ to $N$ do
    for $j = 1$ to $N$ do
        Compute $d_{ij}$ according to Eq. 6
    end for
    $dMin_i = \min(d_{i,j}; j = 1 : N)$
end for
Initialize $ranklist = \emptyset$;
for $i = 1$ to $N$ do
    if $i \neq N$ then
        $j = \text{argmin}(dMin_j)$
    else
        $j = 1 : N - ranklist$ {Set difference}
    end if
    $ranklist = [ranklist\ j]$;
    $d(:, j) = \infty$; $d(j, :) = infty$
    $dMin_i = \min(d_{ij}; j = 1 : N)$
end for
Return final ranks $R$ = reverse(ranklist)
```

search in lieu of true evaluations. The true evaluation is then evoked only for relatively few solutions during the search that have been identified as promising based on the predictions from the surrogate model.

We use the Kriging model [4] to approximate the function $f(\mathbf{x})$, and by extension, the quantile function and associated quantiles in the neighborhood of any candidate solution $\mathbf{x}$. Instead of using a large sample size, say $N_s = 100$ points in $\mathbf{x}_{\boldsymbol{\Delta}}$, we use much fewer samples, say $N_{ss} = 10$. A Kriging model is built using the set of data $(\mathbf{x}, f(\mathbf{x}))$ such that for any unknown $\mathbf{x}$, the value of $f(\mathbf{x})$ can be predicted. The required number of samples ($N_s = 100$) are then extracted using this surrogate model to construct the quantile function based on *predicted* $f(\mathbf{x})$ values.

Re-using samples from neighboring solutions: Another way to reduce the computation is to reuse the previously evaluated samples that fall under the $\mathbf{x}_{\boldsymbol{\Delta}}$ of the solution currently under consideration. The sample and its fitness value can be inherited in such cases in lieu of evaluating a new sample. However, the number of available solutions could be unevenly distributed, and have larger or smaller size than the required number of samples N_s. This would adversely affect the quality of the surrogate built in the region, as consequently the quantile function and objectives. To counter this, we propose a simple strategy that augments the existing points (if any), with new samples required to reach the required number N_s, while maintaining relative uniformity between the samples.

The process is illustrated in Fig. 4. Suppose that the point currently under consideration is $x = 5.5$, let $N_s = 10$, and $\Delta = 0.5$. This implies that 11 points (including $x = 5.5$) need to be sampled uniformly in [5.0,6.0] to estimate the quantile values of the point. These are labeled as 'Ideal' points, shown with black dots. If two other points, $x = 5.18$ and 6.354 have previously undergone robustness evaluation, this means that 11 uniformly sampled solutions (each) are available in $x \in [4.68, 5.68]$ and $x \in [5.854, 6.854]$, respectively, shown as blue dots. We examine each of the uniformly distributed samples (black dots) and check if its closest existing sample (blue dot) is $\leq \frac{2\Delta}{N_s}$ away. If so, this original sample and its f value are directly used. If not, then the sample is evaluated instead. Moreover, the point under the robustness evaluation, i.e., $x = 5.5$ is evaluated unless an exact copy of it exists already. Thus, in this case only 2 samples needed to be evaluated (shown in red circles), whereas the remaining 9 samples are picked from an archive. It is also possible to use more sophisticated mechanisms to select the new sample locations, e.g., the one proposed in [6].

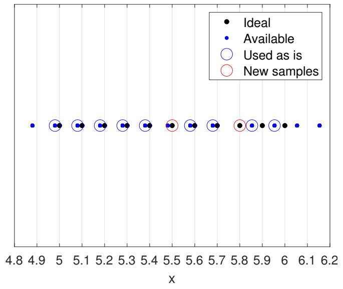


Fig. 4. Re-using the samples to reduce NFE

4 Numerical experiments

In this section, we evaluate the proposed approach on a range of benchmark problems. Please note that due to the space limitations, the results for the full set of problems are included in supplementary online material (SOM), which is available at http://www.mdolab.net/Hemant/Research-Data/ppsn22sup.zip; while only a few representative figures are included in this main manuscript.

