Proceedings Paper:

Salomon, S., Purshouse, R.C., Giagkiozis, I. et al. (1 more author) (2016) A Toolkit for Generating Scalable Stochastic Multiobjective Test Problems. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference. GECCO 2016, 20 - 24th July, 2016, Denver, Colorado. ACM , pp. 597-604. ISBN 978-1-4503-4206-3

https://doi.org/10.1145/2908812.2908873

A Toolkit for Generating Scalable Stochastic Multiobjective Test Problems

S. Salomon
Automatic Control & Systems Engineering
University of Sheffield

R. C. Purshouse
Automatic Control & Systems Engineering
University of Sheffield

I. Giagkiozis
Automatic Control & Systems Engineering
University of Sheffield

P. J. Fleming
Automatic Control & Systems Engineering
University of Sheffield

ABSTRACT

Real-world optimization problems typically include uncertainties over various aspects of the problem formulation. Some existing algorithms are designed to cope with stochastic multiobjective optimization problems, but in order to benchmark them, a proper framework still needs to be established. This paper presents a novel toolkit that generates scalable, stochastic, multiobjective optimization problems. A stochastic problem is generated by transforming the objective vectors of a given deterministic test problem into random vectors. All random objective vectors are bounded by the feasible objective space, defined by the deterministic problem. Therefore, the global solution for the deterministic problem can also serve as a reference for the stochastic problem. A simple parametric distribution for the random objective vector is defined in a radial coordinate system, allowing for direct control over the dual challenges of convergence towards the true Pareto front and diversity across the front. An example for a stochastic test problem, generated by the toolkit, is provided.

Keywords

robust optimization; benchmark problems; multiobjective optimization

1. INTRODUCTION

Uncertainties are inherent to real-world optimization problems. The sources of uncertainty can be broadly divided into three categories: (i) changing or uncertain environmental parameters, (ii) inaccuracy of evaluation methods, and (iii) deviation of the actual solution from the identified nominal solution. Therefore, the *robustness* of a candidate solution to the various uncertainties needs to be considered in addition to the predicted performance under nominal conditions.

The field of *Robust Optimization* has been gaining increasing attention over the last two decades, including its extension to robust multiobjective optimization (RMO). In a robust optimization framework, a candidate solution is favoured if it can demonstrate high performance in the face of the various uncertainties. The objective and constraint functions are treated as random variables, and robustness metrics are used to assess and compare candidate solutions. Common robustness metrics are the worst case scenario, or aggregated indicators such as the expected value or variance [2].

The range of available algorithms for robust optimization is constantly growing. In order to use the right algorithm for a given application, it is important to clearly identify the strengths and weaknesses of the different candidate algorithms. This requires a benchmarking framework, similar to those developed in the field of evolutionary multiobjective optimization for deterministic problems. These frameworks include assessment measures [20], test problems [1, 4, 11, 17, 19] and an experimental setting [1, 6, 10]. To our knowledge, no such framework exists for RMO. This paper makes a first contribution toward the development of a framework by focusing on the test problem aspect. When benchmarking multiobjective optimizers, a variety of properties can be included in the test problems, including: geometry of the optimal set in decision and objective spaces, scalability, separability, modality, bias, deceptive and flat regions. When benchmarking robust multiobjective optimizers, these features should be retained whist adding additional properties relating to problem uncertainty.

There is a small literature on test problems for RMO. Typically, in these studies, symmetric random noise is added to the decision variables [3, 7, 14, 15, 16] or the objective functions [5, 9, 12, 18] in a bespoke way that meets the requirements for the specific uncertainties and definitions of robustness being considered. The problem that is used to benchmark every suggested algorithm is tailored to the type of uncertainty and definition of robustness considered in the study. For example, Deb and Gupta [3], Gaspar-Chuna et al. [7] and Mirjalili and Lewis [14, 15, 16] considered variation in decision variables, and therefore robustness is defined

GECCO '16, July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4206-3/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2908812.2908873

as either sensitivity of the objectives to variations in decision space, or as the averaged performance over a neighbourhood of the candidate solution. The problems developed in these studies are deterministic functions that demonstrate differences between the nominal Pareto front and the 'robust' front. Similarly, Goh and Tan [9], Knowles and Corne [12], Syberfeldt et al. [18] and Fieldsend and Everson [5] used deterministic test problems with added noise to the objective values to benchmark their algorithms. This type of problem is sufficient for the assumption that the 'true' objective vector is masked by measurement noise. The algorithm is required to overcome the noise and find the deterministic Pareto front.

The greatest contribution towards a general framework for creating RMO problems is the work of Goh et al. [8]. In their study, the authors identified some features required from an RMO test problem:

1. The existing challenges of a deterministic multiobjective test problem.
2. Tradeoffs between optimality and robustness.
3. Scalability of the stochastic components.

Additionally, a novel method to transform deterministic test problems into stochastic problems was presented in the study. The method is based on a parametric configurable noise function that can be inserted into different parts of the deterministic problem. In this manner, the stochastic problem can include uncertain decision variables, parameters and/or evaluation functions. However, Goh et al.'s method is an indirect approach to specifying the desired features. It requires a deep understanding of the deterministic problem to properly control the stochastic properties of the resulting function. The generalisability of the approach is therefore somewhat limited.

