Generalized information reuse for optimization under uncertainty with non-sample average estimators

Laurence W. Cook1 | Jerome P. Jarrett1 | Karen E. Willcox2

1Department of Engineering, University of Cambridge, Cambridge, UK
2MIT Aeronautics and Astronautics, Cambridge, Massachusetts

Correspondence
Laurence W. Cook, University of Cambridge Department of Engineering Trumpington Street, Cambridge CB2 1PZ, UK.
Email: lwc24@cam.ac.uk

Funding information
Engineering and Physical Sciences Research Council (EPSRC), Grant/Award Number: EP/L504920/1; Air Force Office of Scientific Research (AFOSR), Grant/Award Number: FA9550-15-1-0038

Summary
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. Multifidelity techniques can significantly reduce the computational cost of Monte Carlo sampling methods for quantifying the effect of uncertain inputs, but existing multifidelity 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 multifidelity method that treats previous optimization iterations as lower fidelity models. This work generalizes information reuse to be applicable to quantities whose estimators are not sample averages. 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 with regular Monte Carlo sampling.

KEYWORDS
multifidelity, optimization, optimization under uncertainty, probabilistic methods, stochastic problems

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 multifidelity formulations have been developed to address this issue.7-14 Multifidelity 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 While 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.

Multifidelity 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 multifidelity 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” (IR) in the work of Ng and Willcox.23 However, this control variate-based approach, as presented in the work of the aforementioned authors,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 (eg, estimators of the first two statistical moments, ie, 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 multifidelity approaches for OUU cannot be used to accelerate such estimators. This paper extends the IR approach of the work Ng and Willcox23 to estimators that cannot be expressed as a sample average. Consequently, IR becomes applicable to a broader range of OUU formulations. Our “generalized information reuse” (gIR) approach makes use of the ideas behind bootstrapping28 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 gIR 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 this paper.

2 | FORMULATIONS OF OUU

This section discusses a general formulation of the OUU problem and the use of MC sampling 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 \( y(x, u) \), which is a function of controllable design variables \( x \) and input parameters \( u \). We treat \( u \) as uncertain, and use a probabilistic representation such that \( u \) is a realization of a random vector \( U(\omega) \), where \( \omega \) denotes the underlying random event with sample space \( \Omega \) such that \( \omega \in \Omega \). Thus, at a given design \( x \), an output is a random variable \( Y_x(\omega) \) defined such that \( Y_x(\omega) = y(x, U(\omega)) \).
An OUU requires an objective, denoted $d_o(x)$, and constraints, denoted $d_j(x)$, $j = 1, \ldots, n_c$, to be defined that characterize the behavior of $Y_x$ as a function of the design $x$. 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

$$\begin{align*}
\text{minimize} & \quad d_o(x) \\
\text{s.t.} & \quad d_j(x) \leq 0 \quad j = 1, \ldots, n_c, \quad (1)
\end{align*}$$

where each of $d_j(x)$, $j = 0, 1, \ldots, n_c$ 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 $d(x)$ that could be any of $d_j(x)$, $j = 0, 1, \ldots, n_c$.

### 2.2 Monte Carlo sampling

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

Monte Carlo sampling is a nonintrusive approach, which uses an estimator, denoted by $\hat{d}$, to obtain an estimate of $d(x)$.\(^{20,21}\) 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 $d_j(x)$, $j = 0, 1, \ldots, n_c$. Many MC estimators are sample averages in that they are a function that can be expressed in the form

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

where $Z^i$, $i = 1, 2, \ldots, n$ 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:

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

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

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

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 $d(x)$ that can be used in OUU for engineering design and give their MC 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, $\mu = \text{E}[Y_x]$. The variance, $\sigma^2 = \text{Var}[Y_x]$, and standard deviation, $\sigma$, 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 multiobjective 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, eg, requiring that the mean value is $\lambda$ standard deviations away from failure. In this case, the constraint in the OUU 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 IR 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 $Y'_{x}$ are $n$ i.i.d. random variables with the distribution of $Y_{x}$. Thus, we can set $Z' = Y'_{x}$ to express this estimator in the form of Equation (2). The estimator of the variance of $Y_{x}$ is given by

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \frac{n}{n-1} (Y'_{x} - \hat{\mu})^2,$$

and we can set $Z' = \frac{n}{n-1} (Y'_{x} - \hat{\mu})^2$ 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 OUU uses the probability of failure $p = \text{Prob}(Y_{x} \geq 0)$, where, here, failure is indicated by $Y_{x} > 0$. This failure probability is usually used as a constraint in OUU formulations such that $p - (1 - \beta) \leq 0$, where $\beta$ 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 $Z' = 1_{[0,\infty)}(Y'_{x})$ in Equation (2)

$$\hat{p} = \frac{1}{n} \sum_{i=1}^{n} 1_{[0,\infty)}(Y'_{x}),$$

where $1_{S}(.)$ 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)

$$q_{\beta} = {F}_{Y_{x}}^{-1}(\beta) \leq 0,$$

where ${F}_{Y_{x}}^{-1}(\beta)$ is the quantile function of $Y_{x}$, evaluated at $\beta$. The quantile has previously seen much use in OUU 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:

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

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

$$d_{hm} = \left( \int_{0}^{1} (h - t(h))^2 \, dh \right)^{1/2},$$

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 the work of Cook and Jarrett,44 the horsetail matching metric was 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:

$$\hat{d}_{hm} = \left( \frac{1}{n} \sum_{k=1}^{n} (Y_{x}^{(k)} - t(k/n))^2 \right)^{1/2},$$
where $Y_x^{(k)}$ is the kth-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 $x_A$, an estimator of $d(x_A)$ 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 $u^i, i = 1, 2, \ldots, n$ from the distribution of $U(\omega)$; then, evaluate the system model at each of these values to obtain n i.i.d. sampled values of $Y_A$, which is defined to be equal to $y(x_A, U(\omega))$, denoted $y^i_A = y(x_A, u^i), i = 1, 2, \ldots, n$. Each of these values $y^i_A, i = 1, 2, \ldots, n$ 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 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 to achieve this acceptable error. In this section, we present and analyze the gIR estimator that can reduce this computational cost.

#### 3.1 | The IR estimator

Information reuse is a variance reduction method for obtaining $\hat{s}_A$, which is based on the control variate approach. The control variate approach uses an auxiliary random variable to make a correction to a regular MC estimator. For IR, 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 $x_C$, and let $Y_C(\omega) = y(x_C, U(\omega))$ be the auxiliary random variable.

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

$$\hat{s}_A = \hat{d}_A + \gamma (\hat{d}_C - \hat{d}_C), \tag{13}$$

where $\hat{d}_C$ is the true value of $d(x_C)$, and $\gamma$ is the correction factor.

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

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

The sampled values $u^i, i = 1, \ldots, n$ 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

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

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

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

and using this value of $\gamma$ gives the following variance of the IR estimator:

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

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 with 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 IR takes advantage. Indeed, in the work of Ng and Willcox,\(^{23}\) it was shown that, in one dimension, the correlation between \( y(x, U(\omega)) \) and \( y(x + \Delta x, U(\omega)) \) is quadratic in \( \Delta x \) for \( |\Delta x| < 1 \) and approaches 1 as \( \Delta x \) approaches 0.

To implement IR, estimates for the variance and covariance terms in Equation (17) must be obtained. It was illustrated in the work of Ng and Willcox\(^{23}\) how the variance reduction effect of IR 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 \( y_A \) and \( y_C \), \( i = 1, 2, \ldots, n \) and still obtain significant reduction in the variance of \( \hat{s}_A \). Such estimates in the original formulation of IR\(^{23,39}\) 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 \( \hat{\beta} \).

Our generalization of IR can be applied to a general \( d \), whose MC estimator cannot necessarily be expressed as a sample average. Examples of such estimators include \( \hat{d}_b \) and \( \hat{d}_c \). Our approach uses the idea of resampling 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 \( u_i, i = 1, \ldots, n \), from the distribution of \( U(\omega) \), consider the empirical distribution consisting of a probability density function made up of delta functions at each sampled value \( u_i \). This empirical distribution approximates the true distribution of \( U(\omega) \). Thus, resampling with replacement from this empirical distribution, by randomly drawing an index from \( i = 1, \ldots, n \) and using \( u_i \) as the resulting sample realization, obtains sampled values that are from an approximation to the distribution of \( U(\omega) \). The idea behind bootstrapping is to use this resampling 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), first, a set of \( n \) i.i.d. sampled values \( u_i, i = 1, \ldots, n \), are drawn from the distribution of \( U(\omega) \), and the system model at both \( x_A \) and \( x_C \) is evaluated to obtain \( n \) sampled values of both \( Y_A(\omega) \) and \( Y_C(\omega) \), denoted \( y_A = y(x_A, u_i) \) and \( y_C = y(x_C, u_i) \), respectively, for \( i = 1, \ldots, n \).

Next, resampling with replacement occurs \( n \) times from \( u_i, i = 1, \ldots, n \) to obtain a new set of values \( u_i, j = 1, \ldots, n \), and the corresponding values of \( y(x_A, u_i) \) and \( y(x_C, u_i) \) (which have already been evaluated) are used as new sets of sampled values of \( Y_A(\omega) \) and \( Y_C(\omega) \), respectively. This resampling is repeated \( N_{\text{boot}} \) times, and using each set of the resampled values \( y(x_A, u_i) \) and \( y(x_C, u_i) \), \( j = 1, 2, \ldots, n \), a new MC estimator is evaluated to obtain \( \hat{d}_A \) and \( \hat{d}_C \), \( b = 1, \ldots, N_{\text{boot}} \). These are sampled values from approximations to the true distributions of \( d_A \) and \( d_C \).

