

Generalized Information Reuse for Optimization Under Uncertainty with Non-Sample Average Estimators

Laurence W Cook, Jerome P Jarrett, Karen E Willcox

June 22, 2018

Abstract

In optimization under uncertainty for engineering design, the behavior of the system outputs due to uncertain inputs needs to be quantified at each optimization iteration, but this can be computationally expensive. Multi-fidelity techniques can significantly reduce the computational cost of Monte Carlo sampling methods for quantifying the effect of uncertain inputs, but existing multi-fidelity techniques in this context apply only to Monte Carlo estimators that can be expressed as a sample average, such as estimators of statistical moments. Information reuse is a particular multi-fidelity method that treats previous optimization iterations as lower-fidelity models. This work generalizes information reuse to be applicable to quantities with non-sample average estimators. The extension makes use of bootstrapping to estimate the error of estimators and the covariance between estimators at different fidelities. Specifically, the horsetail matching metric and quantile function are considered as quantities whose estimators are not sample-averages. In an optimization under uncertainty for an acoustic horn design problem, generalized information reuse demonstrated computational savings of over 60% compared to regular Monte Carlo sampling.

Nomenclature

\mathbf{x}	Vector of design variables
n_x	The number of design variables
ω	Underlying random outcome
$\mathbf{U}(\omega)$	Random vector of uncertain input parameters
\mathbf{u}	Realization of input parameters
n_u	The number of uncertain input parameters
n_c	The number of constraints
y	General system model output
$Y(\omega)$	Random variable representing a system output
n	Number of sampled values
Y^i	The i^{th} sampled value of $Y(\omega)$

$y^{(s)}$	The s^{th} order statistic out of n sampled values of $Y(\omega)$
d	Quantity used in an optimization formulation
\hat{d}	Naive Monte Carlo estimator of d from sampled values of $Y(\omega)$
\hat{s}	Estimator of d used in the optimization
N_{boot}	Number of times re-sampling occurs in bootstrapping
\hat{d}^b	The value of \hat{d} evaluated using the b^{th} set of re-sampled values of $Y(\omega)$
\bar{V}	Bootstrapped estimate of the variance of an estimator
$\bar{\Sigma}$	Bootstrapped estimate of covariance between two estimators
$F_Y(y)$	Cumulative distribution function of $Y(\omega)$
x	As a subscript/superscript: of a given design point
A	As a subscript/superscript: of the current design point in an optimization
C	As a subscript/superscript: of the design point selected as the control point
β	Minimum acceptable probability of constraint satisfaction

1 Introduction

Optimization techniques are becoming increasingly integrated within the engineering design process when computational models of the system are available. Often, various inputs to models of the system are uncertain in reality [1], and it is important to account during the design process for their influence on the performance of the system. The field of optimization under uncertainty (OUU) has therefore developed various optimization formulations for engineering design that can improve the behavior of a system subject to uncertain inputs [2–6]. These formulations involve defining quantities that describe the behavior under uncertainty of the system, which are then used as the objective or as constraints in an optimization.

For practical engineering systems, the system model can be computationally expensive to evaluate, and so optimizing such systems represents a significant computational burden. Surrogate-based and multi-fidelity formulations have been developed to address this issue [7–14]. Multi-fidelity approaches leverage cheap surrogate models to reduce the computational cost, while retaining occasional recourse to the high-fidelity model to ensure accuracy.

In OUU, the effects of uncertain inputs on the system must be propagated at every optimization iteration. Various methods have been developed to achieve this propagation at low computational expense, including (but not limited to) quadrature based integration [15], stochastic expansions [9, 15–17], and surrogate models [18, 19]. Whilst effective in many scenarios, especially when the problem is smooth, these methods suffer from the “curse of dimensionality” whereby their cost increases exponentially with the number of uncertain inputs.

Monte Carlo (MC) sampling, on the other hand, is a method whose convergence rate is independent of both the dimensionality and smoothness of the problem [20, 21], making it attractive for problems with a large number of uncertainties or with particularly non-smooth behavior. This convergence rate is slow, however, and some problems may simply be too computationally expensive for MC sampling; for example if only a few hundred expensive high-fidelity model evaluations

can be afforded within an optimization, then MC sampling is not recommended since a designer would have little confidence in the accuracy of the results.

Multi-fidelity approaches can offer a way of reducing the computational expense of MC sampling [22], allowing it to be useful when the cost would otherwise be marginally too high, or allowing it to be better integrated within an engineer’s workflow for practical problems. In the context of OUU with MC sampling, a multi-fidelity formulation based on control variates has been proposed to reduce the number of system model evaluations required to propagate the effect of the uncertainties [23]. The formulation in that work applies to general low-fidelity models, but of particular benefit in the OUU setting is the use of information from previous optimization iterations, termed “information reuse” in Ref. 23. However, this control-variate-based approach, as presented in Ref. 23, is limited to formulations of the OUU problem where estimators of the quantities being optimized or constrained must be expressed as a sample average (e.g., estimators of the first two statistical moments, mean and variance).

There exists a rich variety of quantities that can be optimized or constrained in OUU for engineering design [3, 4, 24–27]. Estimators of many of these quantities cannot be represented as a sample average, and so existing multi-fidelity approaches for OUU cannot be used to accelerate such estimators. This paper extends the information reuse approach of Ref. 23, to estimators that cannot be expressed as a sample average. Consequently information reuse becomes applicable to a broader range of OUU formulations. Our “generalized information reuse” approach makes use of the ideas behind bootstrapping [28] to evaluate both the error of estimators and the covariance between estimators using different fidelity models.

Section 2 outlines the general formulation of an OUU for engineering design and gives examples of quantities optimized/constrained in an OUU. Section 3 presents the generalized information reuse approach. Section 4 tests the approach on an algebraic test problem, and applies it to the design of an acoustic horn under uncertainty. Section 5 concludes the paper.

2 Formulations of Optimization Under Uncertainty

This section discusses a general formulation of the OUU problem and the use of Monte Carlo simulation in solving it. We then present several specific quantities that can be used as objective or constraint functions in OUU, including statistical moments, the probability of failure, the quantile/value at risk, and the horsetail matching metric.

2.1 General Formulation of the Optimization Problem

Models of practical engineering systems often produce various outputs of interest. These outputs enter into the design problem through the objective or constraints. We denote a general output by $y(\mathbf{x}, \mathbf{u})$, which is a function of controllable design variables, \mathbf{x} , and input parameters, \mathbf{u} . We treat \mathbf{u} as uncertain, and use a probabilistic representation such that \mathbf{u} is a realization of a random vector $\mathbf{U}(\omega)$, where ω denotes the underlying random event with sample space Ω such that $\omega \in \Omega$. Thus at a given design \mathbf{x} , an output is a random variable $Y_x(\omega)$, defined such that $Y_x(\omega) = y(\mathbf{x}, \mathbf{U}(\omega))$.

An OUU requires an objective, denoted $d_0(\mathbf{x})$, and constraints, denoted $d_j(\mathbf{x})$, to be defined that characterize the behavior of Y_x as a function of the design \mathbf{x} . For example the expected value of Y_x could be minimized subject to a constraint on the variance of Y_x . In general, the formulation

of a (single objective) OUU problem can be written as:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && d_0(\mathbf{x}) \\ & \text{s.t.} && d_j(\mathbf{x}) \leq 0 \quad j = 1, \dots, n_c \end{aligned} \tag{1}$$

where each of $d_j(\mathbf{x}), j = 0, 1, \dots, n_c$ is an appropriately defined function. Examples of functions to use in this OUU formulation are given in Sections 2.3 to 2.6. In the following, we drop the subscript notation to refer to a general $d(\mathbf{x})$ that could be any of $d_j(\mathbf{x}), j = 0, 1, \dots, n_c$.

2.2 Monte Carlo Sampling

To implement an OUU, an estimate of $d(\mathbf{x})$ must be obtained at each design point \mathbf{x} visited by the optimization algorithm. Such an estimate is usually obtained by evaluating the output $y(\mathbf{x}, \mathbf{u})$ at particular values of the input parameters. In many practical design problems, evaluating $y(\mathbf{x}, \mathbf{u})$ involves using black box simulations that model the system being optimized, and so a non-intrusive approach is required.

Monte Carlo (MC) simulation is a non-intrusive approach which uses an estimator, denoted by \hat{d} , to obtain an estimate of $d(\mathbf{x})$ [20, 21]. Solving the OUU problem using MC sampling gives a nested formulation, where in the outer loop an optimizer determines the sequence of design points to visit, and in the inner loop MC sampling obtains estimates of $d_j(\mathbf{x}), j = 0, 1, \dots, n_c$. Many MC estimators are *sample averages* in that they are a function that can be expressed in the form:

$$\hat{d} = \frac{1}{n} \sum_{i=1}^n Z^i, \tag{2}$$

where $Z^i, i = 1, 2, \dots, n$ are n independent and identically distributed (i.i.d.) random variables that depend on Y_x . A sample average estimator is an unbiased estimator of the mean of Z^i , and an analytical form of the estimator variance can be obtained as follows:

$$\text{Var}[\hat{d}] = \frac{1}{n^2} \text{Var} \left[\sum_{i=1}^n Z^i \right] = \frac{\sigma_Z^2}{n} \tag{3}$$

where σ_Z^2 is the variance of Z^i . Therefore the standard deviation, and hence root mean squared error, of a sample average estimator is proportional to $n^{-1/2}$ [20, 21]. However, in general, MC estimators are of the form:

$$\hat{d} = f(Z^1, Z^2, \dots, Z^n) \tag{4}$$

where f is a function that cannot necessarily be expressed as a sample average, in which case we cannot use Equation (3) to derive their variance.

Sections 2.3 to 2.6 discuss several key choices of $d(\mathbf{x})$ that can be used in OUU for engineering design, and give their Monte Carlo estimators. Whether they are used as objectives or constraints in the OUU formulation is left up to a designer setting up the problem.

2.3 Statistical Moments

A common quantity used as the objective function in OUU is the mean value, $\mu = E[Y_x]$. The variance, $\sigma^2 = \text{Var}[Y_x]$, and standard deviation, σ , are also often used. The mean and variance (or standard deviation) are used as the objective function(s) in traditional robust design optimization,

where they can be optimized separately in a multi-objective formulation [3, 29–33], but are often combined into a single objective using a weighted sum [3, 17, 29, 34, 35].

Constraints are also often formulated using statistical moments, for example requiring that the mean value is λ standard deviations away from failure. In this case, the constraint in the OOU formulation is $d = \mu + \lambda\sigma \leq 0$ [36–38]. In the case where Y_x is distributed normally, this gives a constraint on the probability of failure. Note that this is the approach applied to the constraints in the original information reuse formulation [23, 39].

The estimator of the mean of Y_x is given by:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n Y_x^i, \quad (5)$$

where Y_x^i are n i.i.d. random variables with the distribution of Y_x . Thus we can set $Z^i = Y_x^i$ to express this estimator in the form of Equation (2). The estimator of the variance of Y_x is given by:

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n \frac{n}{n-1} (Y_x^i - \hat{\mu})^2, \quad (6)$$

and we can set $Z^i = \frac{n}{n-1} (Y_x^i - \hat{\mu})^2$ to express this estimator in the form of Equation (2). Since both of these estimators are sample averages (they can be expressed in the form of Equation (2)) we can use Equation (3) to estimate their variance.

