

Expected improvement based infill sampling for global robust optimization of constrained problems

Samee ur Rehman¹ · Matthijs Langelaar¹

Received: 8 November 2015 / Revised: 26 August 2016 / Accepted: 29 November 2016 /
Published online: 24 January 2017
© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract A novel adaptive sampling scheme for efficient global robust optimization of constrained problems is proposed. The method addresses expensive to simulate black-box constrained problems affected by uncertainties for which only the bounds are known, while the probability distribution is not available. An iterative strategy for global robust optimization that adaptively samples the Kriging metamodel of the computationally expensive problem is proposed. The presented approach is tested on several benchmark problems and the average performance based on 100 runs is evaluated. The applicability of the method to engineering problems is also illustrated by applying robust optimization on an integrated photonic device affected by manufacturing uncertainties. The numerical results show consistent convergence to the global robust optimum using a limited number of expensive simulations.

Keywords Robust optimization · Kriging · Expected improvement · Efficient global optimization

1 Introduction

Robustness against uncertainties in a problem is steadily rising as an important aspect in the field of optimization. Uncertainties can be dealt with in different ways depending on the level of information available about the problem. If the probability distributions of all uncertainties are known then the problem is usually addressed with the so called Stochastic Programming approach (Birge and Louveaux 1997).

✉ Matthijs Langelaar
M.Langelaar@tudelft.nl

¹ Precision and Microsystems Engineering, Delft University of Technology, Delft, The Netherlands

However, often full probabilistic data is not available, but bounds on the uncertainties can be given. Such uncertainties are denoted as bounded-but-unknown (Gurav et al. 2005; Elishakoff and Ohsaki 2010). Robust optimization can tackle problems of this nature. Robust optimization involves the search for the best worst-case cost, i.e. the minimization of the maximum realizable value of the objective with respect to the uncertainty set, subject to the non-violation of the worst-case constraints. Robust optimization of expensive simulation-based problems is especially challenging since the nested optimization process, when applied to expensive problems, can potentially be very inefficient.

In this work, we present a novel approach for efficient global robust optimization of expensive simulation-based constrained problems affected by bounded-but-unknown uncertainties. The method operates on a surrogate model of the expensive problem which is iteratively improved based on novel infill sampling criteria adapted for constrained robust optimization. The present algorithm enables efficient and accurate determination of the global robust optimum of constrained problems.

Abundant research has been performed in recent decades on robust optimization of problems affected by uncertainties. However, much of this work has been focused on solving convex problems. Considerable progress has therefore been made in robust optimization of linear, convex-quadratic, conic-quadratic and semi-definite problems (Ben-Tal et al. 2009). In contrast, literature related to robust optimization of non-convex problems affected by uncertainties is relatively limited. Bertsimas et al. recently treated robust optimization of non-convex problems with constraints, but the method is aimed at identifying local robust optima only (Bertsimas et al. 2010). Additionally, the approach assumes that gradients are available. In general, however, the availability of gradients cannot always be guaranteed.

A vast number of practical problems affected by uncertainties is non-convex. Furthermore, the objective function of such problems is often not explicitly known and therefore has to be treated as a black-box. Such a scenario is typically observed when the objective is a result of a computer simulation. An additional difficulty often encountered is that the inherent simulation is computationally expensive to perform. This further prevents the application of robust optimization in engineering practice.

Applying optimization directly on expensive computer simulations is prohibitively expensive. This is especially true in the case of robust optimization, since robust optimization involves solving a nested min-max optimization problem where the objective to be optimized is itself a result of an optimization. The problem to be tackled therefore renders itself suitable to a surrogate-based optimization strategy where a cheap model is initially constructed via Design of Experiments (DoE) and thereafter, the model is updated using an infill sampling criterion.

Of the various surrogate-based modeling techniques that exist, for the proposed method we employ Kriging (Sacks et al. 1989). The statistical framework of Kriging provides an estimator of the variance of the Kriging interpolator. Using this potential error, different metrics have been proposed to adaptively sample the design domain such that the deterministic optimum of an unconstrained problem can be found efficiently. The metric of Probability of Improvement (PI) and Expected

Improvement (EI) have been shown to be sound statistical infill sampling criteria. By constructing an initial metamodel using a suitable DoE (Morris and Mitchell 1995) and then employing EI, Jones et al. showed that the deterministic global optimum of an unconstrained problem can be found using relatively few expensive simulations (Jones et al. 1998). Jones et al. used the term Efficient Global Optimization to refer to this method.

In order to apply deterministic optimization on an expensive simulation-based problem with constraints, additional surrogates could be constructed for each constraint. In the Kriging framework, an adaptive sampling scheme for a constrained problem was first explored by Schonlau (Schonlau 1997). Based on the Kriging variance, the method suggests a metric of Probability of Feasibility (PF) for each constraint, analogous to the Probability of Improvement in the objective. The product of probability of feasibility of the constraints and expected improvement of the objective can then be used to suggest new sampling locations. Parr et al. (2012a, b) showed that by employing this approach the deterministic global optimum of a constrained problem can be found using relatively few expensive function evaluations.

In addition to providing a basis for an adaptive sampling scheme, Kriging also has the advantage that it generally exhibits superior performance compared to polynomial models when robustness is taken into account (Stinstra and den Hertog 2008). However, a disadvantage of Kriging is that the correlation matrix it generates is prone to ill-conditioning, which may require stabilization (Jones 2001). Moreover, Kriging is also known to underestimate the variance in its interpolation (den Hertog et al. 2006).

To the best of our knowledge, an infill sampling based approach for surrogate-based global robust optimization of computationally expensive constrained problems affected by *bounded-but-unknown* uncertainties has not been previously explored. Equivalent work has been done for constrained problems affected by probabilistic uncertainties (Zhu et al. 2015; Arendt et al. 2013). These techniques cannot be applied to problems involving bounded-but-unknown uncertainties. Yet, there is a strong need to address constrained problems affected by *bounded-but-unknown* uncertainties since most practical problems are, more often than not, subject to constraints. The considered uncertainties are bounded-but-unknown and can affect the design variables of the problem (implementation error) as well as the parameters that cannot actively be adjusted by the optimizer (parametric uncertainties). For *unconstrained* problems affected by bounded-but-unknown uncertainties, Marzat et al. (2012, 2013) demonstrated algorithms for tackling expensive simulation-based problems. Similarly, for unconstrained expensive problems, Ur Rehman et al. also showcased an approach for estimating the global robust optimum (ur Rehman et al. 2014; ur Rehman and Langelaar 2015).

The primary contribution of this work is to provide a sound criterion for infill sampling of expensive simulation-based *constrained* problems such that a feasible global robust optimum can be found cheaply. We restrict the focus to problems involving inequality constraints, since equality constraints that are affected by uncertainty cannot be satisfied in general.

This work builds on the algorithm for unconstrained robust optimization developed by ur Rehman and Langelaar 2015. The developed approach extends the

original algorithm (ur Rehman and Langelaar 2015) to encompass computationally expensive problems that have single or multiple constraints. The metamodels of the objective and the constraints are built using Kriging. We derive novel infill sampling criteria for *constrained* expected improvement in the control variable space and *constrained* expected deterioration in the uncertainties space. The combination of these sampling criteria enables the global robust optimum of the *constrained* problem to be estimated using relatively few expensive simulations.

This extension is especially relevant to computationally expensive engineering problems affected by uncertainties, since engineering problems, in general, involve constraints. The effectiveness of the proposed algorithm to tackle engineering problems *with constraints* is demonstrated by applying it on an integrated optics device affected by fabrication uncertainties. A feasible robust design, in the presence of two constraints, is found for an optical filter which is affected by bounded-but-unknown geometrical variations due to fabrication uncertainties. It is also shown that the deterministic optimum of the optical filter results in an infeasible design if the worst-case uncertainties are realized. Using the proposed approach, users can generate robust designs that provide 100% manufacturing yield for *expensive constrained* engineering problems affected by uncertainties. While the deterministic engineering design may perform better at the nominal parameter values, the robust design will be superior if the worst-case uncertainties are realized. In addition to the application of the algorithm on the engineering problem, the average convergence and mean performance of the method is tested statistically by applying it 100 times on several numerical problems.