The bootstrapped estimate of the variance of an MC estimator \( \hat{d} \), denoted \( \hat{V}[\hat{d}] \), is obtained from the sample variance of these \( N_{\text{boot}} \) values \( \hat{d}_b \), \( b = 1, \ldots, N_{\text{boot}} \). Similarly, since the resampled values of \( Y_A \) and \( Y_C \) are the system model evaluated at the same values of \( u \), the bootstrapped estimate of covariance between the estimators at design points \( x_A \) and \( x_C \), denoted \( \Sigma \), can be obtained from the sample covariance

\[
E[\hat{d}] = \frac{1}{N_{\text{boot}}} \sum_{b=1}^{N_{\text{boot}}} \hat{d}_b
\]

\[
\hat{V}[\hat{d}] = \frac{1}{N_{\text{boot}}} \sum_{b=1}^{N_{\text{boot}}} \frac{N_{\text{boot}}}{N_{\text{boot}} - 1} \left( \hat{d}_b - E[\hat{d}] \right)^2
\]

\[
\Sigma[\hat{d}_A, \hat{d}_C] = \frac{1}{N_{\text{boot}} - 1} \sum_{b=1}^{N_{\text{boot}}} \left( \hat{d}_b - E[\hat{d}_A] \right) \left( \hat{d}_b - E[\hat{d}_C] \right).
\]
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.

Note that the covariance $\Sigma[\hat{d}_A, \hat{d}_C]$ is different to the covariance between $y(x_A, U(\omega))$ and $y(x_C, U(\omega))$, which could be estimated analytically from the sampled values $y_i^A$ and $y_i^C, i = 1, \ldots, n$. 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 $x_C$. These estimated values of $\bar{V}[\hat{d}_A], \bar{V}[\hat{d}_C], \text{and } \Sigma[\hat{d}_A, \hat{d}_C]$ are substituted into Equation (16) to obtain an estimated optimal $\gamma$, which we denote as $\bar{\gamma}^*$. This $\bar{\gamma}^*$ is subsequently used to obtain an estimated variance $\bar{V}[\hat{\delta}_A]$ given by Equation (17).

**Algorithm 1** Bootstrapping for variance and covariance of an MC estimator given $n$ evaluations of $y$ at design point $x_A$, and at the control point $x_C$, using $N_{boot}$ sets of resampled values.

1: for $b = 1, \ldots, N_{boot}$ do
2: for $j = 1, \ldots, n$ do
3: $k$ ← random draw from $\{1, \ldots, n\}$
4: $y^j_A$ ← $y(x_A, U^{(k)})$ taken from sampled values $y_i^A, i = 1, \ldots, n$
5: $y^j_C$ ← $y(x_C, U^{(k)})$ taken from sampled values $y_i^C, i = 1, \ldots, n$
6: $\hat{d}^b_A$ ← MC estimator of $d_A$ from resampled values $y^j_A, j = 1, \ldots, n$
7: $\hat{d}^b_C$ ← MC estimator of $d_C$ from resampled values $y^j_C, j = 1, \ldots, n$
8: $\bar{E}[\hat{d}_A], \bar{E}[\hat{d}_C] \leftarrow \frac{1}{N_{boot}} \sum_{b=1}^{N_{boot}} \frac{1}{N_{boot}} \sum_{b=1}^{N_{boot}} \hat{d}^b_A$
9: $\bar{V}[\hat{d}_A], \bar{V}[\hat{d}_C] \leftarrow \frac{1}{N_{boot} - 1} \sum_{b=1}^{N_{boot}} (\hat{d}^b_A - \bar{E}[\hat{d}_A])^2, \frac{1}{N_{boot} - 1} \sum_{b=1}^{N_{boot}} (\hat{d}^b_C - \bar{E}[\hat{d}_C])^2$
10: $\Sigma[\hat{d}_A, \hat{d}_C] \leftarrow \frac{1}{N_{boot} - 1} \sum_{b=1}^{N_{boot}} (\hat{d}^b_A - \bar{E}[\hat{d}_A])(\hat{d}^b_C - \bar{E}[\hat{d}_C])$

To use this estimation procedure within an optimization loop, a required acceptable variance for the estimator $\hat{\delta}_A$ is specified (which for unbiased estimators corresponds to an acceptable MSE) and $y(x_A, u)$ and $y(x_C, u)$ 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 MC
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, \( \tilde{V} = \bar{v}(n) = \bar{V}_0 \times (n/n_0)^{-1} \), 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 \( v_A \), \( v_C \), 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 \( V_{\text{req}} \), gives a function in \( n \) whose roots give the predicted value of \( n \), ie, the number of sampled values of \( Y_A(\omega) \) and \( Y_C(\omega) \) required to reach the acceptable variance value

\[
\phi(n) = v_A(n) - \frac{(v_{A,C}(n))^2}{\bar{V}(\hat{s}_C) + v_C(n)} - V_{\text{req}}.
\]