2.4 Probability of Failure

Another common formulation of OOU uses the probability of failure:

$$p = \text{Prob}(Y_x \geq 0), \quad (7)$$

where here failure is indicated by $Y_x > 0$. This failure probability is usually used as a constraint in OOU formulations, such that $p - (1 - \beta) \leq 0$, where β is the minimum acceptable probability of success, so $(1 - \beta)$ is the maximum acceptable probability of failure. The sample average estimator of p is given by using $Z^i = \mathbb{1}_{[0, \infty)}(Y_x^i)$ in Equation (2):

$$\hat{p} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{[0, \infty)}(Y_x^i), \quad (8)$$

where $\mathbb{1}_S(\cdot)$ is the indicator function that is equal to 1 when its argument is in the set S and 0 otherwise.

2.5 Quantile/ Value at Risk

An alternative to constraining the probability of failure is to constrain the quantile function (the inverse of the CDF):

$$q_\beta = F_{Y_x}^{-1}(\beta) \leq 0, \quad (9)$$

where $F_{Y_x}^{-1}(\beta)$ is the quantile function of Y_x , evaluated at β . The quantile has previously seen much use in OOU formulations for operational research and financial applications [40–42] (where it is often known as the Value at Risk), and recently it is being considered more in engineering design applications [26, 27]. The following estimator of the quantile function $F_{Y_x}^{-1}(h)$ is used [43]:

$$\hat{q}_\beta = \begin{cases} Y_x^{(n\beta)}, & \text{if } n\beta \text{ is an integer.} \\ Y_x^{(\lfloor n\beta \rfloor + 1)}, & \text{otherwise.} \end{cases} \quad (10)$$

where $\lfloor \cdot \rfloor$ indicates rounding down to the nearest integer, and $Y_x^{(k)}$ is the k^{th} order statistic of a set of n realizations of Y_x (which is defined as the k^{th} smallest out of the realized values).

This estimator cannot be expressed as a sample average and so an analytical form of the variance cannot be derived from Equation (3).

2.6 Horsetail Matching Metric

Horsetail matching is a formulation of OUU that minimizes the difference between the inverse CDF of Y_x and a target [44], where this difference is given by:

$$d_{hm} = \left(\int_0^1 (F_{Y_x}^{-1}(h) - t(h))^2 dh \right)^{1/2}, \quad (11)$$

where $F_{Y_x}^{-1}(h)$ is the quantile function of Y_x , and $t(h)$ is the target. Note that this target does not necessarily have to consist of the inverse of a valid CDF, it can be any function of h [44].

In [44], the horsetail matching metric is evaluated in differentiable form using kernels, but here since only the value of d_{hm} (not the gradient) is required, the metric is evaluated by directly integrating the empirical CDF. This is given by the following estimator:

$$\hat{d}_{hm} = \left(\frac{1}{n} \sum_{k=1}^n (Y_x^{(k)} - t(k/n))^2 \right)^{1/2}, \quad (12)$$

where $Y_x^{(k)}$ is the k^{th} order statistic of a set of n realizations of Y_x . Again, this is not a sample average so an analytical expression for the error of this estimator cannot be derived from Equation (3).

3 Generalized Information Reuse

At the current design point selected by the optimizer outer loop, denoted \mathbf{x}_A , an estimator of $d(\mathbf{x}_A)$ must be evaluated; d could be (but is not limited to) any of the quantities discussed in Section 2. To evaluate an estimator using regular MC, we draw n i.i.d. sampled values $\mathbf{u}^i, i = 1, 2, \dots, n$ from the distribution of $\mathbf{U}(\omega)$, then evaluate the system model at each of these values to obtain n i.i.d. sampled values of $Y_A := y(\mathbf{x}_A, \mathbf{U}(\omega))$, denoted $y_A^i = y(\mathbf{x}_A, \mathbf{u}^i), i = 1, 2, \dots, n$. Each of these values $y_A^i, i = 1, 2, \dots, n$ is then used as a realization of each of the random variables Y_x^i in the estimator definition (such as those given in Section 2), to evaluate the estimator and obtain an estimate of d .

If the MSE (mean squared error) of an estimator is too high such that the optimizer is not given accurate information about the objective and constraints, the optimizer will not be able to effectively select new design points to improve the design and meet the constraints. Therefore a tolerance is set on the MSE of the estimator evaluated by the MC inner loop, which must be met at each design point. The estimator that meets this acceptable error at the current design point is denoted by \hat{s}_A . A regular MC estimator may require a large value of n , and hence require many expensive system model evaluations, in order to achieve this acceptable error. In this section we present and analyze the generalized information reuse estimator that can reduce this computational cost.

3.1 The Information Reuse Estimator

Information reuse (IR) is a variance reduction method for obtaining \hat{s}_A which is based on the control variate approach [20, 45]. The control variate approach uses an auxiliary random variable to make

a correction to a regular MC estimator. For information reuse, this auxiliary random variable is the system output at the closest point in design space (in terms of smallest Euclidian distance) previously visited during an optimization. We denote this closest point as the control point \mathbf{x}_C , and let $Y_C(\omega) = y(\mathbf{x}_C, \mathbf{U}(\omega))$ be the auxiliary random variable.

Given a set of i.i.d. sampled values $\mathbf{u}^i, i = 1, \dots, n$ that are drawn from the distribution of $\mathbf{U}(\omega)$, we evaluate the system model using these sampled values at the current design point and the control point to obtain $y_A^i = y(\mathbf{x}_A, \mathbf{u}^i)$ and $y_C^i = y(\mathbf{x}_C, \mathbf{u}^i), i = 1, \dots, n$ respectively. Then we use y_A^i and $y_C^i, i = 1, \dots, n$, to evaluate \hat{d}_A and \hat{d}_C which are regular MC estimators of $d(\mathbf{x}_A)$ and $d(\mathbf{x}_C)$ respectively. The classical control variate estimator is then given by:

$$\hat{s}_A = \hat{d}_A + \gamma(d_C - \hat{d}_C). \quad (13)$$

where d_C is the true value of $d(\mathbf{x}_C)$, and γ is the correction factor.

In practice, the true value of $d(\mathbf{x}_C)$ is not known and the estimator with acceptable error at the control point, \hat{s}_C , is used instead. The information reuse (IR) estimator is thus given by:

$$\hat{s}_A = \hat{d}_A + \gamma(\hat{s}_C - \hat{d}_C). \quad (14)$$

The sampled values $\mathbf{u}^i, i = 1, \dots, n$ used to evaluate \hat{d}_A and \hat{d}_C are drawn independently for each new design point visited. This ensures \hat{s}_C is uncorrelated with either \hat{d}_A or \hat{d}_C , and thus the variance of the IR estimator is given by:

$$V[\hat{s}_A] = V[\hat{d}_A] + \gamma^2(V[\hat{s}_C] + V[\hat{d}_C]) - 2\gamma\Sigma[\hat{d}_A, \hat{d}_C]. \quad (15)$$

where $V[\cdot]$ denotes variance and $\Sigma[\cdot, \cdot]$ denotes covariance. The optimal choice of γ minimizes the variance of the IR estimator:

$$\gamma^* = \frac{\Sigma[\hat{d}_A, \hat{d}_C]}{V[\hat{s}_C] + V[\hat{d}_C]}, \quad (16)$$

and using this value of γ gives the following variance of the IR estimator:

$$V[\hat{s}_A] = V[\hat{d}_A] - \frac{\Sigma[\hat{d}_A, \hat{d}_C]^2}{(V[\hat{s}_C] + V[\hat{d}_C])}, \quad (17)$$

which is derived from Equation (15) and so $V[\hat{s}_A] \geq 0$:

If the covariance between the MC estimators at the current design point and the control design point is high, then the variance of the estimator \hat{s}_A can be significantly reduced compared to the regular MC estimator \hat{d}_A . Within an optimization, it is the norm that the design points visited at consecutive iterations get closer together as the optimization progresses, therefore the covariance between MC estimators at these design points is expected to increase the closer together they are; it is this phenomenon of which information reuse takes advantage. Indeed, in Ref. [23], it was shown that in 1D the correlation between $y(x, \mathbf{U}(\omega))$ and $y(x + \Delta x, \mathbf{U}(\omega))$, is quadratic in Δx for $|\Delta x| \ll 1$, and approaches 1 as Δx approaches 0.

To implement information reuse, estimates for the variance and covariance terms in Equation (17) must be obtained. It was illustrated in Ref. [23] how the variance reduction effect of information reuse is robust with respect to inaccuracies in estimates of these variance and covariance terms. Therefore since we do not have access to exact values of these terms, we can use estimates obtained from the values of y_A^i and $y_C^i, i = 1, 2, \dots, n$ and still obtain significant reduction in the variance of \hat{s}_A . Such estimates in the original formulation of information reuse [23, 39] rely on analytical expressions for the variance and covariance of sample average MC estimators

similar to Equation (3). Therefore, of the quantities from Section 2, this original formulation is only valid for $\hat{\mu}$, $\hat{\sigma}^2$, and \hat{p} .

Our generalization of information reuse can be applied to a general d , whose MC estimator cannot necessarily be expressed as a sample average. Examples of such estimators include \hat{d}_{hm} and \hat{q}_β . Our approach uses the idea of re-sampling from an existing set of sampled values, which is the backbone of bootstrapping [28, 46]. Bootstrapping is widely used to estimate standard errors of estimators for which analytical expressions based on sampled values are not available. The formulation presented here uses this idea to estimate not only the error of an estimator, but also the covariance between estimators at two separate design points.

3.2 Bootstrapped Estimates of Variance and Covariance

Given a set of n i.i.d. sampled values $\mathbf{u}^i, i = 1, \dots, n$, from the distribution of $\mathbf{U}(\omega)$, consider the empirical distribution consisting of a probability density function (PDF) made up of delta functions at each sampled value \mathbf{u}^i . This empirical distribution approximates the true distribution of $\mathbf{U}(\omega)$. Thus, re-sampling with replacement from this empirical distribution, by randomly drawing an index from $i = 1, \dots, n$ and using \mathbf{u}^i as the resulting sample realization, obtains sampled values that are from an approximation to the distribution of $\mathbf{U}(\omega)$. The idea behind bootstrapping is to use this re-sampling to obtain a new set of sampled values of $Y_A(\omega)$ and $Y_C(\omega)$ from which an estimator is evaluated, giving a realization of the estimator from an approximation to the estimator's distribution [28, 46].

In order to use bootstrapping to estimate the terms in Equation (17), firstly a set of n i.i.d. sampled values $\mathbf{u}^i, i = 1, \dots, n$, are drawn from the distribution of $\mathbf{U}(\omega)$, and the system model at both \mathbf{x}_A and \mathbf{x}_C is evaluated to obtain n sampled values of both $Y_A(\omega)$ and $Y_C(\omega)$, denoted $y_A^i = y(\mathbf{x}_A, \mathbf{u}^i)$ and $y_C^i = y(\mathbf{x}_C, \mathbf{u}^i)$ respectively for $i = 1, \dots, n$.

Next, re-sampling with replacement occurs n times from $\mathbf{u}^i, i = 1, \dots, n$ to obtain a new set of values $\mathbf{u}^j, j = 1, \dots, n$, and the corresponding values of $y(\mathbf{x}_A, \mathbf{u}^j)$ and $y(\mathbf{x}_C, \mathbf{u}^j)$ (which have already been evaluated) are used as new sets of sampled values of $Y_A(\omega)$ and $Y_C(\omega)$ respectively. This re-sampling is repeated N_{boot} times, and using each set of the re-sampled values $y(\mathbf{x}_A, \mathbf{u}^j)$ and $y(\mathbf{x}_C, \mathbf{u}^j), j = 1, 2, \dots, n$, a new MC estimator is evaluated to obtain \hat{d}_A^b and $\hat{d}_C^b, b = 1, \dots, N_{boot}$. These are sampled values from approximations to the true distributions of \hat{d}_A and \hat{d}_C .