This study presents a new approach to RMO test problem generation that allows for direct control of the stochastic properties of the test problem, whilst retaining the benefits of re-use of existing, well-designed and familiar deterministic test problems. As part of this approach, Goh et al.'s list of requirements for RMO problems is extended. The following uncertainty characteristics need to be controlled in order to create meaningful benchmark problems:

1. Distribution of the uncertain objective vector. This includes correlation between the objectives, central tendency, variance, skewness etc.
2. Constant or changing distribution for different solutions.

In case the distribution is not constant, the way it changes dictates additional properties:

3. Increasing/decreasing uncertainty towards more optimal regions.
4. Separate contribution for the uncertainty of individual decision variables and/or objectives.
5. Changes of the uncertainty due to interacions between decision variables and/or objectives.

Additionally, these requirements should also be addressed by the benchmark problem:

6. Existence of a similar deterministic problem to serve as a well-understood reference for optimal performance.
7. Confinement of the random objective vectors to the feasible objective space of the deterministic problem.

Taking inspiration from the *WFG toolkit* developed for deterministic multiobjective problems [11], this study encapsulates a novel toolkit for generating scalable, stochastic, multiobjective optimization problems. The toolkit, named after the *Complex Optimization and Decision Making* group (CODeM) from the University of Sheffield, addresses the above requirements, and can serve as a base for a wider framework for benchmarking algorithms for RMO. An open-source `C++` implementation of the toolkit is available online from https://github.com/UoS-CODeM/CODeM-Toolkit. The various components of the toolkit are described in Section 2, In Section 3, the usage of the toolkit is demonstrated by creating a stochastic benchmark problem. Section 4 concludes the paper with the advantages and limitations of the current work and the outstanding work required to establish a complete framework for benchmarking algorithms for RMO.

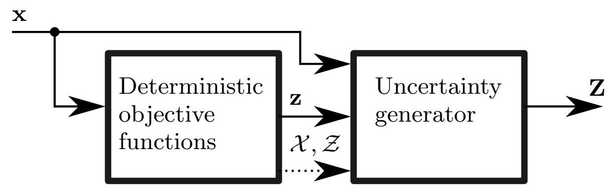


Figure 1: Information flow in a CODeM uncertainty generator. A random objective vector Z is generated according to the deterministic decision and objective vectors x and z, and their relative locations in decision and objective spaces $\mathcal{X}$ and $\mathcal{Z}$, respectively.

2. UNCERTAINTY GENERATOR TOOLKIT

In this section, the CODeM toolkit for generating stochastic multiobjective optimization problems is introduced. A block diagram for a CODeM generated uncertain problem is depicted in Figure 1. It consists of a deterministic problem and an uncertainty generator. Consider a general stochastic multiobjective test problem that maps a vector $\mathbf{x}$ from the decision space $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ to a random vector $\mathbf{Z}$ in objective space $\mathcal{Z} \subset \mathbb{R}^{n_z}$. In our approach, first a decision vector $\mathbf{x}$ is evaluated by the deterministic objective functions to produce a deterministic objective vector $\mathbf{z}$. Next, a random objective vector $\mathbf{Z}$ is generated by the uncertainty generator, according to the relative location of the solution in both decision and objective spaces.

2.1 Geometric Representation of Uncertainty

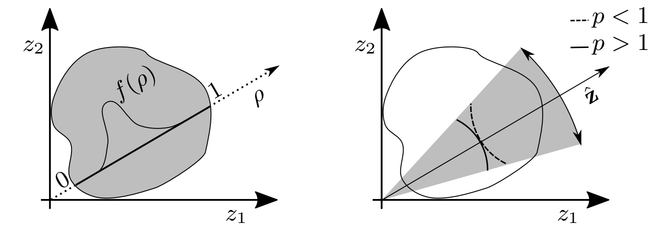
All operators within the toolkit use normalised descriptions of the decision and objective spaces, and the random objective vector is also defined in normalised objective space. Before a candidate solution is being processed by the uncertainty generator, it is first normalised as follows:

$$\tilde{x}_i = \frac{x_i - x_{i,\text{lb}}}{x_{i,\text{ub}} - x_{i,\text{lb}}}, \qquad i = 1, \ldots, n_x, \tag{1}$$

$$\tilde{z}_j = \frac{z_j - z^\star{}_j}{z^{\star\star}{}_j - z^\star{}_j}, \qquad j = 1, \ldots, n_z, \tag{2}$$

where $x_{i,\text{lb}}$ and $x_{i,\text{ub}}$ are the lower and upper bounds, respectively, of the i^{th} decision variable, and $\mathbf{z}^\star$ and $\mathbf{z}^{\star\star}$ are the ideal and anti-ideal vectors, respectively[1]. Whenever a sample $\tilde{\mathbf{z}}'$ is drawn from the normalised random objective

[1]The toolkit assumes minimization of all objectives. If an objective needs to be maximized, its inverse should be used.



(a) The magnitude is described by a probability density function with a finite support between the lower and upper limits of the feasible, normalised objective space.