This paper is organized as follows. Robust optimization of problems with constraints is introduced in Sect. 2. Section 3 provides a brief description of Kriging as well as expected improvement and probability of feasibility along with their use in unconstrained and constrained deterministic optimization. We introduce the proposed algorithm for efficient global robust optimization of constrained problems in Sect. 4. Finally, Sect. 5 and 6 contain the results and conclusions, respectively. The method for efficient global optimization introduced in Sect. 3 is an established and well known approach in literature for *constrained deterministic* optimization of expensive problems. The novel contribution of this work is contained in Sect. 4, which provides the description of the algorithm for *constrained robust* optimization of computationally expensive problems, and Sect. 5, which demonstrates the application of the algorithm on engineering and numerical constrained problems affected by uncertainties.

2 Robust optimization of problems with constraints

A deterministic optimization problem subject to constraints may be defined as,

$$\begin{aligned} \min_{\mathbf{x}_c} f(\mathbf{x}_c) \\ \text{s.t. } h_j(\mathbf{x}_c) \leq 0 \quad \forall j, \end{aligned} \quad (1)$$

where $\mathbf{x}_c \in \mathbb{X}_c$ is the set of design variables, $f(\mathbf{x}_c)$ is the objective function and $h_j(\mathbf{x}_c)$ is the set of constraints. If the problem is affected by implementation error

$\Delta \in \mathcal{U}$, with \mathcal{U} as the uncertainty set, then this directly impacts the design variables. In this scenario, the robust optimization problem is given by,

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\Delta} f(\mathbf{x}_c + \Delta) \\ \text{s.t. } \max_{\Delta} h_j(\mathbf{x}_c + \Delta) \leq 0 \quad \forall j. \end{aligned} \tag{2}$$

The above formulation shows that robust optimization involves minimizing the worst-case cost instead of the deterministic cost. Let us now assume that the problem is affected by uncertainties in the problem data only. Examples of problem data for a physical problem can be parameters such as temperature or humidity. These parameters are usually uncontrollable factors that could be subject to change due to uncertainties. These parametric uncertainties can be modeled using a set of environment variables $\mathbf{x}_e \in \mathbb{X}_e$ where \mathbb{X}_e is the uncertainty set. It should be noted here that \mathbb{X}_c and \mathbb{X}_e are compact sets. A robust optimization problem subject to constraints can then be written as,

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\mathbf{x}_e} f(\mathbf{x}_c, \mathbf{x}_e) \\ \text{s.t. } \max_{\mathbf{x}_e} h_j(\mathbf{x}_c, \mathbf{x}_e) \leq 0 \quad \forall j. \end{aligned} \tag{3}$$

Observing the above equation, we note that the *worst-case* constraints with respect to the uncertainty set \mathbb{X}_e should not be violated in order to find a feasible solution. Therefore, the global robust optimum would be the location that provides the best worst-case cost, given that that location does not violate the worst-case constraints.

For some problems, uncertainties could impact both the design variables and the parameters. In this case, the solution has to be robust against parametric uncertainties as well as implementation error. The robust optimization problem, subject to constraints, for this general problem is given by

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\mathbf{x}_c, \Delta} f(\mathbf{x}_c + \Delta, \mathbf{x}_e) \\ \text{s.t. } \max_{\mathbf{x}_c, \Delta} h_j(\mathbf{x}_c + \Delta, \mathbf{x}_e) \leq 0 \quad \forall j. \end{aligned} \tag{4}$$

The objective function and the constraints are considered to be non-convex. Furthermore, we assume that the function and the constraints are based on the response of an expensive computer simulation. Therefore, the ultimate goal of this work is to estimate a feasible global robust optimum of the constrained problem in Eq. (4) using a relatively small number of expensive simulations. This is performed by operating on cheap Kriging models of the objective and constraints instead of on the expensive computer simulation. The problem is expressed as

$$\begin{aligned} \min_{\mathbf{x}_c} \max_{\mathbf{x}_c, \Delta} \mathcal{F}_f(\mathbf{x}_c + \Delta, \mathbf{x}_e) \\ \text{s.t. } \max_{\mathbf{x}_c, \Delta} \mathcal{H}_j(\mathbf{x}_c + \Delta, \mathbf{x}_e) \leq 0 \quad \forall j, \end{aligned} \tag{5}$$

where \mathcal{F}_f is the Kriging model of the objective and $\mathcal{H}_j \quad \forall j$ are the Kriging models of the constraints. In order to estimate the global robust optimum accurately, the

surrogate models need to approximate the corresponding reference functions very well, especially in the neighbourhood of the robust optimum. Extra emphasis needs to be paid to the metamodel error in the constraint models $\mathcal{H}_j \forall j$, since a feasible robust optimum on the metamodel should ideally also be feasible on the true function.

In the following section, we will discuss Kriging and its application to the deterministic optimization of constrained problems. This will provide a basis for the algorithm proposed in Sect. 4, which strives to solve Eq. (4). The scheme uses infill sampling criteria based on Kriging that enable Eq. (5) to approximate Eq. (4) increasingly well in potential regions of interest for global robust optimization of a given problem.

3 Kriging-based deterministic optimization of constrained problems

3.1 Kriging

A very brief description of the metamodeling technique known as Kriging is provided herein. For detailed explanation concerning the model construction and interpolation please refer to Sacks et al. (1989).

Kriging is an interpolation technique that assumes that the function response at any position in the design domain can be described as a normally distributed random variable. Appendix 1 provides further details on Kriging. Kriging employs a tunable Gaussian basis function, Eq. (22)—Appendix 1, to describe the correlation between any two sample points. Optimal values for the tunable parameters of this basis function are found by maximizing the likelihood of obtaining the observed data. Using this tunable basis function, the Kriging prediction, \hat{y} , is estimated by maximizing the combined likelihood of the observed data and the predicted value, Eq. (23). The statistical basis of Kriging provides an estimate of the variance, s^2 , in the Kriging interpolator, Eq. (24).

3.2 Deterministic unconstrained optimization

Rather than working with a predetermined and static set of samples, it proves more efficient to adaptively extend the set of samples to refine the approximation. The combination of the Kriging interpolator and its variance has been successfully used to devise adaptive sampling schemes for efficient optimization of black-box functions. Jones et al. proposed the Efficient Global Optimization (EGO) algorithm (Jones et al. 1998) for deterministic unconstrained optimization based on the Kriging framework. Their method used the adaptive sampling criterion of Expected Improvement (EI).

The EI criterion is developed by assuming that the uncertainty in the predicted value $\hat{y}(\mathbf{x})$ at a position \mathbf{x} can be described in terms of a normally distributed random variable $Y(\mathbf{x})$. The Kriging interpolator $\hat{y}(\mathbf{x})$ is assumed to be the mean of this random variable while the variance is assumed to be given by the Kriging mean

squared error $s^2(\mathbf{x})$. There is a possibility for improving on the current observed minimum, y_{\min} , if a part of the distribution $Y(\mathbf{x})$ lies below y_{\min} . Let this improvement be denoted by I . Finding the expectation of the improvement I , i.e. $E[I(\mathbf{x})] = E[\max(y_{\min} - Y, 0)]$, gives us the expected improvement. A cheaply computable analytical expression may be derived for the EI (Schonlau and Welch 1996),

$$E[I(\mathbf{x})] = (y_{\min} - \hat{y}(\mathbf{x}))\Phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{s(\mathbf{x})}\right) \tag{6}$$

where $\Phi(\cdot)$ is the normal cumulative distribution function and $\phi(\cdot)$ is the normal probability density function. A global optimizer can be used to estimate the global maximizer of EI. The expensive function is evaluated at the maximizer location and the metamodel is rebuilt with the augmented set of samples and responses. By iteratively sampling the metamodel using EI the global deterministic optimum of the problem can be found in relatively few iterations.

3.3 Deterministic constrained optimization

The constraints are also considered to be a result of an expensive computer simulation. Therefore, a cheap model has to be built for them as well. An option could be to include each constraint as a penalty term, but for more complex constraints this approach does not work well (Parr et al. 2012b).