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 \( n_{\text{init}} \) evaluations of \( y \) at both \( x_A \) and \( x_C \), and thus \( n_{\text{init}} \) 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. \( v_A(n) \leftarrow \tilde{V}[\hat{d}_A]_0 \times n_0 / n \)
2. \( v_C(n) \leftarrow \tilde{V}[\hat{d}_C]_0 \times n_0 / n \)
3. \( v_{A,C}(n) \leftarrow \tilde{V}[\hat{d}_A, \hat{d}_C]_0 \times n_0 / n \)
4. \( \phi(n) \leftarrow \text{predicted variance of } \hat{s}_A \text{ from Equation (19)} \)
5. \( n_{\text{pred}} \leftarrow \max(\inf(n \mid \phi(n) = 0), 0) \)

For the regular MC estimator, this is applied to a single term, ie, \( n_{\text{pred}, \text{MC}} = n_0 \times V[\hat{d}_A]_0 / V_{\text{req}} \). Once it has been decided whether to revert back to regular MC or continue with IR at a given design point, \( y(x_A, u) \) and \( y(x_C, u) \) are evaluated at additional sampled values of \( U(\omega) \) until \( V[\hat{s}_A] \) 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 \( N_{\text{boot}} \) 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 nonnegligible. 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 nonissue; 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, \( n_{\text{init}} \), are used, the prediction given by Algorithm 2 is likely to be inaccurate. Therefore, to avoid overestimating the value of \( n \) needed to give an acceptable variance, the number of sampled values of \( U \) drawn at a time, at which the system model is evaluated, is limited to be a constant \( \alpha \) 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 $(1 + \delta)n_{\text{pred}}$, 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 $\alpha$ can be used.

### 3.4 Bias

If the estimator $\hat{d}$ is unbiased, then the IR estimator is unbiased.\(^{23}\) However, for a general quantity $d$ with MC estimator $\hat{d}$, the bias of the IR estimator is given by

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

(20)

where $B[.]$ 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, IR does not affect the bias of the estimator.

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

### 3.5 Variance of bootstrapped estimators

The bootstrapped estimates $\hat{V}$ and $\hat{\Sigma}$ are random variables that depend on the resampled values of $\hat{d}^b$. If $N_{\text{boot}}$ 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

$$\text{Var}[\hat{V}[\hat{s}_A]] = \text{Var}[\hat{V}_A] + \gamma^4 \text{Var}[\hat{V}_C] + \gamma^4 \text{Var}[\hat{V}_C] + 4\gamma^2 \text{Var}[\hat{\Sigma}] + 2\gamma^2 \text{Cov}[\hat{V}_A, \hat{V}_C] - 4\gamma \text{Cov}[\hat{V}_A, \hat{\Sigma}] - 4\gamma^3 \text{Cov}[\hat{V}_C, \hat{\Sigma}],$$

(21)

where, for clarity, $\text{Var}[\hat{d}_A]$ is denoted by $\text{Var}_A$, $\text{Var}[\hat{d}_C]$ by $\text{Var}_C$, $\text{Var}[\hat{s}_C]$ by $\text{Var}_S$, and $\text{Var}[\hat{d}_A, \hat{d}_C]$ by $\text{Cov}$.

Since $\hat{V}$ and $\hat{\Sigma}$ are sample average estimators, their variance is approximated by $\frac{1}{N_{\text{boot}}} \hat{\sigma}^2$, where $\hat{\sigma}^2$ is the sample variance of the values being averaged. With a reasonable value of $N_{\text{boot}}$, this variance will be small; however, if $\gamma$ is large, then the $\gamma^4 \text{Var}[\hat{V}_C]$ term in Equation (21) is potentially nonnegligible. If the resulting value of $\text{Var}[\hat{V}[\hat{s}_A]]$ is then used as $\text{Var}[\hat{V}_S]$, in a subsequent iteration, it could be magnified further leading to a continuously growing inaccuracy in the estimated variance of the IR estimator and so giving meaningless results.

In IR, the optimal choice for $\gamma$ is rarely greater than 1, and is often $\approx 1$, so this is treated as a nonissue in this paper’s implementation. Furthermore, the main application of this work is for problems where evaluating the system model $y$ is far more expensive than the bootstrapping procedure, and a large $N_{\text{boot}}$ 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, an MC estimator can be resorted back to to break the chain of dependence.