4.1 Test problems

We demonstrate the proposed approach on the set of problems (TP1-9) formulated in [13]. Moreover, one problem, TP10, is additionally created for this study, and defined as $f = x \sin(2\pi x - \pi)$ with $x \in [0, 10]$ and $\Delta = 0.5$. The interesting feature of the problem is that (by design) the set of stochastically non-dominated solutions can be readily inferred from observation as $x = \{1, 2, \ldots 9\}$.

4.2 Experimental Setup

The algorithmic parameters used for solving the problems considered are given in Table 1. Four versions of the algorithm are used to solve each problem, configured by setting the use of surrogates and re-use of the previous samples as ON/OFF.

- V1: This is the baseline version, where both the surrogates and re-use of previous points is set to OFF.
- V2: Surrogates ON, re-use previous points OFF
- V3: Surrogates OFF, re-use previous points ON
- V4: Surrogates ON, re-use previous points ON

For each problem, 21 independent runs are conducted using each each algorithm variant. The quality of the resulting solutions are assessed visually as well as via unary metrics (discussed in next sub-section). In addition to the quality of solutions, the savings incurred in the cheaper versions (V2-V4) compared to the baseline (V1) version are also observed.

Table 1. Parameters used for the EA

Parameter	**Value**
Number of quantiles (M)	11
Population size (N)	20
No. of generations (G)	50
Crossover probability (p_c)	0.9
Mutation probability (p_m)	0.1
SBX Crossover index (η_c)	10
Polynomial Mutation index (η_m)	20
Neighborhood sampling points (N_s)	100 (1000 for TP10)
Reduced sampling size for surrogate-based versions (V3/V4) (N_{ss})	10

4.3 Performance measurement

In order to quantify the performance of the proposed algorithm and its variants, we resort to the inverted generational distance (IGD) metric[3]. IGD is commonly used in evolutionary multi-objective optimization for benchmarking the performance of algorithms. IGD compares the Pareto front (PF) approximation P obtained by an algorithm with a given *reference set* Q, which is the best estimate of the PF. Both sets P and Q refer to a set of points in the objective space. To compute the IGD, for each point in Q, the nearest point in P is identified and the corresponding Euclidean distance is recorded. Then, IGD is calculated as the mean of these distances; with a lower IGD indicating better performance.

For many of the standard benchmark problems, the true PF is known analytically, so a given number of points can be sampled on it to generate the reference set. If the true optimum of the problem under consideration may not be exactly known, a reference set is constructed e.g. by accumulating a large set of non-dominated solutions by combining solutions examined in multiple runs of all compared algorithms. Among the problems considered in this study, the theoretical optimum can be readily inferred only for three problems - TP1, TP7, and TP10. For TP1 and TP7, the function is monotonically decreasing in the range of $x = [2, 8]$. The only deterministic (global) optimum lies at $x = 8$, and the function value then steps up to 0 (its maximum value) thereafter. Therefore, in terms of stochastic non-dominance, $x = 8 - \Delta = 7.5$ is the true optimum solution for the problem. As for TP10, it is defined in a way as to have multiple peaks with the same periodicity but different, monotonically increasing, amplitudes. The Δ value chosen for the problem is 0.5, which is half the cycle of the function, thereby making 2Δ the full cycle. By observation, the points at the middle of the cycles, i.e., $x = \{1, 2, 3, \ldots 9\}$ therefore form the true optimum (stochastically non-dominated) solutions to the problem.

For the remainder of the problems, approximate reference sets have been generated by considering a set of uniformly sampled 1001 solutions within $\pm 0.5\Delta$ of their local and global optimum solutions. Then, the stochastically non-dominated solutions among these are considered to be the reference set.