In order to deal with constraints a measure known as Probability of Feasibility (PF) was developed by Schonlau (1997). The criterion is analogous to the probability of improvement for the objective. Let the constraint metamodel be denoted by $\mathcal{H}(\mathbf{x})$ and the Kriging prediction by $\hat{h}(\mathbf{x})$. To derive the expression for probability of feasibility, it is again assumed that the uncertainty in the predicted value $\hat{h}(\mathbf{x})$ at a position \mathbf{x} can be described in terms of a normally distributed random variable $H(\mathbf{x})$ with mean $\hat{h}(\mathbf{x})$ and variance $s_h^2(\mathbf{x})$. The measure gives the area of the distribution $H(\mathbf{x})$ that is below the constraint limit h_{\min} or $P[F(\mathbf{x}) < h_{\min}]$. For a single constraint the probability of feasibility is given by

$$P[F(\mathbf{x}) < h_{\min}] = \Phi\left(\frac{h_{\min} - \hat{h}(\mathbf{x})}{s_h(\mathbf{x})}\right), \tag{7}$$

where $F(\mathbf{x}) = h_{\min} - H(\mathbf{x})$ is a measure of feasibility. Typically, the constraint expression is rearranged so that the constraint limit $h_{\min} = 0$. Just like expected improvement and probability of improvement, the probability of feasibility is an analytical expression that is cheaply computable. The probability of feasibility is basically a metric that gives an indication of possible feasible regions in the domain.

The product of expected improvement in the objective and the probability of feasibility of the constraint can then provide a suitable infill criterion for constrained problems (Parr et al. 2012a, b),

$$EI_F = E[I(\mathbf{x})]P[F(\mathbf{x}) < h_{\min}]. \quad (8)$$

By estimating the global maximizer of the constrained expected improvement, EI_F , a suitable location at which to sample both the objective metamodel and constraint metamodels can be found. The method can readily be extended to multiple constraints by using the total probability of feasibility, which is given by the product of the individual probability of feasibility of each constraint.

Figure 1 shows the flowchart of the algorithm for deterministic optimization of constrained problems using Kriging and adaptive sampling. The algorithm is initialized in Step 1. N_T gives the total number of expensive simulations available to the algorithm, while n is the initial number of samples. ϵ_{PI} and ϵ_{EI} are the minimal probability of improvement and minimal expected improvement thresholds respectively. These quantities determine when the algorithm is terminated due to a lack of sufficient predicted improvement in the optimum value. A suitable Design of Experiments strategy is used to choose the initial sampling regions in Step 2. Once the responses of the objective and the constraints are found, the Kriging models are constructed in Step 4. Thereafter, the global maximizer of EI_F is estimated and this is assigned as the new location to be sampled, \mathbf{x}_{new} . The response of the objective and the constraints at \mathbf{x}_{new} are computed in Step 6. The process of constructing the objective and constraint metamodels and adaptively sampling the domain is repeated until the stopping criterion in Step 7 is reached. A possible

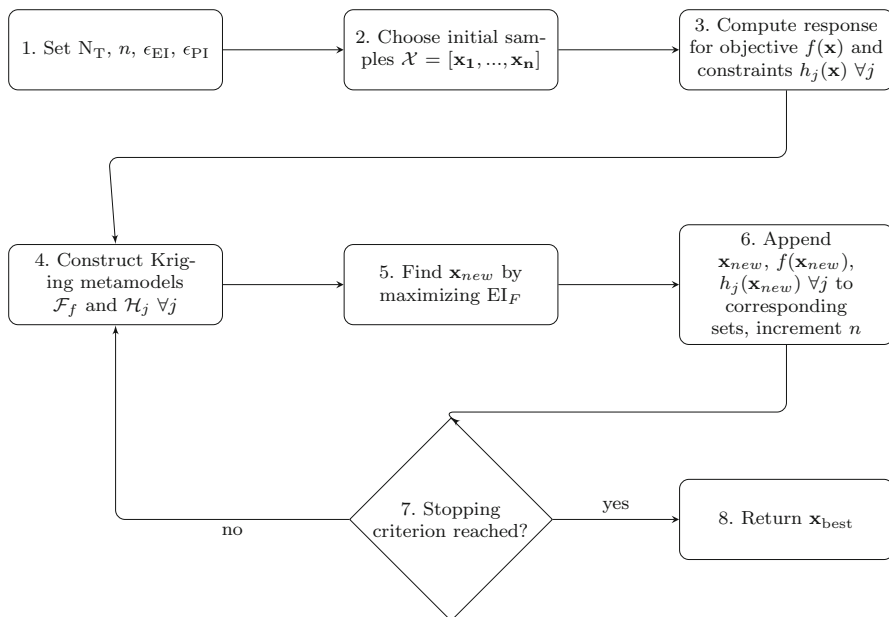


Fig. 1 Flowchart of deterministic optimization using constrained EI. The algorithm finds the deterministic optimum of a constrained problem with relatively few function calls of an expensive to evaluate function

stopping criterion could be the point at which $n = NT$. Alternatively, depending on the sampling criterion used, the algorithm may be stopped when the maximum EI_F falls below ϵ_{EI} . At this stage, the feasible sample corresponding to the minimum objective value is returned as the solution \mathbf{x}_{best} . This algorithm has successfully been demonstrated on deterministic constrained problems by Parr et al. (2012a, b). Parr et al. used EI_F as the infill sampling criterion in their work.

4 Efficient global robust optimization of constrained problems

A scheme for Kriging-based deterministic optimization of expensive simulation-based constrained problems was introduced in the previous section. We now propose an efficient technique, based on Kriging, for the *global robust* optimization of expensive simulation-based constrained problems.

In this section, it is shown how the robust optimum can be found for a problem affected by parametric uncertainties only, Eq. (3). The basic principle of the algorithm does not change even if the problem to be solved is affected by implementation error only, Eq. (2) or is affected by both implementation error and parametric uncertainties, Eq. (4). For clarity, we focus our discussion on an algorithm that solves Eq. (3). The separate treatment of implementation error can result in efficiency improvement. We refer to ur Rehman et al. (2014) for a detailed discussion of this aspect.

Figure 2 illustrates the steps that are involved in estimating the robust optimum. The foundation of the method is the same as the one for deterministic Kriging-based optimization. Both approaches depend on the same initialization phase, i.e. Step 1 to Step 4 are identical except for the fact that ϵ_{PI} is not defined in Fig. 2. It is important to point out here, however, that the initial samples in Step 2 in the flowchart in Fig. 2 are chosen not only in the design variable space \mathbb{X}_c but also in the environment variable space \mathbb{X}_e . This is followed by an iterative surrogate update process where a single new adaptive sample is added in each iteration.

The significant difference between the two methods is the actual process by which this new sample \mathbf{x}_{new} is found at each iteration. In deterministic optimization, the search for \mathbf{x}_{new} simply involved maximizing Eq. (8). However for robust optimization this process has to be broken down into several steps. A reference metric for the robust optimum is first required. This is given by the constrained robust optimum, $r_{\mathcal{K}}$, on the metamodel.

When estimating the robust optimum $r_{\mathcal{K}}$ on the metamodel, the effect of the metamodel error also has to be included. In particular, errors in the constraint surrogate can result in an infeasible solution being chosen as the robust optimum. To mitigate the effect of the metamodel error, Stinstra and den Hertog (2008) suggested a method that makes use of the variance in the Kriging interpolator of the constraints. The strategy basically involved adding the standard deviation of the constraint metamodel to the Kriging prediction of the constraint. This would result in a more conservative constraint metamodel, especially in regions with high