### 3.6 Overall algorithm

The overall algorithm for performing an optimization using the proposed gIR method is given in Algorithm 3.
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, \( u_1, \ldots, u_{n_u} \). There are two outputs

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

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

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 MC 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 OUU. The derivative-free optimizer Constrained Optimization By Linear Approximation (COBYLA)\textsuperscript{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.\textsuperscript{49}

### 4.1.1 Validation

First, in order to validate the gIR method, the variance of the gIR estimator predicted by Equation (15) and Algorithm 1 is compared with an empirically generated variance (both using $N_{\text{boot}} = 5000$). To achieve this, using $n_u = 2$, a sequence of design points are taken from an optimization and fixed; then, the IR 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 gIR 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 (22); 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 gives the following optimization problem:

$$\text{minimize } d_{hm} = \left( \int_0^1 \left( F^{-1}_0(h) - t(h) \right)^2 \, dh \right)^{\frac{1}{2}}$$

subject to

$$F^{-1}_1(0.9) \leq 0,$$

where $F^{-1}_0$ and $F^{-1}_1$ are the inverse CDFs (quantile function) for the random variables given by $y_0(x, U(\omega))$ and $y_1(x, U(\omega))$, respectively. These choices for objective and constraints are propagated as discussed in Sections 2.5 and 2.6, ie, neither of their estimators can be expressed as a sample average. For this problem, we use the risk-averse target $t(h) = -h^4$ in the horsetail matching metric. This target is risk averse because it emphasizes minimizing values of the quantile function $F^{-1}_0(h)$ 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 the works of Cook and Jarrett\textsuperscript{44} and Cook et al\textsuperscript{50} for further discussion on horsetail matching targets.

**FIGURE 2** Comparison, for two non-sample average estimators, of the variance predicted by generalized information reuse and the empirical variance from 200 runs. A, Horsetail matching metric with $t(h) = -h^4$; B, Quantile function $F^{-1}_1(0.9)$ [Colour figure can be viewed at wileyonlinelibrary.com]
We use a required variance of $1 \times 10^{-3}$ for the objective function and $4 \times 10^{-3}$ 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 $x_0 = [-1, -1]$; additionally, $n_{\text{init}} = 50$, $\alpha = 10$, and $\delta = 0.1$.

We run optimizations using only regular MC and using gIR. Figure 3 gives the convergence, in terms of computational cost (in terms of number of evaluations of $y(x, u)$), of the objective function (the horsetail matching metric) and constraint (quantile function) and the computational cost of each optimization iteration. The optimization using IR achieves significant computational savings compared with regular MC 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 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 $i = 1, \ldots, n_u$. This random variable is skewed and has support $(-\infty, 1]$. The results of optimizations with these input uncertainties are given in Figure 4.

On Figure 4, gIR 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 nonlinearity and variance

![Figure 3](image_url)

**FIGURE 3** Optimization progress of generalized information reuse (gIR) and regular Monte Carlo (MC). A, Total number of evaluations; B, Number of evaluations per iteration

![Figure 4](image_url)

**FIGURE 4** Optimization progress of generalized information reuse (gIR) and regular Monte Carlo (MC), using input parameters distributed according to a gamma distribution. A, Total number of evaluations; B, Number of evaluations per iteration
of the system model and the distribution of $U(\omega)$. Since the nonlinear test function given by Equation (22) is being used here, changing the distribution of $U(\omega)$ does not significantly affect the ability of IR to improve upon regular MC.

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 IR 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.

### 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 (therefore, $1 \times 10^{-4}$ for the objective and $4 \times 10^{-4}$ 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, IR is still able to reduce this cost.

Additionally we can see that, compared with Figure 3, more optimization iterations are needed before IR is able to obtain the required variance values using only \( n_{\text{init}} \) sampled values. This is because, for only \( n_{\text{init}} \) sampled values to be used 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 IR to be more expensive than regular MC, i.e., if \( n_{\text{init}} \) is chosen to be too high (relative to the required variances) such that regular MC can achieve the required variance in less than \( n_{\text{init}} \) sampled values, IR wastes the \( n_{\text{init}} \) 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, (therefore, \( 1 \times 10^{-2} \) for the objective and \( 4 \times 10^{-2} \) for the constraint) and a value of \( n_{\text{init}} = 50 \) is used.

For the majority of design points, the \( n_{\text{init}} \) sampled values were enough to achieve the required variance using regular MC, and so IR wasted the additional \( n_{\text{init}} \) system model evaluations at the control point. In this case, this required variance is so high 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 \( n_{\text{init}} \).

If only a very few number of model evaluations can be afforded such that evaluating \( n_{\text{init}} \) 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 two-dimensional acoustic horn governed by the nondimensional 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 \( y(x, u) \), 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 \( x_{k,\text{nom}} \), and then uncertainty is introduced representing manufacturing tolerances such that the actual value \( x_k \) is given by a random variable centered on \( x_{k,\text{nom}} \). 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.