(b) Once the magnitude ρ is sampled, the vector perturbs on a p-norm sphere with radius ρ within a maximum angle. The norm p determines the curvature of perturbation.

Figure 2: A representation of a random objective vector.

vector, it is scaled back to the original objective space to obtain the sampled objective vector $\mathbf{z}'$:

$$z'_j = z^\star{}_j + \tilde{z}'_j(z^{\star\star}{}_j - z^\star{}_j), \qquad j = 1, \ldots, n_z. \tag{3}$$

Since this paper discusses the internals of the CODeM toolkit, all vectors are assumed to be normalised. For the sake of clarity, the tilde notation is omitted hereafter. Unless otherwise stated, $\mathbf{x}$ and $\mathbf{z}$ refer to normalised decision and objective vectors, respectively.

The CODeM toolkit uses polar representation to describe vectors in objective space. A deterministic objective vector $\mathbf{z}$ is represented by its magnitude and direction:

$$\mathbf{z} = z\hat{\mathbf{z}}, \tag{4}$$

where z is the vector's Euclidean length, and $\hat{\mathbf{z}}$ is its direction vector, defined on the $n_z - 1$ simplex, i.e.,

$$\hat{\mathbf{z}} = \frac{\mathbf{z}}{\sum z_i}, \quad i = 1, \ldots, n_z. \tag{5}$$

A distribution of a random objective vector $\mathbf{Z}$ is also defined in a similar manner. It is represented by a univariate distribution for its magnitude Z and another distribution for its direction vector $\hat{\mathbf{Z}}$. The geometric representation of a random bi-objective vector is demonstrated in Figure 2.

For a given direction $\hat{\mathbf{z}}$, the magnitude is defined in a normalised scale $\rho \in [0, 1]$, where 0 corresponds to the lower bound of the feasible objective space in the $\hat{\mathbf{z}}$ direction, and 1 to the upper bound (see Figure 2(a)). The toolkit enables the user to describe ρ as a random variable, and control its probability density function (PDF) $f(\rho)$. The support of $f(\rho)$ is the interval $0 \leq \rho \leq 1$. In this manner the toolkit makes sure that for every direction, the magnitude of the objective vector is bounded within the feasible objective space.

The direction vector is also represented by a random vector $\hat{\mathbf{Z}}$. It follows a uniform distribution among all direction vectors $\hat{\mathbf{v}}$ that lie within a maximum (Euclidean) distance δ from the deterministic direction vector $\hat{\mathbf{z}}$:

$$\hat{\mathbf{Z}} \sim \mathrm{U}\left(\left\{\hat{\mathbf{v}} \in \mathbb{R}^{n_z} \;\middle|\; \|\hat{\mathbf{v}} - \hat{\mathbf{z}}\|_2 \leq \delta \quad \text{and} \quad \sum_{i=1}^{n_z} \hat{v}_i = 1\right\}\right). \tag{6}$$

To make sure every sample from the uncertain objective vector will reside within the feasible objective domain, the shape of the domain is considered when the uncertain vector is defined. Once a direction vector $\hat{\mathbf{z}}'$ is sampled from the uncertain direction $\hat{\mathbf{Z}}$, it is projected onto a unit hyper-sphere of norm p. The new direction vector, denoted $\hat{\mathbf{z}}'_p$ becomes:

$$\hat{\mathbf{z}}'_p = \frac{\hat{\mathbf{z}}'}{\|\hat{\mathbf{z}}'\|_p}. \tag{7}$$

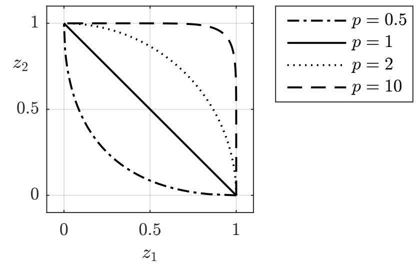


Figure 3: The positive quadrant of a p-norm unit circle for different values of p.

The norm p defines the curvature of perturbation. Figure 3 depicts the first quadrant of a two-dimensional unit circle with different norms. When a norm larger than $p = 1$ is used, the perturbed objective vector follows a concave geometry, and a norm smaller than $p = 1$ follows a convex geometry. The effect of the norm p on the directional perturbation can be seen in Figure 2(b).

2.2 Parametric Description of Uncertainty

An uncertain objective vector generated by the uncertainty generator is composed of three components:

1. Univariate distribution function for the radial component.
2. Maximum perturbation distance for the random direction vector.
3. Curvature norm for perturbations in the perpendicular direction.

These components can either remain constant or change according to the properties of the candidate solution. For example, the perturbation norm can stay constant for problems with simple geometric shapes and vary according to direction for more complicated shapes of the objective space.

2.2.1 Parametric Distributions

The radial component of an uncertain objective vector is defined as a PDF $f(\rho)$ over the interval $\rho \in [0, 1]$. The PDF is composed of n simple parametric distribution functions:

$$f(\rho) = \sum_{i=1}^{n} w_i f_i(\rho), \quad \text{where} \quad \sum_{i=1}^{n} w_i = 1. \tag{8}$$

Three basic distributions are currently available to choose from, all defined in a similar fashion according to their *position* and *locality*. The *position* defines where the distribution function resides on the interval, and the *locality* defines how concentrated the pdf is in this region.