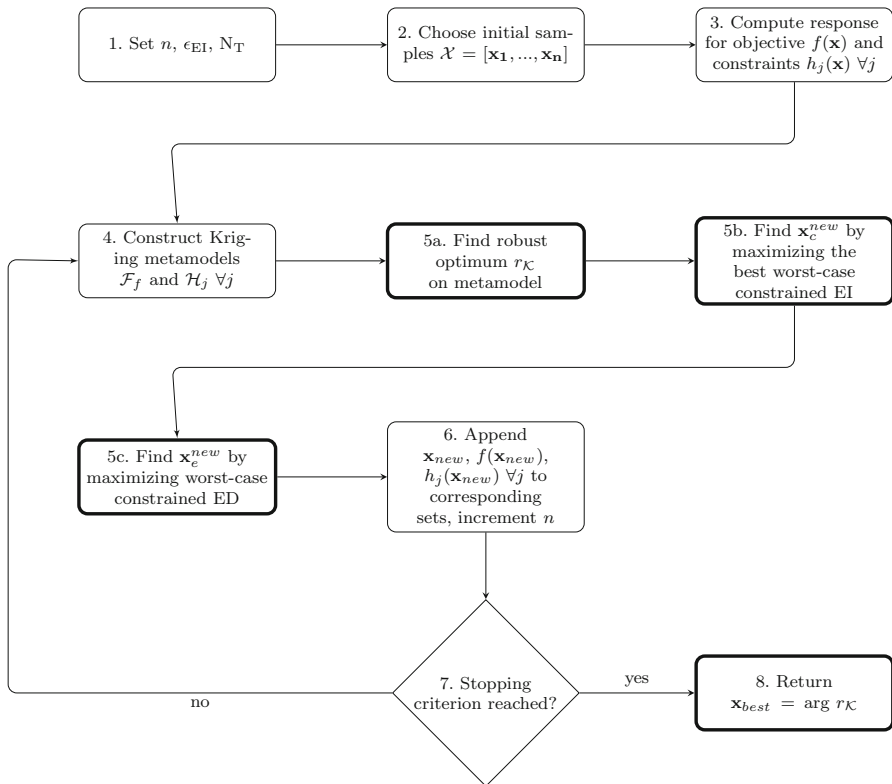


Fig. 2 Flowchart shows the algorithm for efficient global robust optimization of constrained problems. The steps with the bold borders represent the changes that have been made to the algorithm in Fig. 1 in order to reflect the fact that we are searching for a robust optimum

uncertainty and thereby reduce the chance of obtaining an infeasible solution. The robust optimum $r_{\mathcal{K}}$ is therefore found via

$$\begin{aligned}
 r_{\mathcal{K}} &= \min_{\mathbf{x}_c} \max_{\mathbf{x}_e} \mathcal{F}_f(\mathbf{x}_c, \mathbf{x}_e) \\
 \text{s.t. } &\max_{\mathbf{x}_c} \mathcal{H}_j(\mathbf{x}_c, \mathbf{x}_e) + \kappa s_j(\mathbf{x}_c, \mathbf{x}_e) \leq 0 \quad \forall j,
 \end{aligned} \tag{9}$$

where $s_j(\mathbf{x}_c, \mathbf{x}_e)$ is the metamodel standard deviation for the j th constraint metamodel at location $(\mathbf{x}_c, \mathbf{x}_e)$. The parameter κ , chosen between $[0, 1]$ is a measure of how conservative we want to be with respect to the metamodel error. A value of zero for κ means that the metamodel error is not included, while higher values indicate a more conservative approach.

After locating $r_{\mathcal{K}}$, we divide the search into two parts. First, the optimal sampling location in the control variable space, \mathbb{X}_c , is found and then we search for the optimal sampling location in the environment variable space, \mathbb{X}_e . The adaptive sampling measures needed to perform this search are also suitably adapted for estimating regions of interest for locating the robust optimum rather than the deterministic optimum.

4.1 Optimal sampling location in \mathbb{X}_c

The optimal sampling location $\mathbf{x}_c^{new} \in \mathbb{X}_c$ should be the infill location corresponding to the highest expectation of improvement over the current constrained robust optimum $r_{\mathcal{K}}$, Eq. (9). The search for \mathbf{x}_c^{new} is performed in Step 5b of the flowchart in Fig. 2.

To illustrate how this sampling location is found, we make use of an example function of a single control variable x_c and a single environment variable x_e . The problem has one constraint, which is also a function of both x_c and x_e . Figure 3a shows the Kriging metamodel \mathcal{F}_f of the two dimensional function, based on a set of initial samples and responses. The plot also contains the constraint boundary of the constraint metamodel \mathcal{H} . The feasible region is also indicated on the plot. The construction of the Kriging metamodel of the objective and the constraint, Eq. (9), involves going through Step 1 to Step 4 of the flowchart in Fig. 2.

Figure 3b, on the other hand, shows the worst-case Kriging metamodel \hat{y}_{max} ,

$$\hat{y}_{max}(\mathbf{x}_c) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{F}_f(\mathbf{x}_c, \mathbf{x}_e). \tag{10}$$

The maximizer of Eq. (10) is denoted by \mathbf{x}_e^{max} . The region where the worst-case constraint, \hat{h}_{max} , has a predicted response greater than the constraint limit is indicated in pink in Fig. 3b,

$$\hat{h}_{max}(\mathbf{x}_c) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{H}(\mathbf{x}_c, \mathbf{x}_e). \tag{11}$$

The minimum value for \hat{y}_{max} within the feasible region gives the constrained robust optimum $r_{\mathcal{K}}$. The process of estimating the robust optimum corresponds to Step 5a in the flowchart in Fig. 2.

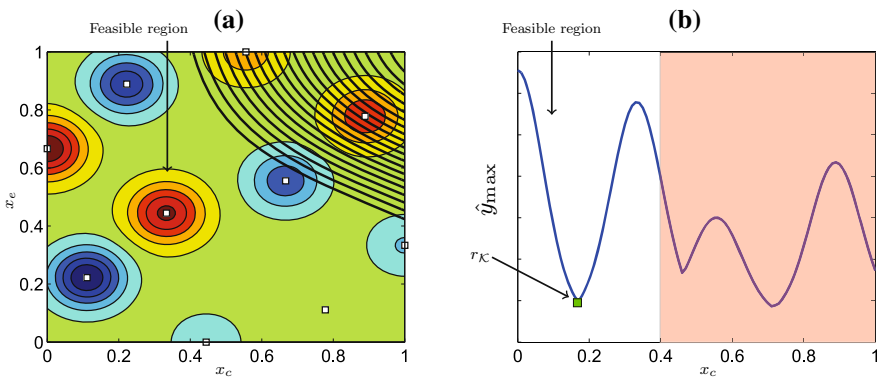


Fig. 3 Plot **a** shows a Kriging metamodel of a two-dimensional function constructed using 10 samples that are marked in white. The constraint boundary of the constraint metamodel is also plotted. The infeasible region is given by the area covered by the black lines on the top right corner. Plot **b** shows the worst-case Kriging metamodel, found via Eq. 10, as well as the worst-case constraint boundary, found via Eq. 11. The infeasible region in (b) is shaded in pink. The robust optimum location, estimated using Eq. 9, is also indicated on the plot. (Color figure online)

A constrained expected improvement criterion is required for identifying a promising location at which to sample in \mathbb{X}_c . Following the method described for deterministic optimization, this would involve obtaining an expected improvement expression for the objective and a probability of feasibility for the constraint.

To formulate the EI in the objective, it is assumed that the uncertainty in the worst-case Kriging prediction \hat{y}_{\max} , at any location $(\mathbf{x}_c, \mathbf{x}_e^{\max})$, can be described in terms of a normally distributed random variable Y_{\max} with mean \hat{y}_{\max} and variance $s^2(\mathbf{x}_c, \mathbf{x}_e^{\max})$. We can improve over the current robust optimum $r_{\mathcal{K}}$ when $Y_{\max} < r_{\mathcal{K}}$. It was shown by ur Rehman and Langelaar (2015) that the expectation of this improvement, I_c , is given by

$$\underbrace{E[I_c(\mathbf{x}_c)]}_{EI_c} = (r_{\mathcal{K}} - \hat{y}_{\max}(\mathbf{x}_c))\Phi\left(\frac{r_{\mathcal{K}} - \hat{y}_{\max}(\mathbf{x}_c)}{s(\mathbf{x}_c, \mathbf{x}_e^{\max}(\mathbf{x}_c))}\right) + s(\mathbf{x}_c, \mathbf{x}_e^{\max}(\mathbf{x}_c))\phi\left(\frac{r_{\mathcal{K}} - \hat{y}_{\max}(\mathbf{x}_c)}{s(\mathbf{x}_c, \mathbf{x}_e^{\max}(\mathbf{x}_c))}\right). \tag{12}$$

The plot in Fig. 4a shows the expected improvement EI_c as a function of the control variable x_c . Intuitively speaking, EI_c represents a balance between exploration and exploitation of the control variable space. Here exploration refers to probing parts of the space where the Kriging error is large. On the other hand, exploitation refers to sampling locations that could be close to the robust optimum. In Eq. (12) the first part of the expression on the right pushes sampling in an exploitative manner while the second portion of the equation pushes samples to be placed in areas that have not yet been searched.