The bootstrapped estimate of the variance of an MC estimator \hat{d} , denoted $\bar{V}(\hat{d})$, is obtained from the sample variance of these N_{boot} values $\hat{d}^b, b = 1, \dots, N_{boot}$. Similarly, since the re-sampled values of Y_A and Y_C are the system model evaluated at the same values of \mathbf{u} , the bootstrapped estimate of covariance between the estimators at design points \mathbf{x}_A and \mathbf{x}_C , denoted $\bar{\Sigma}$, can be obtained from the sample covariance:

$$\begin{aligned}\bar{E}[\hat{d}] &= \frac{1}{N_{boot}} \sum_{b=1}^{N_{boot}} \hat{d}^b \\ \bar{V}[\hat{d}] &= \frac{1}{N_{boot}} \sum_{b=1}^{N_{boot}} \frac{N_{boot}}{N_{boot} - 1} (\hat{d}^b - \bar{E}[\hat{d}])^2 \\ \bar{\Sigma}[\hat{d}_A, \hat{d}_C] &= \frac{1}{N_{boot} - 1} \sum_{b=1}^{N_{boot}} (\hat{d}_A^b - \bar{E}[\hat{d}_A])(\hat{d}_C^b - \bar{E}[\hat{d}_C])\end{aligned}\tag{18}$$

The \bar{V} and $\bar{\Sigma}$ notation highlights the fact these are sample-average estimators from the finite number of sampled values of \hat{d}^b , and so are themselves random variables. This is in contrast to the

estimate of variance for a sample-average estimator, which is analytically derived and so is fixed for a given set of sampled values. The estimators \bar{V} and $\bar{\Sigma}$ thus have a variance that depends on N_{boot} , and this is discussed further in Section 3.5.

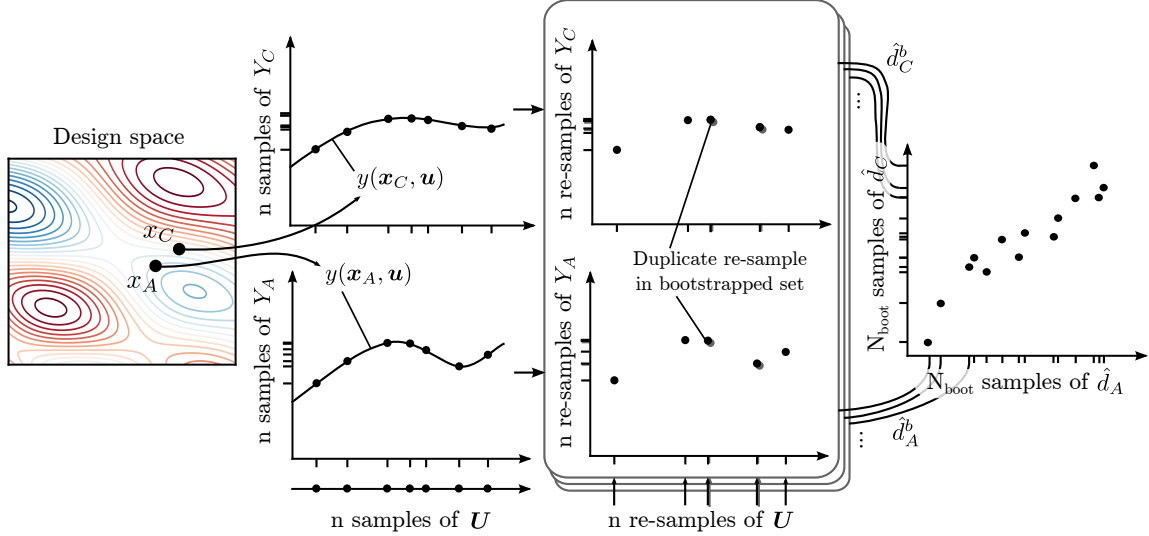


Figure 1: Illustration of the generalized information reuse concept

Note that the covariance $\Sigma[\hat{d}_A, \hat{d}_C]$ is different to the covariance between $y(\mathbf{x}_A, \mathbf{U}(\omega))$ and $y(\mathbf{x}_C, \mathbf{U}(\omega))$, which could be estimated analytically from the sampled values y_A^i and y_C^i $i = 1, \dots, n$.

This bootstrapping procedure is illustrated in Figure 1, and outlined in Algorithm 1. The same algorithm is used to obtain the regular MC estimator \hat{d}_A and its variance at a single design point, by ignoring all the steps involving \mathbf{x}_C . These estimated values of $V[\hat{d}_A]$, $V[\hat{d}_C]$ and $\Sigma[\hat{d}_A, \hat{d}_C]$, are substituted into Equation (16) to obtain an estimated optimal γ , which we denote as $\bar{\gamma}^*$. This $\bar{\gamma}^*$ is subsequently used to obtain an estimated variance $\bar{V}[\hat{s}_A]$ given by Equation (17).

Algorithm 1 Bootstrapping for variance and covariance of an MC estimator given n evaluations of y at design point \mathbf{x}_A , and at the control point \mathbf{x}_C , using N_{boot} sets of re-sampled values.

- 1: **for** $b = 1, \dots, N_{boot}$ **do**
 - 2: **for** $j = 1, \dots, n$ **do**
 - 3: $k \leftarrow$ random draw from $\{1, \dots, n\}$
 - 4: $y_A^j \leftarrow y(\mathbf{x}_A, \mathbf{U}^k)$ taken from sampled values $y_A^i, i = 1, \dots, n$
 - 5: $y_C^j \leftarrow y(\mathbf{x}_C, \mathbf{U}^k)$ taken from sampled values $y_C^i, i = 1, \dots, n$
 - 6: $\hat{d}_A^b \leftarrow$ MC estimator of d_A from re-sampled values $y_A^j, j = 1, \dots, n$
 - 7: $\hat{d}_C^b \leftarrow$ MC estimator of d_C from re-sampled values $y_C^j, j = 1, \dots, n$
 - 8: $\bar{E}[\hat{d}_A], \bar{E}[\hat{d}_C] \leftarrow \frac{1}{N_{boot}} \sum_{b=1}^{N_{boot}} \hat{d}_A^b, \frac{1}{N_{boot}} \sum_{b=1}^{N_{boot}} \hat{d}_C^b$
 - 9: $\bar{V}[\hat{d}_A], \bar{V}[\hat{d}_C] \leftarrow \frac{1}{N_{boot}-1} \sum_{b=1}^{N_{boot}} (\hat{d}_A^b - \bar{E}[\hat{d}_A])^2, \frac{1}{N_{boot}-1} \sum_{b=1}^{N_{boot}} (\hat{d}_C^b - \bar{E}[\hat{d}_C])^2$
 - 10: $\bar{\Sigma}[\hat{d}_A, \hat{d}_C] \leftarrow \frac{1}{N_{boot}-1} \sum_{b=1}^{N_{boot}} (\hat{d}_A^b - \bar{E}[\hat{d}_A])(\hat{d}_C^b - \bar{E}[\hat{d}_C])$
-

To use this estimation procedure within an optimization loop, a required acceptable variance for the estimator \hat{s}_A is specified (which for unbiased estimators corresponds to an acceptable MSE)

and $y(\mathbf{x}_A, \mathbf{u})$ and $y(\mathbf{x}_C, \mathbf{u})$ are evaluated at new realizations of $U(\omega)$ until the estimator variance is below this value. If the covariance between estimators at a particular design point and the control point is not sufficiently high, the IR estimator may require more computational effort than regular MC, since it requires evaluation of the system model y at both design points. In this case, recourse to regular Monte Carlo is required for the current design point, but in order to determine whether IR or regular MC requires more evaluations, a way of predicting how many sampled values each approach would use is required.

3.3 Predicting Number of Samples Required

Since a general quantity d with an unknown MC estimator \hat{d} is being considered, an assumption must be made about how the terms in Equation (17) vary as a function of n .

This paper specifically considers the quantile and the horsetail matching metric, and for both of these it is known that asymptotically the variance of their MC estimators is proportional to n^{-1} . Therefore for each term in Equation (17), a convergence function is fit proportional to n^{-1} through each term. For example for a variance term, $\bar{V} = v(n) = \bar{V}_0 \times (n/n_0)^{-1}$ where n_0 is the current number of sampled values and \bar{V}_0 is the current estimate of variance. In general, the same could be done for any function $v(n)$ for estimators with different convergence behavior.

This is done for the variance of \hat{d}_A , the variance of \hat{d}_C , and the covariance between \hat{d}_A and \hat{d}_C , giving the predicted functions of n v_A , v_C and $v_{A,C}$ respectively. Substituting these into each of the terms in Equation (17), with $V(\hat{s}_A)$ set to the required value V_{req} , gives a function in n whose roots give the predicted value of n —the number of sampled values of $Y_A(\omega)$ and $Y_C(\omega)$ required to reach the acceptable variance value:

$$\phi(n) = v_A(n) - \frac{(v^{A,C}(n))^2}{\bar{V}(\hat{s}_C) + v^C(n)} - V_{req}. \quad (19)$$

If there are multiple solutions to the equation the smallest is chosen, and then the solution is rounded down to the nearest integer to get the predicted value of n .

Clearly the further into the asymptotic convergence regime each of the terms in Equation (17) are the more accurate this prediction will be. This prediction is used at the start of each optimization iteration with an initial n_{init} evaluations y at both \mathbf{x}_A and \mathbf{x}_C , and thus n_{init} should be chosen to be sufficient to give a reasonable prediction. This approach is outlined in Algorithm 2.

Algorithm 2 Predicting the number of sampled values needed to reach a given acceptable estimator variance from estimates of variance and covariance at the current number n_0 .

- 1: $v_A(n) \leftarrow \bar{V}[\hat{d}_A]_0 \times n_0/n$
 - 2: $v_C(n) \leftarrow \bar{V}[\hat{d}_C]_0 \times n_0/n$
 - 3: $v^{A,C}(n) \leftarrow \hat{\Sigma}[\hat{d}_A, \hat{d}_C]_0 \times n_0/n$
 - 4: $\phi(n) \leftarrow$ predicted variance of \hat{s}_A from Equation (19)
 - 5: $n_{pred} \leftarrow \max(\inf\{n \mid \phi(n) = 0\}, 0)$
-

For the regular Monte Carlo estimator, this is applied to a single term: $n_{pred,MC} = n_0 \times \bar{V}(\hat{d}_A)_0/V_{req}$. Once it has been decided whether to revert back to regular Monte Carlo or continue with information reuse at a given design point, $y(\mathbf{x}_A, \mathbf{u})$ and $y(\mathbf{x}_C, \mathbf{u})$ are evaluated at additional sampled values of $U(\omega)$ until $\bar{V}(\hat{s}_A)$ is below the required level.

When using bootstrapping, estimating the variance terms is no longer at negligible computational cost, especially if evaluating \hat{d} requires sorting of the sampled values, since this is to be

done N_{boot} times. Therefore if only a small number of sampled values were added at a time, \hat{d} could be evaluated a large number of times and the computational cost may become non-negligible. While in practical problems, obtaining a sampled value of Y_x by evaluating the system model might far outweigh the cost of the bootstrapping procedure and so this is a non-issue, if the number of times bootstrapping is performed should be limited, Algorithm 2 can also be used to determine the number of additional realizations of Y_x to obtain.