Uniform Distribution. A uniform distribution $\mathrm{U}(lb, ub)$ is defined by two parameters: lower bound $lb \in [0, 1]$ and locality $l \in [0, 1]$. The upper bound ub is a function of the lower bound and locality:

$$ub = 1 - l(1 - lb). \tag{9}$$

The magnitude of the probability function is $1/(ub - lb)$. For $l = 1$, the PDF has infinite value at lb, i.e., lb is the deterministic value of ρ.

Triangular Distribution. A triangular distribution is also defined over the interval $[lb, ub]$. The lower and upper bounds are determined via the parameters lb and l, like for the uniform distribution. The PDF of a triangular distribution either linearly increases or linearly decreases. Increasing triangular distribution is defined:

$$f_{t\uparrow}(\rho) = \begin{cases} \frac{2(\rho - lb)}{(ub-lb)^2}, & \text{for } lb \leq \rho \leq ub, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Decreasing triangular distribution is defined:

$$f_{t\downarrow}(\rho) = \begin{cases} \frac{2(ub - \rho)}{(ub-lb)^2}, & \text{for } lb \leq \rho \leq ub, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Smooth Peak Distribution. The toolkit provides an additional smooth distribution, borrowed from the field of quantum mechanics. This distribution typically describes the location of a particle in an infinitely deep potential well (also known as "particle in a box") [13, pp. 59–65]. The particle has zero probability to exist outside a defined interval, and a certain probability to exist anywhere within the interval. Since it is a smooth function, the probability at the boundaries is zero.

The PDF is composed of k basis functions of the form

$$b_n(\rho) = A \sin(n\pi\rho), \quad n = 1, \dots, k, \quad (12)$$

where A is a normalisation factor. Every basis function is multiplied by a complex coefficient

$$c_n = \sqrt{\frac{N^n \exp(-N)}{n!}} \exp(-i(n + 0.5)\pi r), \quad (13)$$

where r is a *position* parameter for ρ and N is a *locality* parameter. A higher value of N corresponds to a narrower distribution function around $\rho = r$. For consistency with the other distributions in the toolkit, a locality parameter $l \in [0, 1]$ is defined such that

$$N = 9\,l + 1. \quad (14)$$

The smooth peak PDF is defined in the interval $\rho \in [0, 1]$:

$$f_p(\rho) = \left| \sum_{n=1}^{k} c_n b_n(\rho) \right|^2. \quad (15)$$

For a complex number $c = a + ib$, the operator $|c|^2$ is the squared modulus, i.e., $|c|^2 = a^2 + b^2$.

A summary of the parametric distributions available in the toolkit is provided in Table 1. Figure 4 depicts how a PDF is constructed from three basic distributions with different weights.

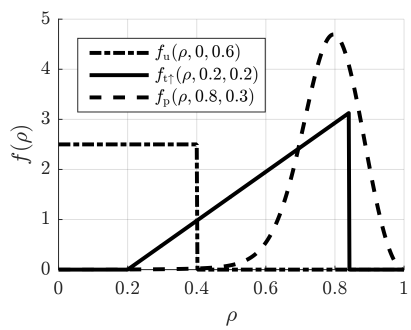
2.3 Direction Perturbation

As described in Section 2.1, the uncertain direction vector $\hat{\mathbf{Z}}$ is controlled by the parameter δ–the radius of perturbation from the deterministic direction vector $\hat{\mathbf{z}}$. An additional parameter that defines the perturbation in the perpendicular directions is the curvature norm p.

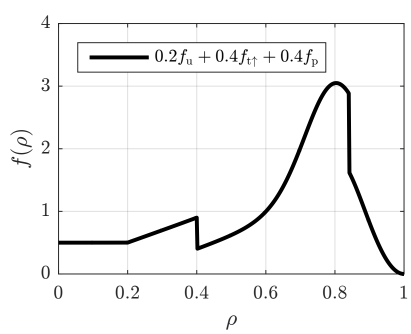
Both parameters can be varied according to the characteristics of the candidate solution as explained in Section 2.4

Distributions	Usage	Parameters	
Smooth Peak	$f_p(r, l)$	r	tendency
Increasing Triangular	$f_{t\uparrow}(lb, l)$	lb	lower bound
Decreasing Triangular	$f_{t\downarrow}(lb, l)$	l	locality
Uniform	$f_u(lb, l)$		

Table 1: Parametric distributions in the toolkit.



(a) PDFs of three parametric distributions.



(b) PDF of the augmented distribution.

Figure 4: A PDF of the objective vector magnitude, combined from three parametric distributions.