To come up with the probability of feasibility expression, we again make use of a normal distribution to model the uncertainty in the worst-case constraint, \hat{h}_{\max} . Therefore, the uncertainty in the worst-case constraint, \hat{h}_{\max} , at any location $(\mathbf{x}_c, \mathbf{x}_e^{\max})$ is treated in terms of a normally distributed random variable H_{\max} with mean \hat{h}_{\max} and variance $s_h^2(\mathbf{x}_c, \mathbf{x}_e^{\max})$. The probability of feasibility is then given by

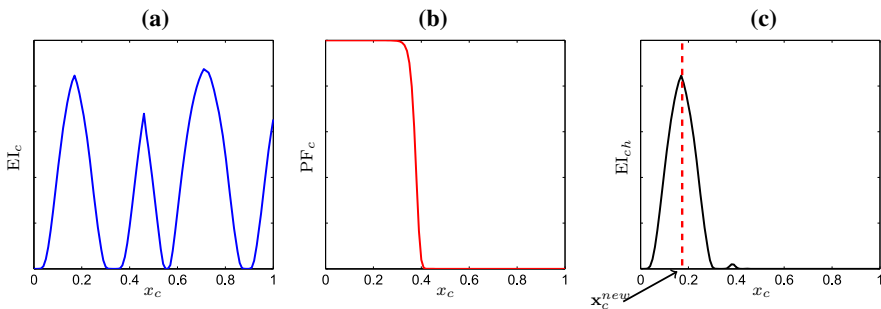


Fig. 4 Plot **a** shows the expected improvement, EI_c , in the objective for the Kriging model from Fig. 3. Since EI_c is only a function of \mathbf{x}_c , Eq. (12), the plot is single dimensional. The Probability of Feasibility of the constraint, which is also only a function of \mathbf{x}_c , Eq. (13), is plotted in **(b)**. Plot **c** shows the product of EI_c and PF_c , which is computed using Eq. (14). The new control variable location \mathbf{x}_c^{new} , given by the maximum value of this function is also indicated

the area of the distribution H_{max} that is below the constraint limit h_{min} . The expression for a single constraint can be written as

$$P[F_c(\mathbf{x}_c)]_{PF_c} = \Phi\left(\frac{h_{min} - \hat{h}_{max}(\mathbf{x}_c)}{s_h(\mathbf{x}_c, \mathbf{x}_e^{max}(\mathbf{x}_c))}\right). \tag{13}$$

The plot in Fig. 4b shows the probability of feasibility PF_c as a function of the control variable x_c when h_{min} is considered to be at the constraint limit.

As in deterministic constrained optimization, a suitable infill criterion in \mathbb{X}_c can be found by maximizing the product of the expected improvement, EI_c , in the objective and the probability of feasibility, PF_c , in the constraint

$$EI_{ch} = E[I_c(\mathbf{x}_c)] P[F_c(\mathbf{x}_c)]. \tag{14}$$

The new sampling location, $\mathbf{x}_c^{new} \in \mathbb{X}_c$ is obtained by determining the global maximizer of Eq. (14). Multiple constraints are handled by replacing the single probability of feasibility in Eq. (14) by the product of the individual probability of feasibility of each constraint. Figure 4c shows a plot of EI_{ch} . The new control variable location \mathbf{x}_c^{new} , given by the location of the global maximum, is also indicated. \mathbf{x}_c^{new} is determined in Step 5b in the flowchart in Fig. 2.

4.2 Optimal sampling location in \mathbb{X}_e

After choosing \mathbf{x}_c^{new} , the algorithm searches for the optimal infill sampling location, \mathbf{x}_e^{new} , in the environment variable space \mathbb{X}_e . Figure 5a shows the same Kriging metamodel of the objective and the constraint boundary of the constraint metamodel along with the feasible region. The location of \mathbf{x}_c^{new} is also shown on the plot. Figure 5b shows the Kriging prediction of the objective at \mathbf{x}_c^{new} , corresponding to the line of plot (a), plotted with respect to \mathbf{x}_e . The worst-case cost g_K is also shown on the plot. The worst-case is given by

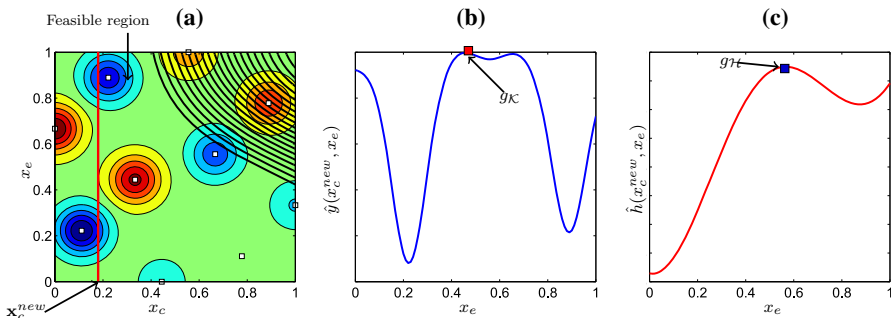


Fig. 5 Plot a shows the same Kriging metamodel along with the constraint boundary of the constraint metamodel and the infeasible region, covered by the *black lines*. The location of \mathbf{x}_c^{new} , based on the maximum value in Fig. 4c, is also shown. The Kriging prediction at \mathbf{x}_c^{new} , corresponding to the response along the *red line*, in plot a, is plotted with respect to \mathbf{x}_e in b. Mathematically this is the response at $\mathcal{F}_f(\mathbf{x}_c^{new}, \mathbf{x}_e)$ and the maximum value, g_K is given by Eq. (15) Plot c shows the Kriging prediction of the constraint at \mathbf{x}_c^{new} , given by $\mathcal{H}(\mathbf{x}_c^{new}, \mathbf{x}_e)$. The maximum value g_{Ht} is given by Eq. (16). (Color figure online)

$$g_{\mathcal{K}}(\mathbf{x}_c^{new}, \mathbf{x}_e) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{F}_f(\mathbf{x}_c^{new}, \mathbf{x}_e). \quad (15)$$

Figure 5c shows the Kriging prediction of the constraint at \mathbf{x}_c^{new} . Again the worst-case constraint value $g_{\mathcal{H}}$ is also shown on the plot. In general, the worst-case constraint value is given by

$$g_{\mathcal{H}}(\mathbf{x}_c^{new}, \mathbf{x}_e) = \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{H}(\mathbf{x}_c^{new}, \mathbf{x}_e). \quad (16)$$

An adaptive sampling criterion is needed in the environment variable space to suggest \mathbf{x}_e^{new} . Choosing \mathbf{x}_e^{new} involves finding a location that could potentially give a higher, i.e. more pessimistic, value than $g_{\mathcal{K}}$ and $g_{\mathcal{H}}$. This is the goal since the aim is to find the most adverse situation in the environment variable space \mathbb{X}_e . An expected *deterioration* (ED) criterion for the objective should therefore help identify a location with the highest expected value relative to $g_{\mathcal{K}}$. Similarly, an ED measure for the constraint should aid in estimating a location with the highest expected constraint value relative to $g_{\mathcal{H}}$.

In an earlier work (ur Rehman and Langelaar, 2015) the authors have derived the expression for the expected deterioration in the objective with respect to the environment variable space. This is given by

$$\underbrace{E[D_e(\mathbf{x}_c^{new}, \mathbf{x}_e)]}_{ED_e} = (\hat{y}(\mathbf{x}_c^{new}, \mathbf{x}_e) - g_{\mathcal{K}}) \Phi \left(\frac{\hat{y}(\mathbf{x}_c^{new}, \mathbf{x}_e) - g_{\mathcal{K}}}{s(\mathbf{x}_c^{new}, \mathbf{x}_e)} \right) + s(\mathbf{x}_c^{new}, \mathbf{x}_e) \phi \left(\frac{\hat{y}(\mathbf{x}_c^{new}, \mathbf{x}_e) - g_{\mathcal{K}}}{s(\mathbf{x}_c^{new}, \mathbf{x}_e)} \right). \quad (17)$$

The plot in Fig. 6a shows the expected deterioration, ED_e , as a function of the environment variable, x_e . Similar to EI_c , the expression for ED_e also enables a balance to be made between exploration of the environment variable space that has been sampled yet and greedily adding point (exploitation) in regions that could potentially give the worst-case cost. The first part of Eq. (17) on the right gives weight to exploitation while the second part biases towards sampling in unexplored regions.