4.4 Results

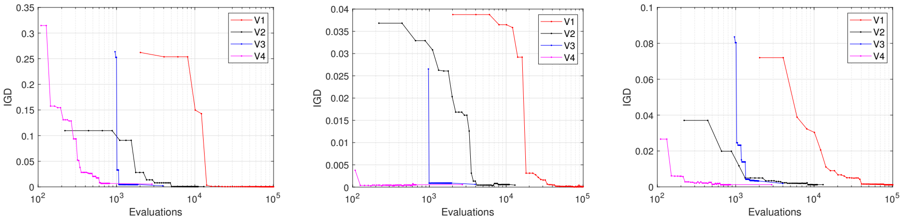
The median IGD values obtained using all variants of the proposed algorithms (V1-V4) are listed in Table 2, while the corresponding median function evaluations across 21 runs are listed in Table 3. Moreover, the convergence plots for the median runs for some representative problems are visualized in Fig. 5. Shown in Fig. 6 are the solutions obtained for TP3 in both x and quantile space; with the full set of problems included in the SOM Figs. 2-5.

Table 2. Median IGD values obtained by the proposed algorithm. The numbers in parenthesis denote the ratio of IGD compared to baseline (V*/V1), with ↑ or ↓ indicating the ratio to be higher or lower than 1, respectively.

Problem	V1	V2	V3	V4
TP1	0.0002	0.0006 (2.45× ↑)	0.0021 (9.23× ↑)	0.0054 (23.59× ↑)
TP2	0.0002	0.0004 (2.82× ↑)	0.0006 (3.93× ↑)	0.0006 (3.54× ↑)
TP3	0.0012	0.0012 (1.05× ↑)	0.0019 (1.62× ↑)	0.0012 (1.05× ↑)
TP4	0.0004	0.0004 (1.21× ↑)	0.0007 (1.90× ↑)	0.0004 (1.16× ↑)
TP5	0.0012	0.0022 (1.78× ↑)	0.0019 (1.61× ↑)	0.0023 (1.89× ↑)
TP6	0.0072	0.0065 (0.91× ↓)	0.0083 (1.15× ↑)	0.0061 (0.85× ↓)
TP7	0.1421	0.5141 (3.62× ↑)	1.3860 (9.75× ↑)	6.6902 (47.09× ↑)
TP8	0.0384	0.0395 (1.03× ↑)	0.0450 (1.17× ↑)	0.0436 (1.13× ↑)
TP9	0.0051	0.0051 (1.01× ↑)	0.0088 (1.73× ↑)	0.0055 (1.08× ↑)
TP10	0.0117	0.0124 (1.06× ↑)	0.0137 (1.17× ↑)	0.0124 (1.06× ↑)

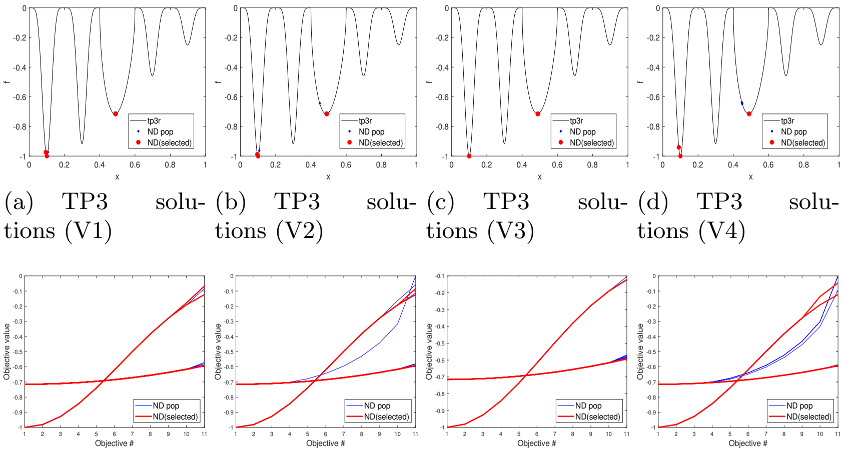
Table 3. Median function evaluations used by the proposed algorithm. The numbers in parenthesis denote the ratio of evaluations compared to baseline (V1/V*).