2.4 The Uncertainty Kernel

The various parameters defining the uncertain objective vector are based on the deterministic properties of the candidate solution in design and objective spaces. These properties can be seen as the kernel of the uncertain objective vector. The kernel, denoted by the vector $\mathbf{\Psi}$, is composed of $n_x + n_z + 2$ parameters:

$$\begin{aligned}\mathbf{\Psi} = [\Psi_{\mathrm{r}}, \Psi_{\mathrm{s}}, \Psi_{\mathrm{x},1}, \ldots, \Psi_{\mathrm{x},n_x}, \Psi_{\mathrm{z},1}, \ldots, \Psi_{\mathrm{z},n_z}]\,, \\ 0 \le \Psi_i \le 1, \quad i = 1, \ldots, n_x + n_z + 2.\end{aligned} \tag{16}$$

Remoteness parameter Ψ_{r}. Remoteness from the deterministic Pareto front is defined in a similar fashion to ρ in Section 2.1:

$$\Psi_{\mathrm{r}} = \frac{z - z^l}{z^u - z^l}, \tag{17}$$

where $\mathbf{z}^l = z^l\hat{\mathbf{z}}$ corresponds to the Pareto optimal vector with the same direction as the deterministic objective vector $\mathbf{z}$, and $\mathbf{z}^u = z^u\hat{\mathbf{z}}$ corresponds to an upper bound for this direction.

Symmetry parameter Ψ_{s}. Symmetry is considered as the similarity of the objective vector's components. A perfectly symmetric vector is one with identical components, i.e., with the direction vector $\hat{\mathbf{z}} = \left[\frac{1}{n_z}, \ldots, \frac{1}{n_z}\right]$. A perfectly asymmetric vector has one non-zero component, and zeros for the rest. The Euclidean length of the direction vector can take value between $\|\hat{\mathbf{z}}\| = 1/\sqrt{n_z}$, if the vector is perfectly symmetric, and $\|\hat{\mathbf{z}}\| = 1$, if the vector is perfectly asymmetric. The symmetry parameter Ψ_{s} can take values between 0 and 1 based on its direction $\hat{\mathbf{z}}$:

$$\begin{aligned}\Psi_{\mathrm{s}} &= \frac{\|\hat{\mathbf{z}}\|_{\max} - \|\hat{\mathbf{z}}\|}{\|\hat{\mathbf{z}}\|_{\max} - \|\hat{\mathbf{z}}\|_{\min}}\ , \\ \text{where:} \quad \|\hat{\mathbf{z}}\|_{\min} &= \frac{1}{\sqrt{n_z}}\ ,\ \|\hat{\mathbf{z}}\|_{\max} = 1.\end{aligned} \tag{18}$$

Decision vector parameters $\mathbf{\Psi}_{\mathrm{x}}$. For every decision variable x_i, where $i = 1, \ldots, n_x$, the decision parameter $\Psi_{\mathrm{x},i}$ is equal to $\tilde{x}_i$ in Equation 1.

Objective vector parameters $\mathbf{\Psi}_{\mathrm{z}}$. For every objective z_j, where $j = 1, \ldots, n_z$, the objective parameter $\Psi_{\mathrm{z},j}$ is equal to $\tilde{z}_j$ in Equation 2.

2.5 Transformation Functions

The toolkit offers a set of functional operators to manipulate the kernel's parameters when they are used as parameters of the uncertain vector. All functions accept a value $\Psi \in [0, 1]$ and transform it into a value $\Phi(\Psi) \in [0, 1]$. Table 2 contains a list of the available transformation operators and their description. Figure 5 depicts the transformation functions.

2.6 Sampling

Every sample of an uncertain objective vector $\mathbf{Z}$ results in a different objective vector $\mathbf{z}'$. The following procedure describes how a sample is generated by the toolkit: First, a sample ρ' is drawn from the magnitude univariate distribution Z. Next, a sample direction vector $\hat{\mathbf{z}}'$ is drawn from $\hat{\mathbf{Z}}$. Next, the direction vector is projected on a p-norm unit hypersphere to obtain $\hat{\mathbf{z}}'_p$. Next, the normalised sample $\tilde{\mathbf{z}}'$ is generated by multiplying the sampled magnitude and direction: $\tilde{\mathbf{z}}' = \rho'\hat{\mathbf{z}}'_p$. Finally, $\mathbf{z}'$ is obtained by rescaling $\tilde{\mathbf{z}}'$ to the original objective space using Equation (3).

The sampling computational cost is considerately smaller than the cost of calculating the deterministic objective vector $\mathbf{z}$ and the uncertainty kernel $\mathbf{\Psi}$. Once $\mathbf{Z}$ and $\mathbf{\Psi}$ are defined, there is no need to recalculate them for every sample. In contrast to other uncertain black-box evaluation functions that require a new evaluation of the evaluation functions for every sample (e.g., [8]), the computational complexity of large samples from a CODeM generated vector is moderate.

3. USING THE CODEM TOOLKIT

The following procedure describes how to use the CODeM toolkit to generate a stochastic multiobjective optimization problem:

1. Define a deterministic multiobjective problem
2. Define a radial probability function:
   (a) Choose parametric distributions
   (b) For each distribution, define parameters as functions of kernel parameters
   (c) Assign a weight for each distribution
3. Define the directional distribution:
   (a) Define perturbation radius as a function of kernel parameters
   (b) Define perturbation curvature norm as a function of kernel parameters

3.1 An Example Stochastic Test Problem

The following example problem is provided to demonstrate to use the toolkit.