The expected deterioration in the constraint is completely analogous to ED_e in the objective. The expression is given by

$$\underbrace{E[D_h(\mathbf{x}_c^{new}, \mathbf{x}_e)]}_{ED_h} = (\hat{h}(\mathbf{x}_c^{new}, \mathbf{x}_e) - g_{\mathcal{H}}) \Phi \left(\frac{\hat{h}(\mathbf{x}_c^{new}, \mathbf{x}_e) - g_{\mathcal{H}}}{s_h(\mathbf{x}_c^{new}, \mathbf{x}_e)} \right) + s_h(\mathbf{x}_c^{new}, \mathbf{x}_e) \phi \left(\frac{\hat{h}(\mathbf{x}_c^{new}, \mathbf{x}_e) - g_{\mathcal{H}}}{s_h(\mathbf{x}_c^{new}, \mathbf{x}_e)} \right). \quad (18)$$

In the case of multiple constraints, the total expected deterioration in the constraint, ED_h , can be found by taking the product of the individual ED for each constraint. The plot in Fig. 6b shows the expected deterioration ED_h in the constraint as a function of the environment variable x_e .

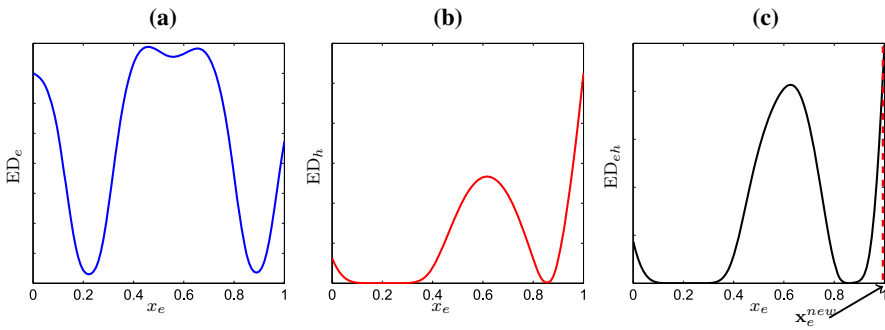


Fig. 6 Plot **a** shows the expected deterioration, ED_e , in the objective with respect to the environment variable for the original Kriging model from Fig. 3a. ED_h is plotted in **(b)** using Eq. (17). Plot **c** shows the constrained expected deterioration ED_{eh} found via Eq. (19). The new environment variable location \mathbf{x}_e^{new} , corresponding to the maximum value in the plot is also indicated

The new sampling location \mathbf{x}_e^{new} can be found by determining the maximizer of the product of the expected deterioration in the objective, ED_e , and the expected deterioration in the constraint, ED_h ,

$$ED_{eh} = E[D_e(\mathbf{x}_c^{new}, \mathbf{x}_e)] E[D_h(\mathbf{x}_c^{new}, \mathbf{x}_e)]. \tag{19}$$

Figure 6c shows a plot of the product ED_{eh} along with the location of the new environment variable location, \mathbf{x}_e^{new} . Step 5c in the flowchart in Fig. 2 involves the search for the new environment variable, \mathbf{x}_e^{new} .

4.3 Implementation aspects

Once \mathbf{x}_{new} is identified, the objective and the constraints are evaluated using the expensive simulation at the new location. Thereafter, if the stopping criterion has not been reached yet, the metamodel is rebuilt and the process of searching for \mathbf{x}_c^{new} and \mathbf{x}_e^{new} is repeated. The algorithm is stopped when the total number of function evaluations available NT is exhausted. Additionally, the algorithm can also be terminated if the robust optimum, $r_{\mathcal{K}}$, found over the last few iterations, does not change significantly. For this purpose, we maintain a history set, \mathcal{S}_h , that consists of the robust optimum found at each iteration.

Apart from its use as a termination criterion, the history set can aid in the search for the robust optimum. Whenever a new search for the robust optimum $r_{\mathcal{K}}$ is initiated at a particular iteration, the starting points for the search can include the history set, \mathcal{S}_h , of the robust optima locations found in the previous iterations. In this manner, it is ensured that a possible robust optimum location found in previous iterations is not missed by the global search in the current iteration. By doing so, we also systematically reduce the estimation error of the internal global optimizers.

Once the algorithm terminates, the location of the robust optimum, $r_{\mathcal{K}}$, found at the last iteration is returned as the final solution. The result from the final iteration is chosen instead of any other iteration since the last iteration includes the most information about the problem.

5 Results

5.1 Testing methodology

In order to test the ability of the algorithm to reliably and efficiently converge to the global robust optimum, its numerical performance is evaluated on five analytical test problems and one engineering case study. The test problems are provided in Appendix 2. The objective functions for the first four problems are well known benchmark problems which were originally employed as test problems for unconstrained min-max optimization by Rustem and Howe (2002). The corresponding expressions for the constraints have been chosen such that a feasible solution exists for all problems while, at the same time, ensuring that the global robust optimum is not given by a trivial solution.

We assess the ability of the method to find the robust optimum of a constrained problem by testing it on Problem P1, P2 and P4. All three problems have a single constraint. To get insight into the capacity of the technique to handle more constraints, it is tested on Problem P3 and P5, which have two inequality constraints. Similarly, the ability of the algorithm to deal with different kinds of constraints is analyzed by choosing some constraints to be nonlinear, e.g. P1 and P2, while keeping others linear, e.g. P4. Additionally, P3 has both linear and nonlinear constraints. Another important aspect is the evaluation of the scalability of the algorithm. To this end, Problem P4, which is a function of 10 dimensions in both the objective and the constraint, is used as a test case. P1, P2 and P3 are a function of 2 control variables and 2 environment variables, while P10 is a function of 5 control variables and 5 environment variables. Note that while a problem with 10 variables may not qualify as a large problem in deterministic optimization, it is substantially more challenging in the robust case. The nested nature of robust optimization makes that computational costs increase significantly faster with problem size than in the deterministic case.

Since the initial sampling, performed via space-filling, is random, the results of each run may be different. However, the method should be able to converge regardless of the initial samples. In order to test the repeatability and reproducibility of the algorithm, it is run 100 times on each test problem and the statistical results are analyzed. The number of initial samples n are chosen as $n = 10 \times n_d$, where n_d is the number of dimensions of the problem. The maximum function evaluations available, NT, is set to 150 for P1, P2 and P3. For the larger problem P4, NT = 450.

The algorithm's performance is also tested on a polynomial problem proposed by Bertsimas et al. as a test case for robust optimization of constrained problems (Bertsimas et al. 2010). The test case, listed as problem P5 in the Appendix, is a 2-dimensional non-convex problem with two non-linear constraints. The problem is assumed to be affected by implementation error $\Delta = [\Delta x_{c1} \Delta x_{c2}]$ such that $\|\Delta\|_2 \leq 0.5$. The uncertainty set, which takes the form of a circle, is convex. The maximum function evaluations for P5 is set to NT = 65.

In addition to testing it on analytical benchmark problems, the algorithm is also applied on an engineering case study. The problem consists of an optical filter based

on a ring resonator that is fabricated as an optical integrated circuit. The fabrication is affected by variations. The behavior of the filter is highly sensitive to these manufacturing uncertainties. Therefore the problem lends itself to evaluation of the effectiveness of the algorithm in a practical situation.

5.2 Numerical performance evaluation

5.2.1 A typical example

Before discussing the statistical performance of the algorithm, we visually compare the metamodel and the robust optimum at the final iteration of the algorithm against the reference function for problem P5. Test problem P5 was used by Bertsimas et al. to demonstrate their method on robust optimization of constrained problems (Bertsimas et al. 2010). In this work, we use the problem simply as a benchmark example. The purpose therefore is not to compare the proposed method against the approach of Bertsimas et al., since their approach is complementary to this work and can be integrated with the presented algorithm.