When a small number of initial sampled values, n_{init} , are used, the prediction given by Algorithm 2 is likely to be inaccurate. Therefore to avoid over-estimating the value of n needed to give an acceptable variance, the number of sampled values of \mathbf{U} drawn at a time, at which the system model is evaluated, is limited to be a constant α times the current value of n . Additionally, to avoid iterating by a small amount many times close to the required number, sampled values are obtained to give a total of $(1 + \delta)n_{pred}$, where $\delta > 0$ is a small value. This paper uses $\alpha = 10$ and $\delta = 0.1$, but if evaluating the system model is far more expensive than bootstrapping, then $\delta = 0$ and a smaller α can be used.

3.4 Bias

If the estimator \hat{d} is unbiased, then the information reuse estimator is unbiased [23]. However, for a general quantity d with MC estimator \hat{d} , the bias of the information reuse estimator is given by:

$$B[\hat{s}_A] = B[\hat{d}_A] + \gamma \left(B[\hat{s}_C] - B[\hat{d}_C] \right), \quad (20)$$

where $B[\cdot]$ indicates the bias of an estimator. In the case where the bias of \hat{d} is independent of the number of sampled values of Y_x used to evaluate it, starting from the initial guess where $\hat{s}_C = \hat{d}_C$ from regular MC, the last two terms always cancel out as $B(\hat{s}_C) = B(\hat{d}_C)$; this is the case for the quantile estimator [47]. Therefore for the optimization formulations considered in this paper information reuse does not affect the bias of the estimator.

In general the bias can be estimated from bootstrapping using $\bar{B}[\hat{d}] = \left(\frac{1}{N_{boot}} \sum_{b=1}^{N_{boot}} \hat{d}^b \right) - \hat{d}$, and thus a bootstrapped estimate of the overall mean squared error (MSE) can be obtained from $\bar{V} + \bar{B}^2$. Therefore Equation (20) could be used to keep track of the bias of estimators as well as the variance, and a γ that minimizes the MSE could be chosen instead of one that minimizes variance.

3.5 Variance of bootstrapped estimators

The bootstrapped estimates \bar{V} and $\bar{\Sigma}$ are random variables that depend on the re-sampled values of \hat{d}^b . If N_{boot} is large enough their variance will be negligible, but since a finite number of sampled values have to be used in reality, the variance of the bootstrapped estimator of the IR estimator variance in Equation (15) is given by:

$$\begin{aligned} \text{Var}[\bar{V}[\hat{s}_A]] = & \text{Var}[\bar{V}_A] + \gamma^4 \text{Var}[\bar{V}_{S_C}] + \gamma^4 \text{Var}[\bar{V}_C] + 4\gamma^2 \text{Var}[\bar{\Sigma}] + \\ & 2\gamma^2 \text{Cov}[\bar{V}_A, \bar{V}_C] - 4\gamma \text{Cov}[\bar{V}_A, \bar{\Sigma}] - 4\gamma^3 \text{Cov}[\bar{V}_C, \bar{\Sigma}], \end{aligned} \quad (21)$$

where for clarity $\bar{V}[\hat{d}_A]$ is denoted by \bar{V}_A , $\bar{V}[\hat{d}_C]$ by \bar{V}_C , $\bar{V}[\hat{s}_C]$ by \bar{V}_{S_C} , and $\bar{\Sigma}[\hat{d}_A, \hat{d}_C]$ by $\bar{\Sigma}$.

Since \bar{V} and $\bar{\Sigma}$ are sample-average estimators, their variance is approximated by $\frac{1}{N_{boot}} \hat{\sigma}^2$ where $\hat{\sigma}^2$ is the sample variance of the values being averaged. With a reasonable value of N_{boot} , this variance will be small, however if γ is large, then the $\gamma^4 \text{Var}[\bar{V}_{S_C}]$ term in Equation (21) is potentially non-negligible. If the resulting value of $\text{Var}[\bar{V}[\hat{s}_A]]$, is then used as $\text{Var}[\bar{V}_{S_C}]$ in a subsequent

iteration, it could be magnified further leading to an exponentially growing inaccuracy in the estimated variance of the information reuse estimator and so giving meaningless results.

In information reuse, the optimal choice for γ is rarely greater than 1, and is often $\simeq 1$, so this is treated as a non-issue in this paper’s implementation. Further, the main application of this work is for problems where evaluating the system model y model is far more expensive than the bootstrapping procedure, and a large N_{boot} can be used to keep the variance of the bootstrapped estimators low. However, if desired, this variance can be kept track of in the optimization using Equation (21), and if it is deemed that the inaccuracy is too great, a MC estimator can be resorted back to in order to break the chain of dependence.

3.6 Overall Algorithm

The overall algorithm for performing an optimization using the proposed generalized information reuse method is given in Algorithm 3.

Algorithm 3 Optimization using generalized information reuse given $V_{required}$, α , δ , initial design point \mathbf{x}_0 , and n_{init} .

```

1:  $\mathbf{x}_A \leftarrow$  initial design point
2:  $n_{predicted}, n_{prev}, \bar{V}(\hat{\mathbf{s}}_A) \leftarrow n_{init}, n_{init}, +\infty$ 
3: while  $\bar{V}(\hat{\mathbf{s}}_A) > V_{required}$  do
4:    $n \leftarrow \min(\lfloor n_{predicted}(1 + \delta) \rfloor, \alpha \times n_{prev})$ 
5:   for  $j = n_{prev}, \dots, n$  do
6:      $\mathbf{u}^j \leftarrow$  sampled value of uncertainty vector from underlying distribution
7:      $y_A^j \leftarrow$  evaluated system model  $y(\mathbf{x}_A, \mathbf{u}^j)$ 
8:      $\hat{\mathbf{s}}_A \leftarrow \hat{d}_A$ , regular MC estimator using stored values of  $y_A^j, j = 1, \dots, n$ 
9:      $\bar{V}(\hat{\mathbf{s}}_A)$  from Algorithm 1 (ignoring steps involving  $\mathbf{x}_C$ )
10:     $n_{predicted} \leftarrow n \times \bar{V}(\hat{\mathbf{s}}_A)/V_{req}$ 
11: while Optimizer Not Converged do
12:    $\mathbf{x}_A \leftarrow$  next design point from optimizer
13:    $\mathbf{x}_C \leftarrow \min_k(\|\mathbf{x}_k - \mathbf{x}_A\|)$  from design points in optimization history
14:    $n_{predicted}, n_{prev}, \bar{V}(\hat{\mathbf{s}}_A) \leftarrow n_{init}, n_{init}, +\infty$ 
15:   RegularMonteCarlo  $\leftarrow$  False
16:   while  $\bar{V}(\hat{\mathbf{s}}_A) > V_{required}$  do
17:      $n_{prev} \leftarrow n$ 
18:      $n \leftarrow \min(\lfloor n_{predicted}(1 + \delta) \rfloor, \alpha \times n_{prev})$ 
19:     if RegularMonteCarlo then
20:       for  $j = n_{prev}, \dots, n$  do
21:          $\mathbf{u}^j \leftarrow$  sampled value of uncertainty vector from underlying distribution
22:          $y_A^j \leftarrow$  evaluated system model  $y(\mathbf{x}_A, \mathbf{u}^j)$ 
23:          $\hat{\mathbf{s}}_A \leftarrow \hat{d}_A$ , regular MC estimator using stored values of  $y_A^j, j = 1, \dots, n$ 
24:          $\bar{V}(\hat{\mathbf{s}}_A)$  from Algorithm 1 (ignoring steps involving  $\mathbf{x}_C$ )
25:          $n_{predicted} \leftarrow n \times \bar{V}(\hat{\mathbf{s}}_A)/V_{req}$ 
26:     else
27:       for  $j = n_{prev}, \dots, n$  do
28:          $\mathbf{u}^j \leftarrow$  sampled value of uncertainty vector from underlying distribution
29:          $y_A^j \leftarrow$  evaluated system model  $y(\mathbf{x}_A, \mathbf{u}^j)$ 
30:          $y_C^j \leftarrow$  evaluated system model  $y(\mathbf{x}_C, \mathbf{u}^j)$ 
31:          $\hat{d}_A, \hat{d}_C, \bar{V}(\hat{d}_A), \bar{V}(\hat{d}_C), \bar{\Sigma}(\hat{d}_A, \hat{d}_C)$  from Algorithm 1.
32:          $\gamma = \gamma^*$  from Equation (16)
33:          $\hat{\mathbf{s}}_A, \bar{V}(\hat{\mathbf{s}}_A)$  from Equations (14) and (15).
34:          $n_{predicted} \leftarrow$  Predicted required number of sampled values from Algorithm 2
35:         if  $n \times \bar{V}(\hat{d}_A)/V_{req} < 2n_{predicted}$  then
36:           RegularMonteCarlo,  $n_{predicted} \leftarrow$  True,  $n \times \bar{V}(\hat{d}_A)/V_{req}$ 

```

4 Numerical Experiments

In this section experiments are performed on an algebraic test problem and a physical acoustic horn design problem.

4.1 Algebraic Test Problem

The algebraic test problem uses two design variables, x_1 and x_2 , and n_u uncertain variables, u_1, \dots, u_{n_u} . There are two outputs:

$$y_0(\mathbf{x}, \mathbf{u}) = (2 - x_1)^2 + 2(x_2 - x_1^2)^2 + \exp\left(\sum_{i=1}^{n_u} \frac{1}{1+i} (1 + 0.2s_i^2)u_i\right) + 1,$$

$$y_1(\mathbf{x}, \mathbf{u}) = \exp\left(\sum_{i=1}^{n_u} \frac{1}{1+i} 0.2(1 + (s_i + 1)^2)u_i\right) - 2,$$

$$\text{where } s_i = \begin{cases} x_2, & \text{if } i \text{ even} \\ x_1, & \text{if } i \text{ odd} \end{cases}$$

The bounds on both design variables are $[-1, 1]$, and the uncertain inputs are all uniformly distributed over $[-1, 1]$.

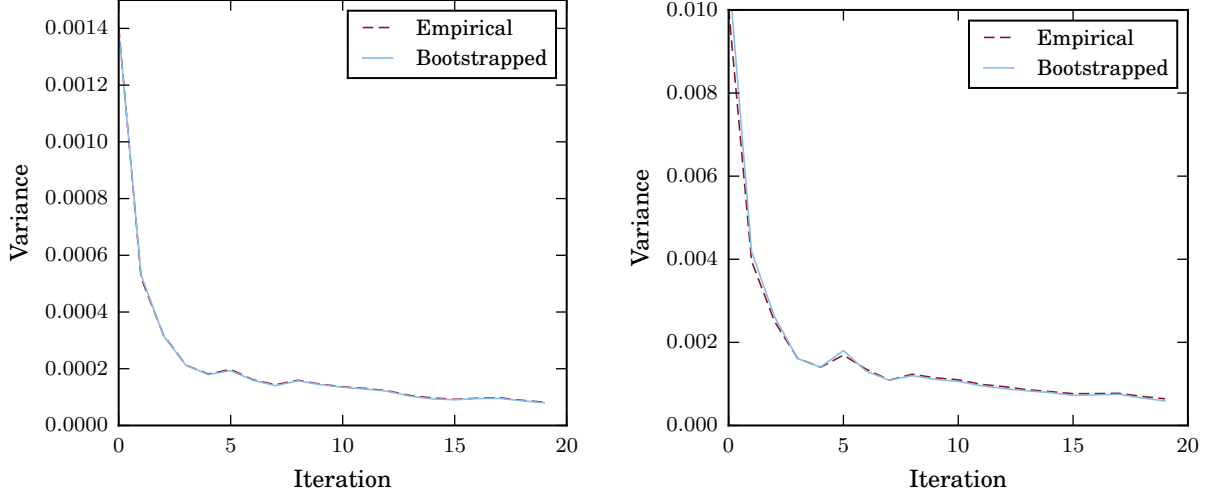
Due to the stochastic nature of using Monte Carlo sampling to evaluate objectives and constraints, an optimization algorithm that is resistant to a small amount of noise is desirable. Derivative-free methods sample over relatively large regions of design space, which can smooth out the effect of the sampling noise, and so are suitable for sampling based optimization under uncertainty. The derivative-free optimizer COBYLA [48] is used here, where it is run until the magnitude of the change in design vectors is less than 10^{-2} ; it is implemented using the NLOpt toolbox [49].