Deterministic Problem. A variant[2] of DTLZ1 [4] is used. It is a scalable, separable, multimodal problem, with $n_z - 1$ variables responsible for the direction of the objective vector, and the remainder for its magnitude. The feasible objective space is bounded between two parallel hyperplanes $\sum_{j=1}^{n_z} z_j = 0.5$ and $\sum_{j=1}^{n_z} z_j = 1.125(n_x - n_z + 1) + 0.5$.

Radial Probability Function. A uniform distribution with the deterministic value as the lower bound. The locality of the distribution is inversely correlated to the optimality of the deterministic vector. Solutions that are very far from the deterministic front (i.e., $\Psi_{\mathrm{r}} \ge 0.5$) have no radial uncertainty at all (i.e., maximum locality). As the solutions get closer to the deterministic front, the locality decreases, and the degree of radial uncertainty increases:

$$f(\rho) = f_{\mathrm{u}}\Big(\rho, \Psi_{\mathrm{r}}, 0.9 + 0.1\Phi_{\mathrm{Z}}(\Psi_{\mathrm{r}}, 0.0, 1.0)\Big). \tag{19}$$

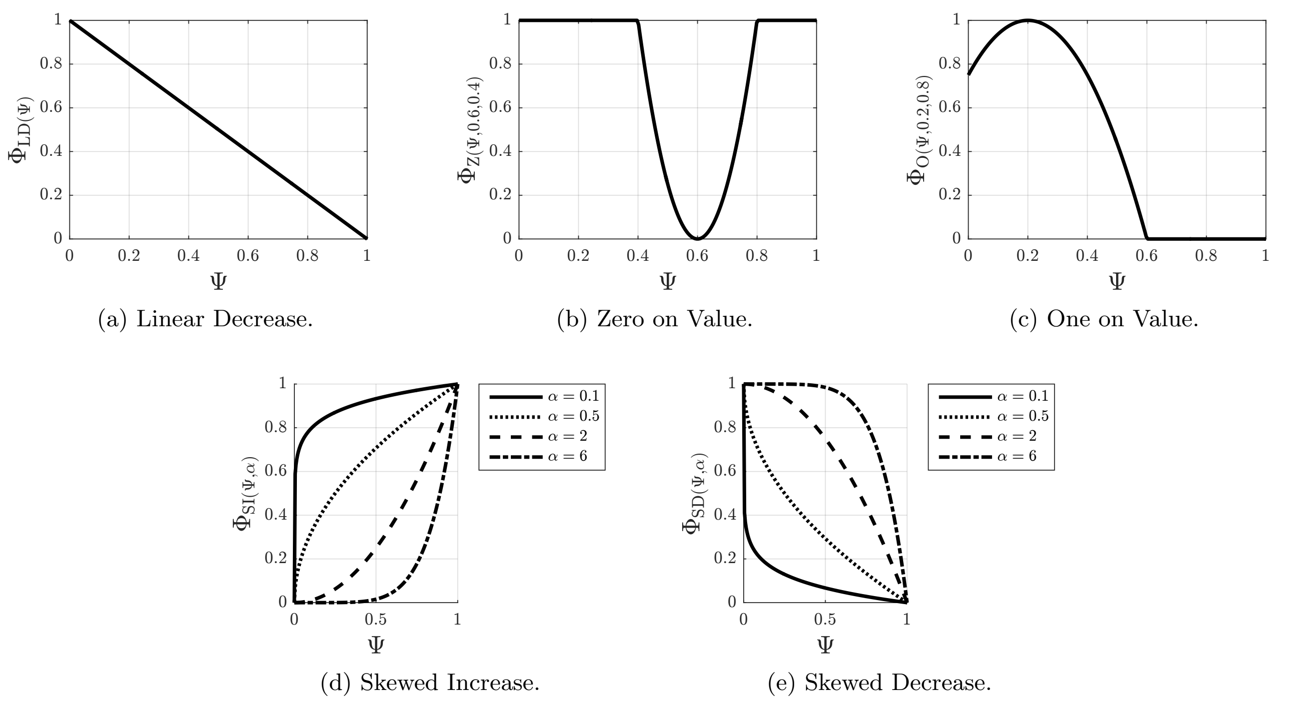
Directional Distribution. The Pareto front of the deterministic problem is affine. Therefore, the perturbation norm $p = 1$ is used. The perturbation radius δ is very small for high symmetry, and higher at the boundaries of the objective space:

$$\delta = 0.02 + 0.1\Phi_{\mathrm{LD}}(\Psi_{\mathrm{s}}). \tag{20}$$

[2] The 100 scaling factor is removed from the distance function $g(\mathbf{x}_M)$ in [4].