The test problem is nonlinear and non-convex. Therefore, it serves as a challenging test case to analyze the ability of the proposed algorithm to estimate the global robust optimum efficiently. Problem P5 is affected by implementation error $\Delta = [\Delta x_{c1} \Delta x_{c2}] \in \mathcal{U}$ such that $\|\Delta\|_2 \leq 0.5$. Since the problem has only 2 control variables that are both affected by implementation error, it is easy to visualize the function and constraints surface. Figure 7a shows a contour plot of the reference function and constraints for problem P5. Figure 7b shows the contour plot of the Kriging prediction of P5 after 45 iterations of the algorithm and 65 expensive simulations have been performed. On both plots, the location of the global robust optimum is given by a green circle. The robust optimum is circumscribed by a red circle in both Fig. 7a, b. The region bounded by this red circle is the 2-norm uncertainty set, $\|\Delta\|_2 \leq 0.5$. The red square, in the two plots, shows the location of the best worst-case cost for the objective. On the other hand, the magenta squares, in both plots, indicate regions with the highest risk of potential constraint violation for each constraint.

Visually, Fig. 7a, b seem quite similar. The location of the reference robust optimum as well as the best worst-case cost with respect to the objective also visually matches on both plots. The sample points are added to the problem in such a way that the Kriging prediction for the objective and the constraints is trustworthy in the local region in the neighborhood of the robust optimum. The figure shows that the algorithm samples the expensive function in such a way that after 65 simulations it is able to accurately estimate the location of the global robust optimum. Additionally, the constrained expected improvement criterion ensures that the whole design domain is explored and a potential solution is not missed due to any inaccuracy in local regions. In the next subsection it will be shown, based on 100 runs of the algorithm on problem P5, that the proposed approach shows consistent convergence to the robust optimum.

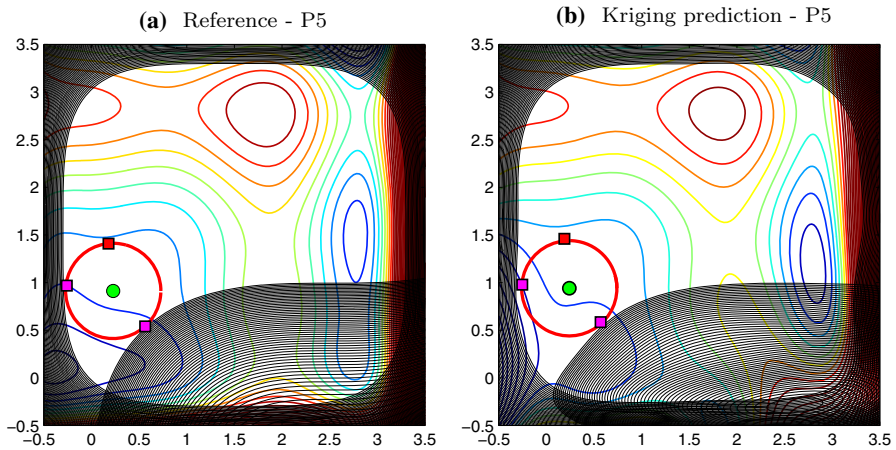


Fig. 7 Plot **a** shows contour lines of the reference function and constraints for problem P5 along with the location of the robust optimum, indicated by the *small green circle*. Plot **b** shows contour lines of the Kriging prediction of the function and constraints along with the location of the robust optimum on the Kriging surface, (*small green circle*). The *big red circle*, in both plots, indicates the extent of the 2-norm uncertainty set. The *red square*, in the two plots, shows the location of the best worst-case cost for the objective. Finally, the *magenta squares*, in both plots, indicate the worst-case cost with respect to the constraints. (Color figure online)

5.2.2 Benchmark statistics

The reference robust optima and their corresponding locations for the five numerical test problems are shown in Table 1. These optima were obtained by direct robust optimization using the analytical expressions, i.e. without metamodel error. The number of constraints, n_h , is shown in column 2 while the number of total dimensions of the problem n_d is given in the last column. The domain size in \mathbb{X}_c and \mathbb{X}_e is provided in column 3 and 4, respectively. Column 5 shows the robust optimum location for \mathbf{x}_c while column 6 gives the robust optimum location for \mathbf{x}_e . The robust optimum objective value is given by $f(\mathbf{x}_c, \mathbf{x}_e)$ in the second last column.

It is important to realize that the worst-case location for \mathbf{x}_e is different for the objective as opposed to the worst-case value for \mathbf{x}_e in the case of a constraint. The locations for \mathbf{x}_e listed in the table represent only the robust optimum location in \mathbb{X}_e . On the other hand, the maximizer in \mathbb{X}_e for each constraint has not been listed.

Table 2 shows the average numerical performance of the proposed approach based on 100 runs of each test problem. The problem number is given by the first column. The second and third column provide the mean robust optimum location in \mathbb{X}_c and \mathbb{X}_e based on the 100 runs, respectively. The mean and standard deviation of the objective value at the robust optimum for each function are given in column four and five, respectively. The average total number of expensive function evaluations required to achieve this average performance is given in the sixth column. The second last column gives the number of dimensions of each problem.

The ratio of the mean robust optimum objective value (column 4 in Table 2) to the reference robust optimum (column 7 in Table 1) is plotted in Fig. 8 for the five

Table 1 Reference results of all the test problems

Problem No.	n_h	\mathbb{X}_c	\mathbb{X}_e	\mathbf{x}_c	\mathbf{x}_e	f	n_d
P1	1	$[-5, 5]^2$	$[-5, 5]^2$	-3.9462 -2.6972	0.6251 -0.6247	87.19	4
P2	1	$[-5, 5]^2$	$[-5, 5]^2$	-1.1005 -1.5803	0.3051 2.5455	44.87	4
P3	2	$[-5, 5]^2$	$[-5, 5]^2$	-0.3502 2.5	-0.0509 5	59.59	4
P4	1	$[-5, 5]^5$	$[-3, 3]^5$	-2.0935 1.8516 -1.4695 -1.0453 0.2691	1.6765 1.2508 0.5925 0.6599 -0.6599	-0.3866	10
P5	2	$[-0.5, 3.5]^2$	-	0.228 0.912	-	7.09	2

The functions are listed in the Appendix. n_h represents the number of constraints in the problem. $f(\mathbf{x}_c, \mathbf{x}_e)$ is the optimum objective value. n_d gives the total number of dimensions

test problems. The error bars indicate the standard deviation around the optimum for each test problem. The standard deviation varies dramatically from one problem to another. This difference is a function of the local gradient in the neighborhood of the robust optimum for the individual problems. Obviously, higher gradients lead to greater relative deviation even when there is a small change in the design variables. In this context, the location of the robust optimum and their relative accuracy is also highlighted. In almost all cases, the numbers compare quite well with the reference optima locations. Where there are larger local deviations in a particular variable, this can be attributed to the fact that the objective could be locally very flat with respect to that variable in the neighborhood of the robust optimum. Additionally, in some cases two different values of a particular variable can lead to the same robust optimum. This is the case for x_{c2} for problem P2. Therefore, the average value for x_{c2} for problem P2 is completely different from the reference location.

It is also pertinent to point out that the average locations given in Table 2 are only meant to show the mean closeness of the result found to the reference location. Since the locations are averages they cannot be used to evaluate the feasibility of the final solution. The last column in Table 2 shows the percentage of solutions that were found to be infeasible when evaluated on the respective functions as a post-processing step. The results indicate that the number of optimization results that are feasible on the metamodel but infeasible on the reference function is, in general, very low.