4.1.1 Validation

Firstly, in order to validate general information reuse using bootstrapping, the variance of the general information reuse estimator predicted by Equation (15) and Algorithm 1 is compared to an empirically generated variance (both using $N_{boot} = 5000$). To achieve this, using $n_u = 2$, a sequence of design points are taken from an optimization and fixed, then the information reuse estimator and its variance are obtained using the method outlined in Section 3 at each point as it would be in an optimization, but using a fixed ($N = 1000$) number of sampled values. This is repeated 5000 times; the average of the information reuse variances gives the theoretical bootstrapped curve on Figure 2 and the variance of the estimators gives the empirical curve on Figure 2. This validation process is performed when estimating both the horsetail matching metric with $t(h) = -h^4$ and the 90% quantile of the random variable defined by y_0 from equation 4.1; both of these curves show good agreement.

4.1.2 Optimization Acceleration

Next, using $n_u = 10$, the horsetail matching metric for y_0 is minimized subject to the 90% quantile function on y_1 being less than zero (indicating a 90% probability of constraint satisfaction). This



(a) Horsetail matching metric with $t(h) = -h^4$

(b) Quantile function $F^{-1}(0.9)$

Figure 2: Comparison, for two non-sample-average estimators, of the variance predicted by general information reuse and the empirical variance from 200 runs

gives the following optimization problem:

$$\begin{aligned} \underset{\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U}{\text{minimize}} \quad d_{hm} &= \left(\int_0^1 (F_0^{-1}(h) - t(h))^2 dh \right)^{1/2} \\ \text{s.t.} \quad &F_1^{-1}(0.9) \leq 0 \end{aligned} \quad (22)$$

where F_0^{-1} and F_1^{-1} are the inverse CDFs (quantile function) for the random variables given by $y_0(\mathbf{x}, \mathbf{U}(\omega))$ and $y_1(\mathbf{x}, \mathbf{U}(\omega))$ respectively. These choices for objective and constraints are propagated as discussed in Sections 2.5 and 2.6—neither of their estimators can be expressed as a sample-average. For this problem, we use the risk-averse target $t(h) = -(4h^6)$ in the horsetail matching metric. This target is risk averse because it emphasizes minimizing values of the quantile function $F_0^{-1}(h)$ for values of h close to 1 more than minimizing values of h close to 0, hence preferring designs with better worst-case behavior. We refer the reader to Refs. 44 and 50 for further discussion on horsetail matching targets.

We use a required variance of 1×10^{-3} for the objective function and 4×10^{-3} for the constraint. In preliminary experiments, using these values in a regular MC estimator of the objective required similar computational cost to a regular MC estimator of the constraint, facilitating ease of illustration on Figure 3. The influence of this choice of required variance is discussed in Section 4.1.3. The optimization is run from an initial design point of $\mathbf{x}_0 = [-1, -1]$; additionally $n_{init} = 50$, $\alpha = 10$, $\delta = 0.1$.

We run optimizations using only regular Monte Carlo (labeled MC) and using generalized information reuse (labeled gIR). Figure 3 gives the convergence, in terms of computational cost (in terms of number of evaluations of $y(\mathbf{x}, \mathbf{u})$), of the objective function (the horsetail matching metric) and constraint (quantile function), as well as the computational cost of each optimization iteration. The optimization using information reuse achieves significant computational savings compared to regular Monte Carlo, in both objective function and constraint evaluations.

Additionally we run optimizations where the input parameters are distributed according to a gamma distribution (instead of being uniformly distributed) with shape parameter 2 and scale

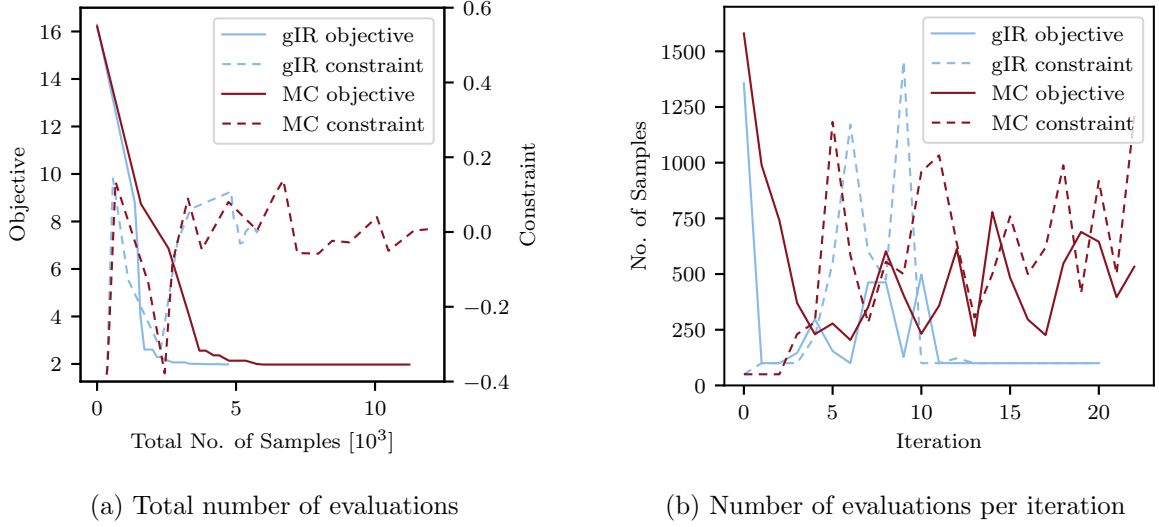


Figure 3: Optimization progress of generalized information reuse and regular Monte Carlo

parameter 0.5, such that u_i is the realization of a random variable given by $1 - U_i(\omega)$ where $U_i \sim \Gamma(2, 0.5)$, for $i = 1, \dots, n_u$. This random variable is skewed and has support $(-\infty, 1]$. The results of optimizations with these input uncertainties are given in Figure 4.

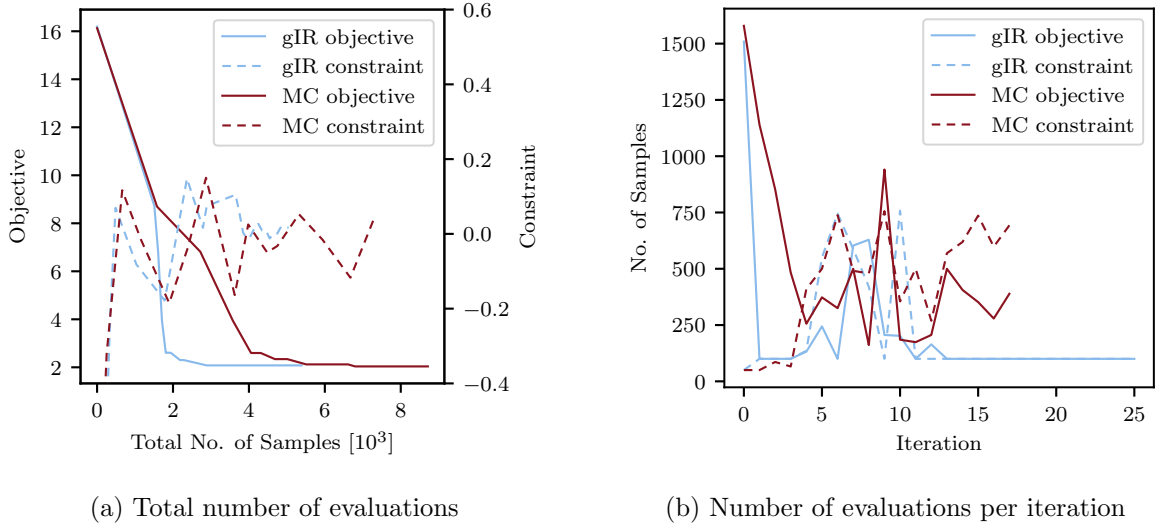
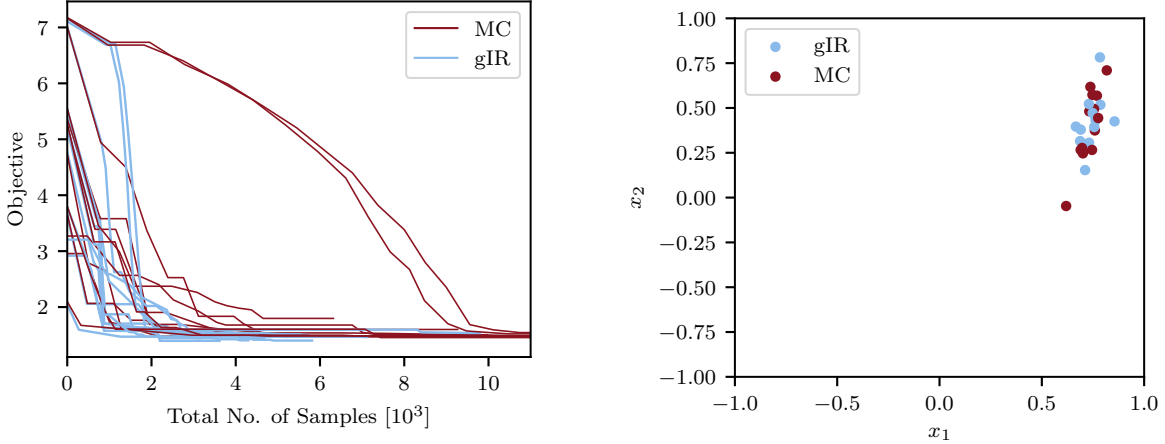


Figure 4: Optimization progress of generalized information reuse and regular Monte Carlo, using input parameters distributed according to a Gamma distribution

On Figure 4, information reuse performs similarly to Figure 3 (when the uncertainties were uniformly distributed). This is because the distributions of $Y_A(\omega)$ and $Y_C(\omega)$, and hence the correlation between \hat{d}_A and \hat{d}_C , depend on the non-linearity and variance of the system model as well as the distribution of $\mathbf{U}(\omega)$. Since the non-linear test function given by Equation (4.1) is being used here, changing the distribution of $\mathbf{U}(\omega)$ does not significantly affect the ability of information reuse to improve upon regular Monte Carlo.

The COBYLA algorithm is not guaranteed to converge to the true optimum, and due to the

stochastic nature of the problem it is not expected that the same design is obtained each time an optimization is performed. To investigate this, we run the optimizations an additional 13 times from random initial design points (with uniformly distributed input uncertainties), and plot the convergence of the objective function on Figure 5, along with the optimal design point obtained from each run. From Figure 5 we can see that the benefit of information reuse is not specific to a particular initial design point. Additionally, the scatter in the optimum design point obtained by the different runs highlights the noisy nature of using MC sampling for OUU, and that the required variance must be low enough for the results of the OUU to be meaningful.



(a) Convergence of objective with total number of evaluations

(b) Final design point

Figure 5: Results of multiple optimizations of the algebraic test problem using COBYLA.