Function	Description	Parameters		Definition
Linear Decrease	Function values decrease at a constant rate.			$\Phi_{\mathrm{LD}}(\Psi) = 1 - \Psi$
Skewed Increase	Function values increase at a changing rate. Sensitive to changes of high Ψ values when $0 < \alpha < 1$ and low values when $\alpha > 1$.	$\alpha > 0$	Skewness	$\Phi_{\mathrm{SI}}(\Psi, \alpha) = \Psi^{\alpha}$
Skewed Decrease	Function values decrease at a changing rate. Sensitivity as for Φ_{SI}.	$\alpha > 0$	Skewness	$\Phi_{\mathrm{SD}}(\Psi, \alpha) = 1 - \Psi^{\alpha}$
Zero on Value	Parapolic function of width β with a minimum $(\alpha, 0)$. Function values equal to 1 outside $\alpha \pm \beta/2$.	$\alpha \in [0,1]$ $\beta > 0$	Zero value Width	$\Phi_{\mathrm{Z}}(\Psi, \alpha, \beta) = \min\left(\frac{4(\Psi-\alpha)^2}{\beta^2}\ ,\ 1\right)$
One on Value	Parapolic function of width β with a maximum $(\alpha, 1)$. Function values equal to 0 outside $\alpha \pm \beta/2$.	$\alpha \in [0,1]$ $\beta > 0$	One value Width	$\Phi_{\mathrm{O}}(\Psi, \alpha, \beta) = \max\left(1 - \frac{4(\Psi-\alpha)^2}{\beta^2}\ ,\ 0\right)$

Table 2: Transformation functions.



(a) Linear Decrease.

(b) Zero on Value.

(c) One on Value.

(d) Skewed Increase.

(e) Skewed Decrease.

Figure 5: Transformation functions.

This increases the difficulty in finding the boundary solutions.

The problem is scalable in both decision and objective spaces. Its variants with two and three objectives are depicted in Figure 6. Ten randomly generated solutions were evaluated by the stochastic problem, together with a smaller set of evenly spaced Pareto optimal solutions to the deterministic problem. For every solution, the deterministic objective vector $\mathbf{z}$ is marked with a grey circle, and 50 samples from the random objective vector $\mathbf{Z}$ are marked with black dots.

4. CONCLUSIONS

This study has articulated the need for a framework for benchmarking robust multiobjective optimization algorithms and identified the requirements for test problems that would form a key component of the framework. Based on these requirements, the CODeM toolkit has been proposed. The toolkit possesses the following advantages:

1. With the use of a small number of operators, various challenges can be easily included.
2. All operators are parametric, which allows the user to create meaningful benchmarks with arbitrary difficulties.
3. Readily available deterministic benchmark problems can be used, without the need to manipulate their internal structure in a bespoke manner.
4. Interesting combinations of challenges introduced by the deterministic function and the uncertainty generator can be created.
5. The toolkit ensures all uncertain objective vectors are defined within the feasible deterministic objective space. This means the deterministic optimal set can serve as a reference for the optimal robust set, and for the approximation set obtained by the algorithm.
6. Difficulty due to uncertainty is related to the main challenges of multiobjective optimization, i.e., spread and convergence. Increased uncertainty in the radial direction hinders convergence while increased uncertainty in the perpendicular directions makes it more difficult to find a set with a good spread across the robust front.

Nevertheless, the toolkit also has some limitations due to the use of a radial coordinate system to describe random objective vectors:

1. The feasible objective space must be known to ensure the random objective vectors are bounded by the deterministic objective space. To describe the objective space in radial coordinates, a functional relation must be identified between the deterministic direction vector and the boundaries of the objective space in this direction. It might be difficult to describe objective spaces with complicated geometries such as disconnected spaces or the presence of 'holes' within the objective space. These examples require a description of infeasible intervals between the lower and upper boundaries. However, once the infeasible intervals are identified, it becomes straightforward to limit the random objective vectors to the feasible space, by setting the PDF over the infeasible intervals to zero.
2. It is not possible to define uncertainty on a per-objective basis and, as a consequence, it is also not possible to define an objective to possess no uncertainty. These issues can be resolved by excluding objectives of this type from the generator and defining their uncertainty, where it exists, separately.

4.1 Future Work

This study is a crucial first step towards the establishment of a benchmarking framework for RMO. Other aspects of the framework that still require consideration include robustness indicator definitions, development of performance indicators, and a structured procedure for comparing algorithms according to useful hypotheses.

Once a prototype framework is established, the following studies can be made as a direct continuation to this study:

1. Generation of a test suite using the toolkit, addressing various challenges for RMO.
2. Identification of robust Pareto fronts according to different robustness indicators.
3. Comparison of algorithms across instances of problems taken from the test suite. Of particular interest will be the extent to which algorithms explicitly designed for RMO are able to outperform existing, state-of-the-art, deterministic optimizers.

4.2 Acknowledgments

This work was supported by Jaguar Land Rover and the UK-EPSRC grant EP/L025760/1 as part of the jointly funded Programme for Simulation Innovation.

5. REFERENCES

[1] A. Auger, D. Brockhoff, N. Hansen, D. Tusar, T. Tusar, and T. Wagner. GECCO workshop on real-parameter black-box optimization benchmarking (BBOB 2016) - focus on multi-objective problems, 2016.

[2] H. G. Beyer and B. Sendhoff. Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, 2007.

[3] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4):463–494, 2006.