![Figure 7](image-url)  
**Figure 7** Optimization progress of generalized information reuse (gIR) and regular Monte Carlo (MC), using higher required variances.  
(A) Total number of evaluations; B, Number of evaluations per iteration
TABLE 1  Design variables for the acoustic horn design problem

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

TABLE 2  Uncertain inputs in 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 | $z_0$   | Uniform      | 10          | 90          |
| 3   | Upper wall impedance | $z_l$   | Uniform      | 10          | 90          |
| 4   | Geometry            | $x_1$    | Uniform      | $0.975x_{1,\text{nom}}$ | $1.025x_{1,\text{nom}}$ |
| 5   | Geometry            | $x_2$    | Uniform      | $0.975x_{2,\text{nom}}$ | $1.025x_{2,\text{nom}}$ |
| 6   | Geometry            | $x_3$    | Uniform      | $0.975x_{3,\text{nom}}$ | $1.025x_{3,\text{nom}}$ |
| 7   | Geometry            | $x_4$    | Uniform      | $0.975x_{4,\text{nom}}$ | $1.025x_{4,\text{nom}}$ |
| 8   | Geometry            | $x_5$    | Uniform      | $0.975x_{5,\text{nom}}$ | $1.025x_{5,\text{nom}}$ |
| 9   | Geometry            | $x_6$    | Uniform      | $0.975x_{6,\text{nom}}$ | $1.025x_{6,\text{nom}}$ |

FIGURE 8  Magnitude of the complex acoustic pressure field for the horn geometry given by the lower and upper bounds on the design variables. A, Lower bounds $y = 0.0522$; B, Upper bounds $y = 0.1264$

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.

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 IR with regular MC, a global optimizer is used. Here, the evolutionary strategy optimizer from the ecspy python toolbox* 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 IR optimizations over multiple optimization runs.

*https://pypi.python.org/pypi/ecspy
First, for reference, a classical approach to robust optimization is used and a weighted sum of the first two statistical moments of $Y(\omega) = y(x, U(\omega))$ is minimized using regular IR and it is compared with regular MC. Then, the horse-tail matching metric under a risk averse target is minimized using gIR. In both cases, $n_{\text{init}} = 100$, $\delta = 0.1$, and $\alpha = 10$.

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

$$\minimize_{x^L \leq x \leq x^U} \quad d_{\text{ws}} = E[Y(\omega)] + 3\sqrt{V[Y(\omega)]}.$$  \tag{24}$$

This function $d_{\text{ws}}$ 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 $d_{\text{ws}}$ about estimators of $E[Y(\omega)]$ and $V[Y(\omega)]$, both of which can be evaluated using regular IR.\textsuperscript{23,39}

A required variance similar to that used to demonstrate the original IR formulation in the work of Ng and Willcox\textsuperscript{23} is used here, ie, a value of $V_{\text{req}} = 2 \times 10^{-6}$. The optimization is run from 10 random starting populations, such that one optimization that uses regular IR (labeled IR) and one that uses regular 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 and the correlation coefficient between sampled values at the design point and control point for the IR run.

The results on Figure 9 demonstrate how regular IR leads to computational savings compared with regular MC. On average, the IR optimizations achieved the objective function value that the regular MC optimizations took $10^6$ evaluations to obtain in $\approx 2.5 \times 10^5$ 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, gIR is used to optimize the horse-tail matching metric, giving the following optimization problem:

$$\minimize_{x^L \leq x \leq x^U} \quad d_{\text{hm}} = \left( \int_0^1 \left( F^{-1}(h) - t(h) \right)^2 dh \right)^{\frac{1}{2}}.$$  \tag{25}$$

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

Figure 10A plots the convergence for both 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.

**FIGURE 9** Regular information reuse (IR) and regular Monte Carlo (MC) optimizers minimizing a weighted sum of mean and variance of the reflection coefficient. A, Convergence with total evaluations for multiple optimizations (dashed), and their average (solid); B, Number of evaluations required per iteration for a typical optimization.
Comparing Figures 9 and 10, it appears that similar computational gains are achieved by the gIR method with a non-sample average estimator as regular IR 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 $\approx 2.5 \times 10^5$ evaluations.

It can also be observed that, in both cases, there is no significant benefit of IR and gIR over regular MC until after $\approx 100 \times 10^3$ 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 9B and 10B, where for the first $\approx 25$ designs evaluated (corresponding to the population size), the gIR and MC optimization runs required a 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 11** Empirical CDFs of notable designs in the acoustic horn design problem. A, CDFs of $Y_A(\omega)$ for the lower and upper bounds on the design variables, and the CDF for a random design point; B, CDFs of $Y_A(\omega)$ for designs from optimizations of a weighted sum of mean and standard deviation (WS) and the horsetail matching metric (HM). LB, lower bound; UB, upper bound
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 illustrate 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, ie, the CDFs of these two designs do not cross. This highlights part of the goal in developing this gIR, ie, 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 gIR approach optimizing the horsetail matching metric to the regular IR approach optimizing a weighted sum of moments, since they use different objectives and different values for the required variances, the similarity between the advantage of generalized IR over regular MC on Figure 10 and the advantage of regular IR over regular MC on Figure 9 suggests that this goal has been achieved.