Problem	V1	V2	V3	V4
TP1	1.01e+05	13020 (7.76× ↓)	3721 (27.14× ↓)	2824 (35.76× ↓)
TP2	1.01e+05	13020 (7.76× ↓)	3958 (25.52× ↓)	2845 (35.5× ↓)
TP3	1.01e+05	13020 (7.76× ↓)	4000 (25.25× ↓)	2845 (35.5× ↓)
TP4	1.01e+05	13020 (7.76× ↓)	3241 (31.16× ↓)	2341 (43.14× ↓)
TP5	1.01e+05	13020 (7.76× ↓)	3984 (25.35× ↓)	2892 (34.92× ↓)
TP6	1.01e+05	13020 (7.76× ↓)	4869 (20.74× ↓)	3180 (31.76× ↓)
TP7	1.01e+05	13020 (7.76× ↓)	3742 (26.99× ↓)	2776 (36.38× ↓)
TP8	1.01e+05	13020 (7.76× ↓)	5315 (19.00× ↓)	3217 (31.4× ↓)
TP9	1.01e+05	13020 (7.76× ↓)	3927 (25.72× ↓)	2949 (34.25× ↓)
TP10	1.001e+06	31020 (32.27× ↓)	30849 (32.45× ↓)	21031 (47.6× ↓)



(a) TP1 (b) TP2 (c) TP3

Fig. 5. Convergence plots corresponding to the median IGD run



(a) TP3 solutions (V1) (b) TP3 solutions (V2) (c) TP3 solutions (V3) (d) TP3 solutions (V4)

(e) TP3 ND-pop (V1) (f) TP3 ND-pop (V2) (g) TP3 ND-pop (V3) (h) TP3 ND-pop (V4)

Fig. 6. Representative median IGD results obtained for TP3 using all versions (V1-V4) of the proposed algorithm. Results for all problems included in the SOM.

To provide context to Fig. 6, please note that there are two types of solutions shown. Those in blue represent the non-dominated solutions (based on quantiles) in the final population obtained after executing the EA run. However, some of these solutions are what is referred to in the literature as *dominance resistant solutions* (DRS)[7]. DRS are those that are significantly poor on one/some objective(s), but are non-dominated in the population due to a marginal improvement over another solution in one/some of the objectives. This particularly comes into play when the number of objectives is high, such as is the case here. To eliminate the DRS, we first normalise all objective values between the maximum and minimum values obtained among all objectives. Then, any differences between the normalized objective values that are less than 1% of the range are eliminated by rounding the values to two digits. These subsets of non-dominated solutions are shown in red color in the figures, and used for computation of the metrics.

From Table 2, it can be observed that the median IGD values are generally small, indicating that all four versions of the proposed algorithm were able to locate the correct regions of stochastically non-dominated solutions. The overall accuracy decreases successively when moving from V1 to V4. The % increase (V*/V1) in the IGD value is listed alongside the median IGD for each of the variants. The factors lie in the range of $\approx [1, 4]$. The notable exception to this are TP1 and TP7, for the versions V3 and V4, i.e. those that operate with surrogate-assistance. These two functions have their optimum exactly at $x = 7.5$, and at the edge of x_{Δ}, i.e, at $x = 8$ there is a significant discontinuity, stepping from the lowest to the highest value of the function instantaneously. However, given that the surrogate models assume a continuous function, the predicted step by the model at $x = 8.0$ will not be exactly vertical, leading to an overestimation of some quantile values. This also implies that any solution right of $x = 7.5$, even slightly, i.e., $x = 7.5 + \delta; \delta \to 0$ will have at least one quantile value as 0 (the highest value taken by the objective function). Note this, for example, for the population members (marked blue) in SOM Figs. 2-5 for TP7. The results for V3 and V4 of TP7 are also affected by the fact that the range of function values is very large ([-216,0]), so small errors will lead to large IGD values. A closer look at SOM Figs. 2-5 reveal that the solutions from the median run obtained by V3 and V4 are quite close to those obtained using V1 and V2. The same observations apply to TP1, as evident from SOM Figs. 2-5.