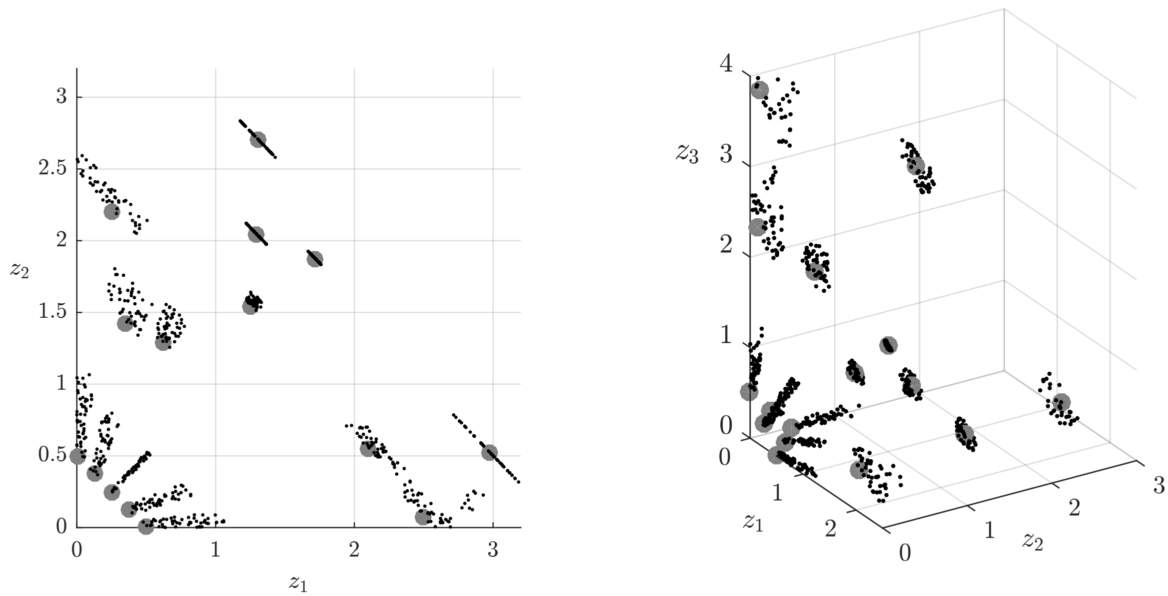
[4] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems, 2002.

[5] J. E. Fieldsend and R. M. Everson. The rolling tide evolutionary algorithm: A multi-objective optimiser for noisy optimisation problems. *Evolutionary Computation, IEEE Transactions on*, 19(1):103–117, 2015.

[6] C. M. Fonseca, J. D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. In *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 216, page 240, 2005.

[7] A. Gaspar-Cunha, J. Ferreira, and G. Recio. Evolutionary robustness analysis for multi-objective optimization: benchmark problems. *Structural and Multidisciplinary Optimization*, 49(5):771–793, 2013.

[8] C. Goh, K. Tan, C. Cheong, and Y. Ong. An investigation on noise-induced features in robust



(a) Two objectives.

(b) Three objectives.

Figure 6: Distribution of Pareto optimal and random solutions for an example stochastic problem.

evolutionary multi-objective optimization. *Expert Systems with Applications*, 37(8):5960–5980, 2010.

[9] C. K. Goh and K. C. Tan. An investigation on noisy environments in evolutionary multiobjective optimization. *Evolutionary Computation, IEEE Transactions on*, 11(3):354–381, 2007.

[10] V. Huang, A. Qin, K. Deb, and E. Zitzler. Problem definitions for performance assessment of multi-objective optimization algorithms. Technical report, Nanyang Technological University, Singapore, 2007.

[11] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *Evolutionary Computation, IEEE Transactions on*, 10(5):477–506, 2006.

[12] J. Knowles, D. Corne, and A. Reynolds. Noisy multiobjective optimization on a budget of 250 evaluations. In *Evolutionary Multi-Criterion Optimization SE - 8*, volume 5467 of *Lecture Notes in Computer Science*, pages 36–50. Springer Berlin Heidelberg, 2009.

[13] D. A. B. Miller. *Quantum Mechanics for Scientists and Engineers*. Cambridge University Press, Cambridge, United Kingdom, 2008.

[14] S. Mirjalili. Shifted robust multi-objective test problems. *Structural and Multidisciplinary Optimization*, 52(1):217–226, 2015.

[15] S. Mirjalili and A. Lewis. Hindrances for robust multi-objective test problems. *Applied Soft Computing*, 35:333–348, 2015.

[16] S. Mirjalili and A. Lewis. Novel frameworks for creating robust multi-objective benchmark problems. *Information Sciences*, 300:158–192, 2015.

[17] T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff. On test functions for evolutionary multi-objective optimization. In *Parallel Problem Solving from Nature - PPSN VIII SE - 80*, volume 3242 of *Lecture Notes in Computer Science*, pages 792–802. Springer Berlin Heidelberg, 2004.

[18] A. Syberfeldt, A. Ng, R. I. John, and P. Moore. Evolutionary optimisation of noisy multi-objective problems using confidence-based dynamic resampling. *European Journal of Operational Research*, 204(3):533–544, 2010.

[19] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.

[20] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert. Performance assessment of multiobjective optimizers : An analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132, 2003.