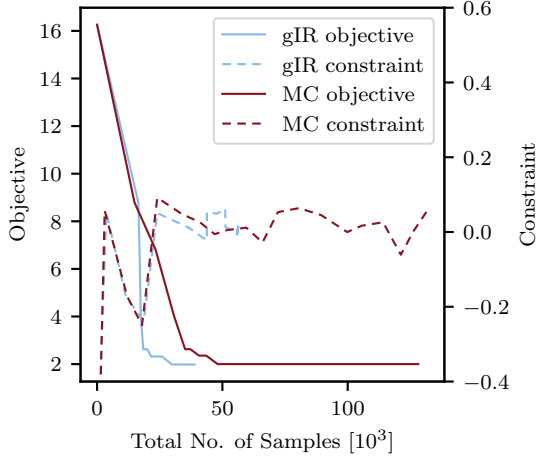
4.1.3 Influence of Required Variance

To investigate the impact of the chosen required variance values, we run an optimization using values an order of magnitude lower than in Section 4.1.2 (so 1×10^{-4} for the objective and 4×10^{-4} for the constraint); the results are given in Figure 6.

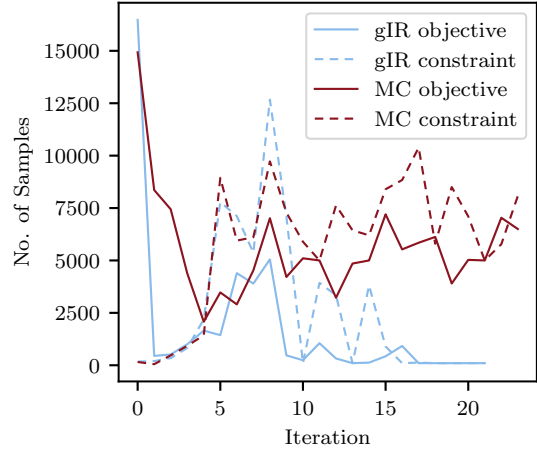
The overall cost of the optimization is roughly an order of magnitude higher than the previous optimizations (note the scale on the y-axis) since, as discussed in Section 2.2, the variance of the MC estimators considered here decreases proportional to n^{-1} . Information reuse still offers significant speedup over regular MC, indicating that while the choice of required variance influences the overall cost of the OUU, information reuse is still able to reduce this cost.

Additionally we can see that, compared to Figure 3, more optimization iterations are needed before information reuse is able to obtain the required variance values using only n_{init} sampled values. This is because, for only n_{init} sampled values to be needed to obtain a lower required variance, a larger correlation between estimators at the design point and control point is needed; therefore the design points need to be closer together, which occurs later in the optimization run.

In some cases it is possible for information reuse to be more expensive than regular MC: if n_{init} is chosen to be too high (relative to the required variances) such that regular MC can achieve the required variance in less than n_{init} sampled values, information reuse wastes the n_{init} evaluations of the system model at the control point. For example Figure 7 gives the results of an optimization where the required variances are set to be an order of magnitude higher than in Section 4.1.2, (so



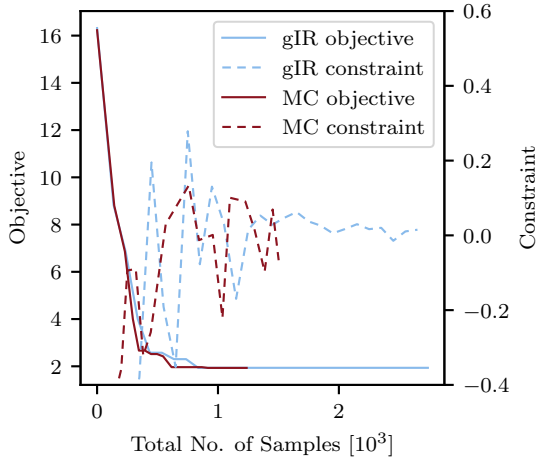
(a) Total number of evaluations



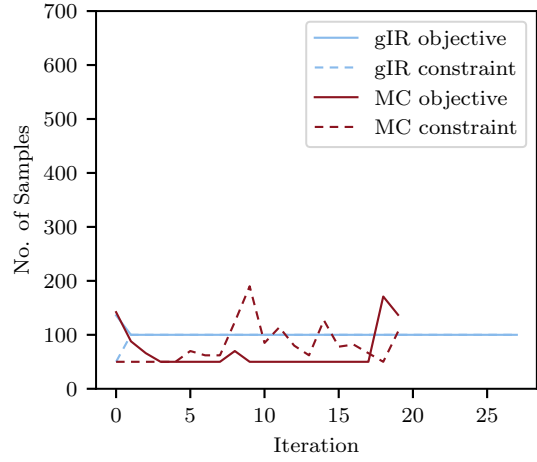
(b) Number of evaluations per iteration

Figure 6: Optimization progress of generalized information reuse and regular Monte Carlo, using lower required variances

1×10^{-2} for the objective and 4×10^{-2} for the constraint) and a value of $n_{init} = 50$ is used.



(a) Total number of evaluations



(b) Number of evaluations per iteration

Figure 7: Optimization progress of generalized information reuse and regular Monte Carlo, using higher required variances

For the majority of design points, the n_{init} sampled values were enough to achieve the required variance using regular MC, and so information reuse wasted the additional n_{init} system model evaluations at the control point. In this case this required variance is so low that the results of this OUU are unlikely to be meaningful. In design scenarios where a designer has a reasonable idea of the computational cost needed to reach their required variance, they should be able to choose a sensible value of n_{init} .

If only a very few number of model evaluations can be afforded such that evaluating n_{init} sampled values at both the current design point and control point is too expensive, then the

current implementation will perform poorly. However such a computational expense implies that using MC sampling to propagate the uncertainties is perhaps not the best choice in the first place.

4.2 Application to Acoustic Horn Design Problem

This problem uses a simulation of a 2D acoustic horn governed by the non-dimensional Helmholtz equation. An incoming wave enters the horn through the inlet and exits the outlet into the exterior domain with a truncated absorbing boundary [23,51]. The flow field is solved using a reduced basis method obtained by projecting the system of equations of a high-fidelity finite element discretization onto a reduced subspace [52]; here 100 reduced basis functions are used.

This system’s single output of interest, given by $y(\mathbf{x}, \mathbf{u})$, is the reflection coefficient, which is a measure of the horn’s efficiency. The geometry of the horn is parametrized by the widths at six equally spaced axial positions from root to tip. The design variables give the nominal value of these widths $x_{k,nom}$, and then uncertainty is introduced representing manufacturing tolerances such that the actual value x_k is given by a random variable centered on $x_{k,nom}$. Additionally the wavenumber of the incoming wave and the acoustic impedance of the upper and lower horn walls are uncertain input parameters. The bounds for the design variables are given in Table 1, and the uncertainty distributions are given in Table 2.

No.	Notation	Lower Bound	Upper Bound
1	$x_{1,nom}$	0.68	1.01
2	$x_{2,nom}$	1.04	1.38
3	$x_{3,nom}$	1.41	1.73
4	$x_{4,nom}$	1.77	1.99
5	$x_{5,nom}$	2.04	2.43
6	$x_{6,nom}$	2.48	2.82

Table 1: Design variables for the acoustic horn design problem

No.	Uncertain Variable	Notation	Distribution	Lower Bound	Upper Bound
1	Wavenumber	k	Uniform	1.25	1.55
2	Lower wall impedance	z_u	Uniform	10	90
3	Upper wall impedance	z_l	Uniform	10	90
4	Geometry	x_1	Uniform	$0.975x_{1,nom}$	$1.025x_{1,nom}$
5	Geometry	x_2	Uniform	$0.975x_{2,nom}$	$1.025x_{2,nom}$
6	Geometry	x_3	Uniform	$0.975x_{3,nom}$	$1.025x_{3,nom}$
7	Geometry	x_4	Uniform	$0.975x_{4,nom}$	$1.025x_{4,nom}$
8	Geometry	x_5	Uniform	$0.975x_{5,nom}$	$1.025x_{5,nom}$
9	Geometry	x_6	Uniform	$0.975x_{6,nom}$	$1.025x_{6,nom}$

Table 2: Uncertain inputs in the acoustic horn design problem

As an illustration of the problem, the magnitudes of the complex acoustic pressure field for the horn geometries given by the lower and upper bounds on the design variables are shown in Figure 8;

the reflection coefficient, at nominal values of the uncertain inputs (the mean of their distribution) is also given.

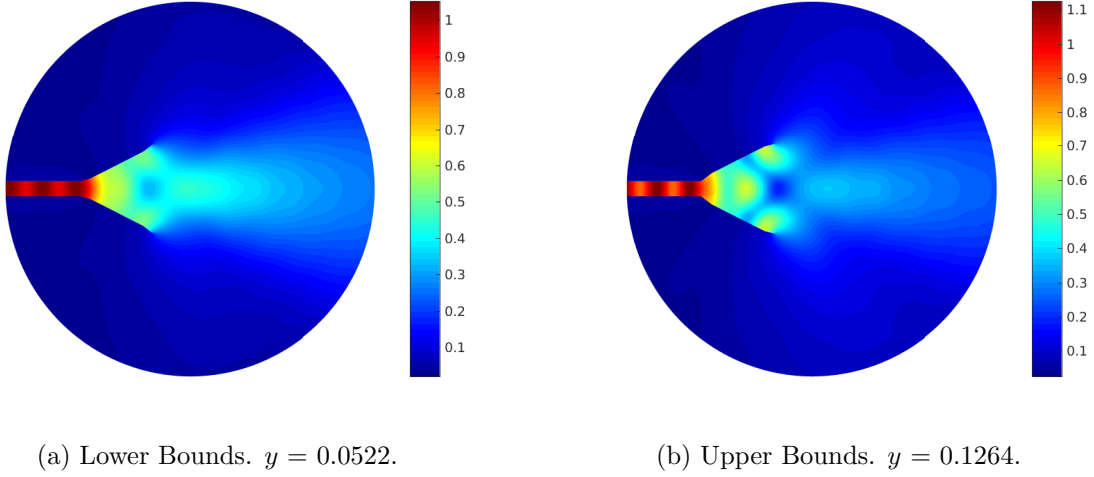


Figure 8: Magnitude of the complex acoustic pressure field for the horn geometry given by the lower and upper bounds on the design variables.

While derivative-free optimizers such as COBLYA are resistant to a small amount of noise, they are local optimizers that will converge to a local minimum, and the number of iterations required for convergence is sensitive to the initial design point. Therefore in order to fairly compare multiple optimizations on this design problem and so more thoroughly compare information reuse with regular MC, a global optimizer is used. Here the evolutionary strategy optimizer from the *ecspy* python toolbox² is used, run with a population size of 25 until a total of 10^6 acoustic horn model evaluations have been sampled. This represents a design scenario in which a fixed computational budget is permitted, and enables a consistent comparison between regular MC and information reuse optimizations over multiple optimization runs.

Firstly for reference, a classical approach to robust optimization is used and a weighted sum of the first two statistical moments of $Y_x(\omega) = y(\mathbf{x}, \mathbf{U}(\omega))$ is minimized using regular information reuse and it is compared to regular Monte Carlo. Then the horsetail matching metric under a risk averse target is minimized using generalized information reuse. In both cases $n_{init} = 100$, $\delta = 0.1$, and $\alpha = 10$.