For other problems with multiple stochastically nondominated solutions, such as TP3, TP6 shown in Fig. 6, the algorithm shows commendable performance by identifying solutions in all the relevant regions. The same extends to other problems (shown in the SOM), with possible exception of TP8 where the solutions were found typically in 4 out of 5 regions in the median run. Reflecting back on Figs. 2-3, it can be seen that the algorithm converged to the two correct regions near points A and C - those with the non-dominated quantile functions among the multiple optima. Notably, the above solutions were obtained with significantly reduced number of evaluations compared to the baseline algorithm (V1/V*). The reduction in function evaluations is typically about 8-fold for V2 in the range of 20-40 folds for V3 and V4 to obtain solutions that are only marginally worse in quality compared to V1. Fig. 5 further provides a visualization of how quickly the computationally efficient variants of the proposed algorithm are able to converge relative to the baseline version.

5 Conclusions and future work

We proposed a new paradigm for black-box robust optimization, providing first order stochastically non-dominated solutions to a decision-maker. Towards this end, we formulated an underlying multi-objective optimization problem with discretized quantile functions and proposed an evolutionary approach to solve the problem. Since the process is computationally expensive in terms of NFEs consumed, strategies to reduce the NFEs substantially were also proposed, including the use of surrogate approximation and re-use of historical data. The results are encouraging and demonstrate the capability of the proposed algorithm in achieving the targeted solutions, as well as reducing the computational effort in doing so with relatively small compromise in solution quality.

In the future, we would like to make the proposed technique scalable for higher numbers of variables by using more efficient sampling methods, and extend the approach to deal with second order stochastic dominance. Also, the impact of the number of quantiles used for discretization and the density of samples used for performance estimation also needs further investigation to assess the proposed approach more comprehensively.

References

1. Asafuddoula, M., Singh, H.K., Ray, T.: Six-sigma robust design optimization using a many-objective decomposition-based evolutionary algorithm. IEEE Transactions on Evolutionary Computation **19**(4), 490–507 (2014)
2. Beyer, H.G., Sendhoff, B.: Robust optimization - a comprehensive survey. Computer Methods in Applied Mechanics and Engineering **196**(33-34), 3190–3218 (2007)
3. Coello, C.A.C., Sierra, M.R.: A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In: Mexican international conference on artificial intelligence. pp. 688–697. Springer (2004)
4. Couckuyt, I., Dhaene, T., Demeester, P.: oodace toolbox: a flexible object-oriented kriging implementation. Journal of Machine Learning Research **15**, 3183–3186 (2014)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
6. Fei, X., Branke, J., Gülpınar, N.: New sampling strategies when searching for robust solutions. IEEE Transactions on Evolutionary Computation **23**(2), 273–287 (2018)
7. Ishibuchi, H., Matsumoto, T., Masuyama, N., Nojima, Y.: Effects of dominance resistant solutions on the performance of evolutionary multi-objective and many-objective algorithms. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference. pp. 507–515 (2020)

8. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: A short review. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). pp. 2419–2426. IEEE (2008)
9. Islam, M.M., Singh, H.K., Ray, T.: A surrogate assisted approach for single-objective bilevel optimization. IEEE Transactions on Evolutionary Computation **21**(5), 681–696 (2017)
10. Jin, Y., Sendhoff, B.: Trade-off between performance and robustness: An evolutionary multiobjective approach. In: International conference on evolutionary multi-criterion optimization. pp. 237–251. Springer (2003)
11. Levy, H.: Stochastic dominance and expected utility: Survey and analysis. Management science **38**(4), 555–593 (1992)
12. Lu, K., Branke, J., Ray, T.: Improving efficiency of bi-level worst case optimization. In: International Conference on Parallel Problem Solving from Nature. pp. 410–420. Springer (2016)
13. Paenke, I., Branke, J., Jin, Y.: Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. IEEE Transactions on Evolutionary Computation **10**(4), 405–420 (2006)
14. Sun, G., Li, G., Zhou, S., Li, H., Hou, S., Li, Q.: Crashworthiness design of vehicle by using multiobjective robust optimization. Structural and Multidisciplinary Optimization **44**(1), 99–110 (2011)
15. Wang, G.G., Shan, S.: Review of Metamodeling Techniques in Support of Engineering Design Optimization. ASME Journal of Mechanical Design **129**(4), 370–380 (05 2006)
16. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: International conference on parallel problem solving from nature. pp. 832–842. Springer (2004)