5 | CONCLUSIONS

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

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

On an algebraic test problem, the gIR method 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 MC sampling. On a physical acoustic horn design problem, when used in an evolutionary strategy global optimizer, optimizations of the horsetail matching metric using gIR provide similar computational savings over regular MC to optimizations of a weighted sum of mean and standard deviation using regular IR. 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.

ACKNOWLEDGEMENTS

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

ORCID

Laurence W. Cook http://orcid.org/0000-0002-0033-1657

REFERENCES

1. Kennedy MC, O’Hagan A. Bayesian Calibration of computer models. J R Stat Soc Ser B Stat Methodol. 2001;63(3):425-464.
2. Enevoldsen I, Sørensen JD. Reliability-based optimization in structural engineering. Struct Saf. 1994;15(3):169-196.
3. Beyer H-G, Sendhoff B. Robust optimization - a comprehensive survey. Comput Methods Appl Mech Eng. 2007;196(33):3190-3218.
4. Schüeller GI, Jensen HA. Computational methods in optimization considering uncertainties - an overview. Comput Methods Appl Mech Eng. 2008;198(1):2-13.
5. Gabrel V, Murat C, Thiele A. Recent advances in robust optimization: an overview. Eur J Oper Res. 2014;235(3):471-483.
6. Beck AT, Gomes WJS, Lopez RH, Miguel LFF. A comparison between robust and risk-based optimization under uncertainty. Struct Multidiscip Optim. 2015;52(3):479-492.
7. Alexandrov NM, Lewis RM, Gumbert CR, Green LL, Newman PA. Approximation and model management in aerodynamic optimization with variable-fidelity models. J. Airr. 2001;38(6):1093-1101.
8. Allaire D, Wilcoxon K. A mathematical and computational framework for multifidelity design and analysis with computer models. Int J Uncertain Quantif. 2013;4(1):1-20.
9. Eldred MS. Design under uncertainty employing stochastic expansion methods. Int J Uncertain Quantif. 2011;1(2):119-146.
10. Eldred M, Giunta A, Collins S. Second-order corrections for surrogate-based optimization with model hierarchies. Paper presented at: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference; 2004; Albany, NY.
11. Choi S, Alonso JJ, Kroo IM. Two-level multifidelity design optimization studies for supersonic jets. J Airr. 2009;46(2):776-790.
12. Keane AJ. Wing optimization using design of experiment, response surface, and data fusion methods. J Airr. 2003;40(4):741-750.
13. Balabanov V, Haftka R, Grossman B, Mason WH, Watson LT. Multifidelity response surface model for HSCT wing bending material weight. Paper presented at: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization; 1998; St. Louis, MO.
14. Balabanov V, Venter G. Multi-fidelity optimization with high-fidelity analysis and low-fidelity gradients. Paper presented at: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference; 2004; Albany, NY.
15. Lee SH, Chen W. A comparative study of uncertainty propagation methods for black-box-type problems. Struct Multidiscip Optim. 2009;37(3):239-253.
16. Kim NK, Wang H, Queipo NV. Efficient shape optimization under uncertainty using polynomial chaos expansions and local sensitivities. AIAA J. 2006;44(5):1112-1116.
17. Zhang Y, Hosder S. Robust design optimization under mixed uncertainties with stochastic expansions. J Mech Des. 2013;135:081005.
18. Rumpfkeil MP. Robust design under mixed aleatory/epistemic uncertainties using gradients and surrogates. J Uncertain Anal Appl. 2013;1(1).
19. Zhang J, Tafanidis AA, Medina JC. Sequential approximate optimization for design under uncertainty problems utilizing Kriging metamodeling in augmented input space. Comput Methods Appl Mech Eng. 2017;315:369-395.
20. Lemieux C. Monte Carlo and Quasi-Monte Carlo Sampling. New York, NY: Springer Science+Business Media; 2009.
21. Smith RC. Uncertainty Quantification: Theory, Implementation, and Applications. Philadelphia, PA: SIAM; 2013.
22. Peherstorfer B, Willcox K, Gunzburger M. Optimal model management for multifidelity Monte Carlo estimation. SIAM J Sci Comput. 2016;38(5):A3163-A3194.
23. Ng LWT, Wilcoxon KE. Robustness approaches for optimization under uncertainty. Int J Numer Methods Eng. 2014;100(10):746-772.
24. Göhler SM, Eifler T, Howard TJ. Robustness metrics: consolidating the multiple approaches to quantify robustness. J Mech Des. 2016;138(11):111407.