The weighted sum of moments formulation is solving the following optimization problem:

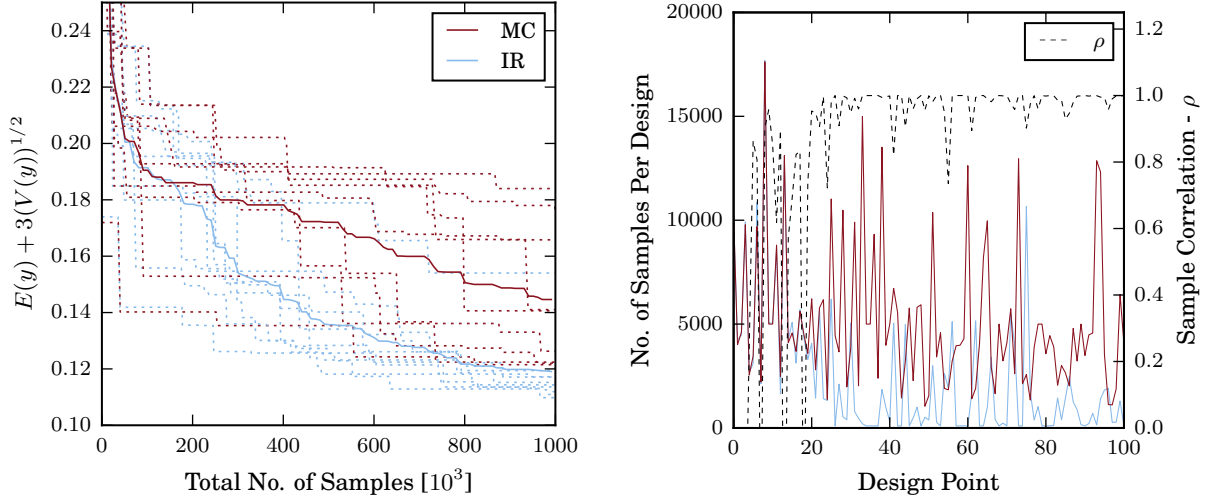
$$\underset{\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U}{\text{minimize}} \quad d_{ws} = E(Y_x(\omega)) + 3\sqrt{V(Y_x(\omega))} \quad (23)$$

This function d_{ws} is a combination of two quantities whose estimators are sample averages, and so the variance of its estimator can be estimated from a first order Taylor expansion of d_{ws} about estimators of $E(Y(\omega))$ and $V(Y(\omega))$, both of which can be evaluated using regular information reuse [23, 39].

A required variance similar to that used to demonstrate the original information reuse formulation in Ref. 23 is used here: a value of $V_{req} = 2 \times 10^{-6}$. The optimization is run from 10 random starting populations, such that one optimization that uses regular information reuse (labeled IR)

²<https://pypi.python.org/pypi/ecspy>

and one that uses regular Monte Carlo (labeled MC) is performed from each starting point. Figure 9a gives the convergence of all the IR and MC optimizations. Figure 9b gives the number of sampled values needed to reach the required variance for the first 100 design candidates in the optimization for both techniques for a single optimization run, as well as the correlation coefficient between sampled values at the design point and control point for the IR run.



(a) Convergence with total evaluations for multiple (b) Number of evaluations required per iteration for optimizations (dashed), and their average (solid). a typical optimization

Figure 9: Regular information reuse and regular Monte Carlo optimizers minimizing a weighted sum of mean and variance of the reflection coefficient.

The results on Figure 9 demonstrate how regular information reuse leads to computational savings compared to regular MC. On average the IR optimizations achieved the objective function value that the regular MC optimizations took 10^6 evaluations to obtain in $\simeq 2.5 \times 10^5$ evaluations. Additionally it can be seen how for the IR optimization, the computational cost is smallest when the correlation between sampled values at the current design point and control point are close to one; the spikes on Figure 9b where the computational cost of IR increases correspond to iterations where the correlation drops significantly below 1.

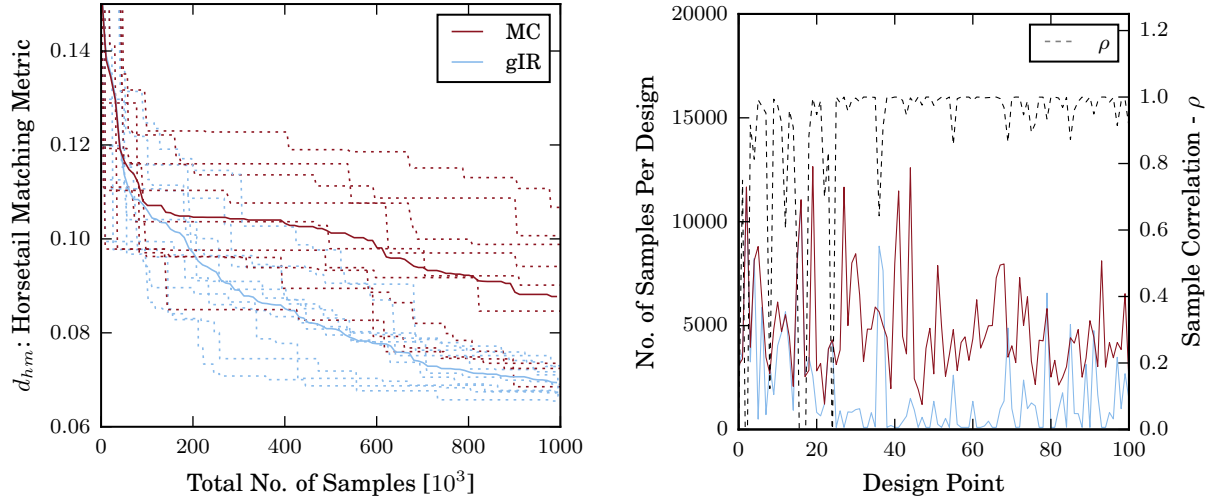
Next generalized information reuse is used to optimize the horsetail matching metric, giving the following optimization problem:

$$\underset{\mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U}{\text{minimize}} \quad d_{hm} = \left(\int_0^1 (F^{-1}(h) - t(h))^2 dh \right)^{1/2}, \quad (24)$$

where F^{-1} is the inverse CDF (quantile function) for the reflection coefficient given by the random variable $Y(\omega) = y(\mathbf{x}, \mathbf{U}(\omega))$ at a given design. The risk averse target $t(h) = -0.05h^4$ and a required variance of 4×10^{-7} are used. Again $n_{init} = 100$, $\delta = 0.1$, and $\alpha = 10$.

Figure 10a plots the convergence for both generalized information reuse (labeled gIR) and MC optimizations. Figure 10b gives the number of evaluations needed to reach the required variance for the first 100 design points evaluated, along with correlation between sampled values at the current design point and the control point for a single optimization run.

Comparing Figures 9 and 10, it appears that similar computational gains are achieved by the general information reuse approach with a non-sample-average estimator as regular information



(a) Convergence with total evaluations for multiple (b) Number of evaluations required per iteration for optimizations (dashed), and their average (solid). a typical optimization

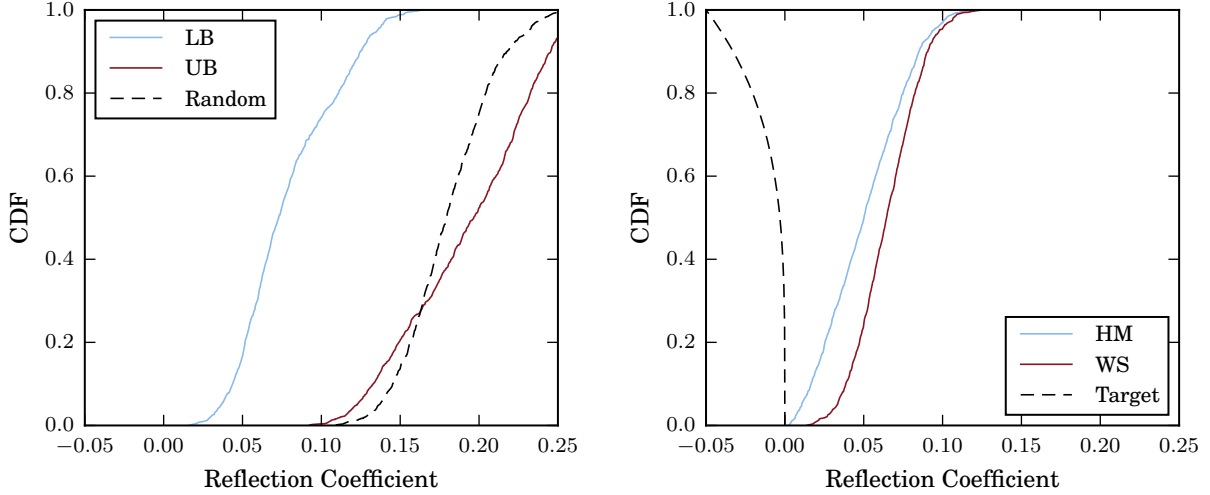
Figure 10: General information reuse and regular Monte Carlo optimizers minimizing the horsetail matching metric of the reflection coefficient with a risk averse target ($t(h) = -0.05h^4$).

reuse with a sample-average estimator. Once again the gIR optimizations achieved the objective function value that the regular MC optimizations took 10^6 evaluations to obtain in $\simeq 2.5 \times 10^5$ evaluations.

It can also be observed that in both cases, there is no significant benefit of IR and gIR over regular MC until after $\simeq 100 \times 10^3$ evaluations. This is due to the fact that the evolutionary strategy optimizer deals with populations of solutions, and so the first population is likely to be spread out in design space with little correlation between design points. This is also observable on Figures 10b and 9b, where for the first $\simeq 20$ designs evaluated, the IR and MC optimization runs required similar number of evaluations.

To compare the designs obtained from these optimizations, an empirical CDF is obtained using 1000 sampled values of $Y_A(\omega)$ for the best design from all 10 runs in both the weighted sum and horsetail matching cases, which are plotted on Figure 11b (along with the target used in the horsetail matching formulation). Additionally, for reference an empirical CDF is obtained for the designs at the upper and lower design variable bounds along with a random point in design space and these are plotted on Figure 11a; these CDFs illustrates the magnitude of the variation of the reflection coefficient induced by the uncertain input parameters, along with the variation in shape of CDF over design space.

Figure 11 demonstrates that using the weighted sum of moments for this problem has given rise to a design that is stochastically dominated by the design obtained by using the horsetail matching metric: the CDFs of these two designs do not cross. This highlights part of the goal in developing this generalized information reuse: certain quantities, whose estimators are not sample averages, can offer more powerful formulations for design under uncertainty, but one does not wish to take a computational penalty for using them over more traditional quantities with sample average estimators. While it is difficult to quantitatively compare the generalized information reuse approach optimizing the horsetail matching metric to the regular information reuse approach optimizing a weighted sum of moments, since they use different objectives and different values



(a) CDFs of $Y_x(\omega)$ for the lower and upper bounds (b) CDFs of $Y_x(\omega)$ for designs from optimizations of a on the design variables, and the CDF for a random weighted sum of mean and standard deviation (WS) design point. and the horsetail matching metric (HM).

Figure 11: Empirical CDFs of notable designs in the acoustic horn design problem.

for the required variances, the similarity between the advantage of general information reuse over regular MC on Figure 10 and the advantage of regular information reuse over regular MC on Figure 9 suggests that this goal has been achieved.

5 Conclusions

Using Monte Carlo (MC) sampling within optimization under uncertainty (OUU) is appropriate when conditions favourable for other, more efficient, uncertainty propagation methods are not present (e.g. when the number of uncertainties is too high or the problem is not sufficiently smooth). OUU using MC sampling requires a relatively large number of system evaluations so may be infeasible for computationally expensive applications, however multi-fidelity methods for MC sampling can reduce the computational cost enough for it to become feasible in many cases.

Information reuse is a multi-fidelity method for MC sampling within OUU that treats neighboring design points in an optimization as lower fidelity models, meaning it is a multi-fidelity method that can be applied to problems where only a single fidelity model is available. This paper proposes a generalized information reuse approach that extends to optimization formulations using quantities whose estimators that cannot be expressed as a sample average.

On an algebraic test problem, the generalized information reuse accurately predicts the variance of two non-sample average estimators, and when used in a derivative-free local optimizer gives significant computational savings over regular Monte Carlo sampling. On a physical acoustic horn design problem, when used in an evolutionary strategy global optimizer, optimizations of the horsetail matching metric using generalized information reuse provide similar computational savings over regular Monte Carlo to optimizations of a weighted sum of mean and standard deviation using regular information reuse. Additionally, the optimal design of the horsetail matching metric stochastically dominates the optimal design of weighted sum of moments, highlighting the importance of being able to use quantities without sample average estimators in optimization.

6 Acknowledgements

This work is funded in part by the Engineering and Physical Sciences Research Council (EPSRC) UK, grant number EP/L504920/1, and in part by the Air Force Office of Scientific Research (AFOSR) MURI on managing multiple information sources of multi-physics systems, Program Manager F. Fahroo, Award Number FA9550-15-1-0038.

References

- [1] Kennedy, M. C. and O’Hagan, A., “Bayesian Calibration of Computer Models,” *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, Vol. 63, No. 3, 2001, pp. 425–464.
- [2] Enevoldsen, I. and Sørensen, J., “Reliability-Based Optimization in Structural Engineering,” *Structural Safety*, Vol. 15, No. 3, 1994, pp. 169–196.
- [3] Beyer, H. G. and Sendhoff, B., “Robust Optimization - A Comprehensive Survey,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 196, No. 33, 2007, pp. 3190–3218.
- [4] Schueller, G. I. and Jensen, H. A., “Computational Methods in Optimization Considering Uncertainties - An Overview,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 1, 2008, pp. 2–13.
- [5] Gabrel, V., Murat, C., and Thiele, A., “Recent Advances in Robust Optimization: An Overview,” *European Journal of Operational Research*, Vol. 235, No. 3, 2014, pp. 471–483.
- [6] Beck, A. T., Gomes, W. J. S., Lopez, R. H., and Miguel, L. F. F., “A comparison between robust and risk-based optimization under uncertainty,” *Structural and Multidisciplinary Optimization*, Vol. 52, No. 3, 2015, pp. 479–492.
- [7] Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L., and Newman, P. A., “Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models,” *Journal of Aircraft*, Vol. 38, No. 6, 2001, pp. 1093–1101.
- [8] Allaire, D. and Willcox, K., “A mathematical and computational framework for multifidelity design and analysis with computer models,” *International Journal on Uncertainty Quantification*, Vol. 4, No. 1, jan 2013, pp. 1–20.
- [9] Eldred, M. S., “Design Under Uncertainty Employing Stochastic Expansion Methods,” *International Journal for Uncertainty Quantification*, Vol. 1, No. 2, 2011, pp. 119–146.
- [10] Eldred, M., Giunta, A., and Collis, S., “Second-Order Corrections for Surrogate-Based Optimization with Model Hierarchies,” *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.
- [11] Choi, S., Alonso, J. J., and Kroo, I. M., “Two-Level Multifidelity Design Optimization Studies for Supersonic Jets,” *Journal of Aircraft*, Vol. 46, No. 2, 2009, pp. 776–790.
- [12] Keane, A. J., “Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods,” *Journal of Aircraft*, Vol. 40, No. 4, 2003, pp. 741–750.
- [13] Balabanov, V., Haftka, R., Grossman, B., Mason, W., and Watson, L., “Multifidelity Response Surface Model for {HSCT} Wing Bending Material Weight,” *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, MO, sep 1998.

- [14] Balabanov, V. and Venter, G., “Multi-Fidelity Optimization with High-Fidelity Analysis and Low-Fidelity Gradients,” *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, aug 2004.
- [15] Lee, S. H. and Chen, W., “A Comparative Study of Uncertainty Propagation Methods for Black-Box-Type Problems,” *Structural and Multidisciplinary Optimization*, Vol. 37, No. 3, 2009, pp. 239–253.
- [16] Nam, H. K., Wang, H., and Queipo, N. V., “Efficient Shape Optimization Under Uncertainty Using Polynomial Chaos Expansions and Local Sensitivities,” *AIAA Journal*, Vol. 44, No. 5, 2006, pp. 1112–1115.
- [17] Zhang, Y. and Hosder, S., “Robust Design Optimization Under Mixed Uncertainties With Stochastic Expansions,” *Journal of Mechanical Design*, Vol. 135, 2013, pp. 081005.
- [18] Rumpfkeil, M. P., “Robust Design Under Mixed Aleatory / Epistemic Uncertainties Using Gradients and Surrogates,” *Journal of Uncertainty Analysis and Applications*, Vol. 1, No. 1, 2013.
- [19] Zhang, J., Taflanidis, A. A., and Medina, J. C., “Sequential approximate optimization for design under uncertainty problems utilizing Kriging metamodeling in augmented input space,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 315, 2017, pp. 369–395.
- [20] Lemieux, C., *Monte Carlo and Quasi-Monte Carlo Sampling*, Springer, 2009.
- [21] Smith, R. C., *Uncertainty Quantification: Theory, Implementation, and Applications.*, SIAM, 2013.
- [22] Peherstorfer, B., Willcox, K. E., and Gunzburger, M., “Optimal Model Management for Multifidelity Monte Carlo Estimation,” *SIAM Journal on Scientific Computing*, 2016.
- [23] Ng, L. W. T. and Willcox, K. E., “Multifidelity Approaches for Optimization Under Uncertainty,” *International Journal for Numerical Methods in Engineering*, 2014.
- [24] Göhler, S. M., Eifler, T., and Howard, T. J., “Robustness Metrics: Consolidating the Multiple Approaches to Quantify Robustness,” *Journal of Mechanical Design*, Vol. 138, No. 11, 2016, pp. 111407.
- [25] Seshadri, P., Constantine, P., Iaccarino, G., and Parks, G., “A Density-Matching Approach for Optimization Under Uncertainty,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 305, 2016, pp. 562–578.
- [26] Quagliarella, D., Petrone, G., and Iaccarino, G., “Optimization Under Uncertainty Using the Generalized Inverse Distribution Function,” *Modeling, Simulation and Optimization for Science and Technology*, 2014.
- [27] Quagliarella, D. and Iuliano, E., “Robust Design of a Supersonic Natural Laminar Flow Wing-Body,” *IEEE Computational Intelligence Magazine*, 2017.
- [28] Efron, B., “Bootstrap Methods. Another Look at the Jackknife,” *Annals of Statistics*, Springer, New York, NY, 1979, pp. 569–593.
- [29] Mulvey, J. M., Vanderbei, R. J., and Zenios, S. A., “Robust Optimization of Large-Scale Systems,” *Operations Research*, Vol. 43, No. 2, 1995, pp. 264–281.

- [30] Ghisu, T., Jarrett, J. P., and Parks, G. T., “Robust Design Optimization of Airfoils with Respect to Ice Accretion,” *Journal of Aircraft*, Vol. 48, No. 1, 2011, pp. 287–304.
- [31] Lee, D. S., Periaux, J., Onate, E., Gonzalez, L. F., and Qin, N., “Active Transonic Aerofoil Design Optimization Using Robust Multiobjective Evolutionary Algorithms,” *Journal of Aircraft*, Vol. 48, No. 3, 2011, pp. 1084–1094.
- [32] Keane, A. J., “Comparison of Several Optimization Strategies for Robust Turbine Blade Design,” *Journal of Propulsion and Power*, Vol. 25, No. 5, 2009, pp. 1092–1099.
- [33] Dodson, M. and Parks, G. T., “Robust Aerodynamic Design Optimization Using Polynomial Chaos,” *Journal of Aircraft*, Vol. 46, No. 2, 2009, pp. 635–646.
- [34] Padulo, M., Campobasso, M. S., and Guenov, M. D., “Novel Uncertainty Propagation Method for Robust Aerodynamic Design,” *AIAA Journal*, Vol. 49, No. 3, 2011, pp. 530–543.
- [35] Lee, S. W. and Kwon, O. J., “Robust Airfoil Shape Optimization Using Design for Six Sigma,” *Journal of Aircraft*, Vol. 43, No. 3, 2006, pp. 843–846.
- [36] Parkinson, A., Sorensen, C., and Pourhassan, N., “A General Approach for Robust Optimal Design,” *Journal of Mechanical Design*, Vol. 115, No. 1, 1993, pp. 74–80.
- [37] Park, G.-J., Lee, T.-H., Lee, K. H., and Hwang, K.-H., “Robust Design: An Overview,” *AIAA Journal*, Vol. 44, No. 1, 2006, pp. 181–191.
- [38] Paiva, R. M., Crawford, C., and Suleman, A., “Robust and Reliability-Based Design Optimization Framework for Wing Design,” *AIAA Journal*, Vol. 52, No. 4, 2014, pp. 711–724.
- [39] Ng, L. W. T. and Willcox, K. E., “Monte Carlo Information Reuse Approach to Aircraft Conceptual Design Optimization Under Uncertainty,” *Journal of Aircraft*, Vol. 53, No. 2, 2016, pp. 427–438.
- [40] Shapiro, A., Dentcheva, D., and Ruszczyński, A., *Lectures on Stochastic Programming: Modeling and Theory*, SIAM, 2009.
- [41] Pflug, G. P., “Some Remarks on the Value-at-Risk and the Conditional Value-at-Risk,” *Probabilistic Constrained Optimization*, Vol. 49, Kluwer Academic Publishers, Dordrecht., pp. 272–281.
- [42] Alexander, S., Coleman, T. F., and Li, Y., “Minimizing CVaR and VaR for a Portfolio of Derivatives,” *Journal of Banking and Finance*, Vol. 30, No. 2, 2006, pp. 583–605.
- [43] David, C. and Steinberg, S., *Quantile estimation*, Wiley, New York, 1986.
- [44] Cook, L. W. and Jarrett, J. P., “Horsetail Matching: A Flexible Approach to Optimization Under Uncertainty,” *Engineering Optimization*, 2017.
- [45] Nelson, B. L., “On Control Variate Estimators,” *Computers and Operations Research*, Vol. 14, No. 3, 1987, pp. 219–225.
- [46] Davison, *Bootstrap Methods and Their Applications*, Cambridge University Press, 1999.
- [47] Okolewski, A. and Rychlik, T., “Sharp Distribution-Free Bounds on the Bias in Estimating Quantiles via Order Statistics,” *Statistics and Probability Letters*, Vol. 52, No. 2, 2001, pp. 207–213.

- [48] Powell, M. J. D., “COBYLA, A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation,” *Advances in Optimization and Numerical Analysis*, Vol. 275, No. 1, 1994, pp. 51–67.
- [49] Johnson, S. G., “The NLOpt Nonlinear-Optimization Package,” <http://ab-initio.mit.edu/nlopt>, 2017.
- [50] Cook, L. W., Jarrett, J. P., and Willcox, K. E., “Extending Horsetail Matching for Optimization Under Probabilistic, Interval and Mixed Uncertainties,” *AIAA Journal*, 2017.
- [51] Eftang, J. L., Huynh, D. B. P., Knezevic, D. J., and Patera, A. T., “A Two-Step Certified Reduced Basis Method,” *Journal of Scientific Computing*, Vol. 51, No. 1, 2012, pp. 28–58.
- [52] Rozza, G., Huynh, D. B. P., and Patera, A. T., “Reduced Basis Approximation and A Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations: Application to Transport and Continuum Mechanics,” *Archives of Computational Methods in Engineering*, Vol. 15, No. 3, 2008, pp. 229–275.