25. Seshadri P, Constantine P, Iaccarino G, Parks G. A density-matching approach for optimization under uncertainty. Comput Methods Appl Mech Eng. 2016;305:562-578.
26. Quagliarella D, Petrone G, Iaccarino G. Optimization under uncertainty using the generalized inverse distribution function. In: Fitzgibbon W, Kuznetsov VA, Neittaanmäki P, Pironneau O, eds. Modeling, Simulation and Optimization for Science and Technology. Dordrecht, The Netherlands: Springer Science+Business Media Dordrecht; 2014:171-190.
27. Quagliarella D, Iuliano E. Robust design of a supersonic natural laminar flow wing-body. IEEE Comput Intell Mag. 2017;12(4):14-27.
28. Efron B. Bootstrap methods: another look at the jackknife. Ann Statist. 1979;7(1):1-26.
29. Mulvey JM, Vanderbei RJ, Zenios SA. Robust optimization of large-scale systems. Oper Res. 1995;43(2):264-281.
30. Ghisu T, Jarrett JP, Parks GT. Robust design optimization of airfoils with respect to ice accretion. J Airr. 2011;48(1):287-304.
31. Lee DS, Periaux J, Onate E, Gonzalez LF, Qin N. Active transonic aerofoil design optimization using robust multiobjective evolutionary algorithms. J Airr. 2011;48(3):1084-1094.
32. Keane AJ. Comparison of several optimization strategies for robust turbine blade design. J Propuls Power. 2009;25(5):1092-1099.
33. Dodson M, Parks GT. Robust aerodynamic design optimization using polynomial chaos. J Airr. 2009;46(2):635-646.
34. Padulo M, Campobasso MS, Guenov MD. Novel uncertainty propagation method for robust aerodynamic design. AIAA J. 2011;49(3):530-543.
35. Lee SW, Kwon OJ. Robust airfoil shape optimization using design for six sigma. J Airr. 2006;43(3):843-846.
36. Parkinson A, Sorensen C, Pourhassan N. A general approach for robust optimal design. J Mech Des. 1993;115(1):74-80.
37. Park G-J, Lee T-H, Lee KH, Hwang K-H. Robust design: an overview. AIAA J. 2006;44(1):181-191.
38. Paiva RM, Crawford C, Suleman A. Robust and reliability-based design optimization framework for wing design. AIAA J. 2014;52(4):711-724.
39. Ng LWT, Wilcoxon KE. Monte Carlo information-reuse approach to aircraft conceptual design optimization under uncertainty. J Airr. 2016;53(2):427-438.
40. Shapiro A, Dentcheva D, Ruszczynski A. Lectures on Stochastic Programming: Modeling and Theory. Philadelphia, PA: SIAM; 2009.
41. Pflug GC. Some Remarks on the value-at-risk and the conditional value-at-risk. In: Uryasev SP, ed. Probabilistic Constrained Optimization: Methodology and Applications. Dordrecht, The Netherlands: Kluwer Academic Publishers; 2000:272-281.
42. Alexander S, Coleman TF, Li Y. Minimizing CVaR and VaR for a portfolio of derivatives. J Bank Finance. 2006;30(2):583-605.
43. David CE, Steinberg SM. Quantile estimation. In: Kotsz N, Balakrishnan N, Read CB, Vidakovic B, eds. Encyclopedia of Statistical Sciences New York, NY: John Wiley & Sons; 1986.
44. Cook LW, Jarrett JP. Horsetail matching: a flexible approach to optimization under uncertainty. *Eng Optim*. 2018;50(4):549-567.
45. Nelson BL. On control variate estimators. *Comput Oper Res*. 1987;14(3):219-225.
46. Davison AC, Hinkley DV. *Bootstrap Methods and Their Applications*. Cambridge, UK: Cambridge University Press; 1999.
47. Okolewski A, Rychlik T. Sharp distribution-free bounds on the bias in estimating quantiles via order statistics. *Stat Probab Lett*. 2001;52(2):207-213.
48. Powell MJD. A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Gomez S, Hennart J-P, eds. *Advances in Optimization and Numerical Analysis*. Kluwer Academic Publishers: Dordrecht, The Netherlands; 1994:51-67.
49. Johnson SG. The NLopt nonlinear-optimization package. 2014. http://ab-initio.mit.edu/nlopt
50. Cook LW, Jarrett JP, Willcox KE. Extending horsetail matching for optimization under probabilistic, interval, and mixed uncertainties. *AIAA J*. 2018;56(2):849-861.
51. Eftang JL, Huynh DBP, Knezevic DJ, Patera AT. A two-step certified reduced basis method. *J Sci Comput*. 2012;51(1):28-58.
52. Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch Comput Methods Eng*. 2008;15(3):229-275.

**How to cite this article:** Cook LW, Jarrett JP, Willcox KE. Generalized information reuse for optimization under uncertainty with non-sample average estimators. *Int J Numer Methods Eng*. 2018;115:1457–1476. [https://doi.org/10.1002/nme.5904](https://doi.org/10.1002/nme.5904)