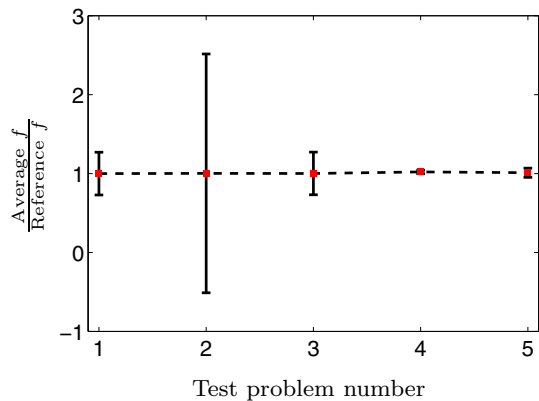
The most crucial numbers in Table 2 are given in the sixth column. This column states the total number of evaluations required, on average, to estimate the robust optimum. We note that the total number of function evaluations is quite small for all

Table 2 The average numerical performance of the algorithm based on 100 runs evaluated on the test problems provided in the Appendix

Problem no	Mean x_c	Mean x_e	Mean f	SD f	Total evaluations	n_d	Infeasible solutions (%)
P1	-3.8042	0.6521	87.11	0.271	77	4	2
	-2.6736	-0.6306					
P2	-1.0256	0.2655	44.99	1.513	83	4	1
	0.1478	2.7550					
P3	-0.3497	0.0474	59.65	0.2699	67	4	0
	2.5017	5.0000					
P4	-2.1160	1.7102	-0.3945	0.0227	392	10	0
	1.8492	1.1378					
	-1.4000	0.5209					
	-1.0605	0.7449					
	0.2625	-0.6598					
P5	0.2321	-	7.162	0.058	60	2	3
	0.9243						

The table shows the mean and standard deviation of the robust optimum along with the average robust optimum locations in x_c and x_e . The total number of evaluations on the expensive function is shown and the number of dimensions n_d are also shown

Fig. 8 Ratio of the mean robust optimum, found by the algorithm based on 100 runs, to the reference robust optimum is plotted. The error bars show the standard deviation around the mean value for each problem



five problems. Apart from problem P5, all of the problems require less than 4 samples per dimension. The largest problem, P10, in fact requires much less than 2 samples per dimension, ($2^{10} = 1024$ samples), to achieve the reported average performance.

5.2.3 Individual run analysis

Apart from studying the average performance, it is instructive to analyze individual runs for the different test cases. For problem P1, we compare the effect of choosing

different number of starting points on the intermediate accuracy of the metamodel at each iteration as well as on the convergence of the algorithm. To this end, Fig. 9 shows the robust optimum found on the objective metamodel for the problem P1 when the metamodel is initialized with only 10 initial samples. The corresponding robust optimum location's objective value on the reference function $f(\mathbf{x}_c, \mathbf{x}_e)$ at each iteration is also plotted. As expected, in the beginning the robust optimum on the metamodel and the corresponding objective value on $f(\mathbf{x}_c, \mathbf{x}_e)$ do not match. But steadily, the values become closer to each other until by about the 15th iteration they are almost the same. Figure 10 shows the same plot for problem P1, but now the number of initial samples is 40. It is interesting to observe that the robust optima found on the metamodel and the reference function are already indistinguishable from the first iteration. But this does not automatically guarantee that the algorithm will converge to the robust optimum faster than for the 10 initial samples, Fig. 9. In fact, in this particular case, the algorithm converges at the same speed for both runs

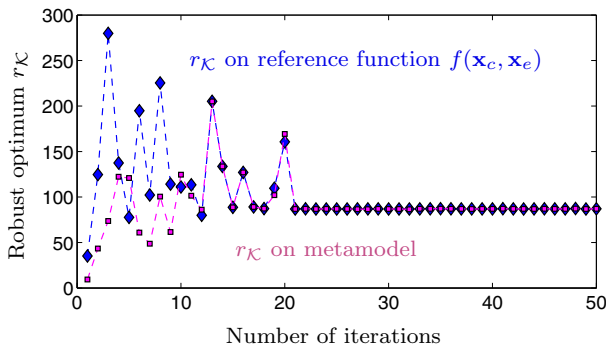


Fig. 9 The robust optimum found on the metamodel at each iteration of the algorithm for problem P1 is plotted. The metamodel is initially constructed using only 10 initial samples. The plot also shows the objective value for the robust optimum on the reference function

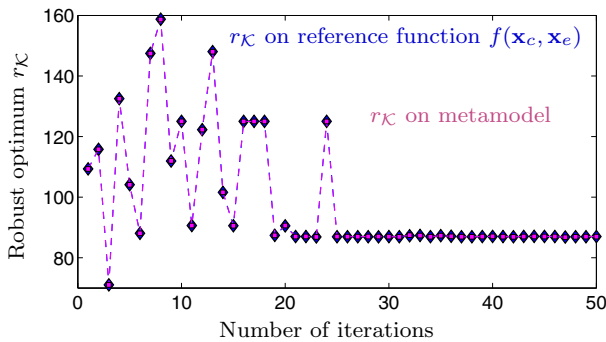


Fig. 10 The robust optimum found on the metamodel at each iteration of the algorithm for problem P1 is plotted. The metamodel is initially constructed using 40 initial samples. The plot also shows the objective value for the robust optimum on the reference function

This suggests that having a larger number of samples for the initial space filling step does not always lead to faster convergence.

Comparison of problem P2 and problem P3 is also enlightening since the two problems have the same objective function and the same first constraint. Problem P3 has a second linear constraint that is not present in P2. It was mentioned in the discussion of the average results in Table 2 that x_{c2} in P2 can attain two values. This is exhibited by Fig. 11 which shows that while x_{c1} has attained a constant value by 15th iteration, x_{c2} sometimes jumps up and down. The algorithm has also converged to the robust optimum by the 15th iteration. It is easy to observe why this happens by turning our attention to the function $f(\mathbf{x}_c, \mathbf{x}_e)$ in problem P2 in the Appendix. It can be seen that x_{c2} appears only once in the objective function and it makes an appearance as a quadratic term. The domain of x_{c2} in \mathbb{X}_c is $[-5, 5]$ as shown by Table 1. The quadratic term and the symmetric domain suggests that as long as the constraint function does not hinder x_{c2} from taking both positive and negative versions of its optimum location, both locations will be equally optimal. Therefore, x_{c2} is able to attain a value of 1.58 or -1.58 without affecting the robust optimum objective value.

The situation changes, however, when the second constraint is taken into account in problem P3. Figure 12 shows the optimum robust optimum location for x_{c1} and x_{c2} . The algorithm has converged to the robust optimum by the 15th iteration for this run. With the presence of the second constraint, x_{c2} is not allowed to have a value below 2.5 which means that we no longer observe the phenomenon exhibited in Fig. 11.

To investigate how the metamodel accuracy is affected by the number of dimensions in a problem, we check how the algorithm fares on the 10 dimensional problem. Figure 13 shows the robust optimum found on the objective metamodel for the problem P4 when the metamodel is initialized with 100 initial samples. Again, the corresponding robust optimum location's objective value on the reference function is also plotted. From iteration 1 to iteration 50, the objective values on the metamodel and the reference function are often completely different.

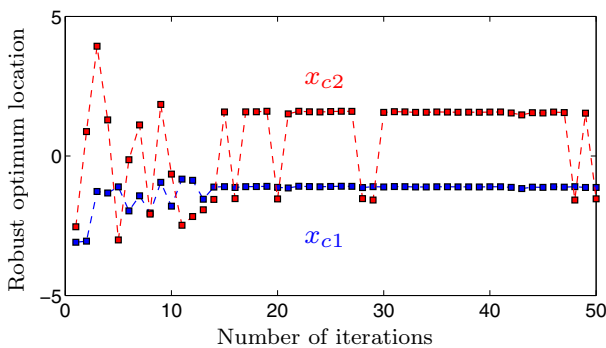


Fig. 11 The robust optimum location for x_{c1} and x_{c2} found on the metamodel at each iteration of the algorithm for problem P2 is shown. The robust optimum has been reached by the 15th iteration even though the location of x_{c2} changes in the following iterations as well

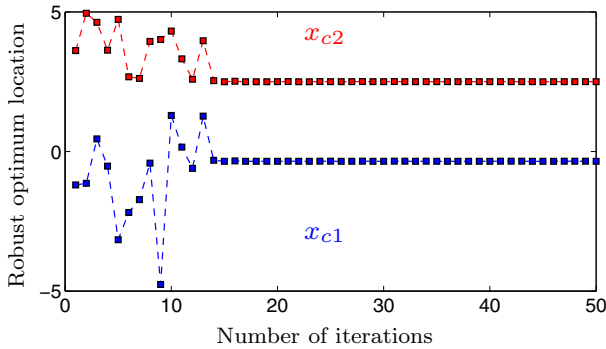


Fig. 12 The robust optimum location for x_{c1} and x_{c2} found on the metamodel at each iteration of the algorithm for problem P3 is plotted. The robust optimum has been reached by the 15th iteration

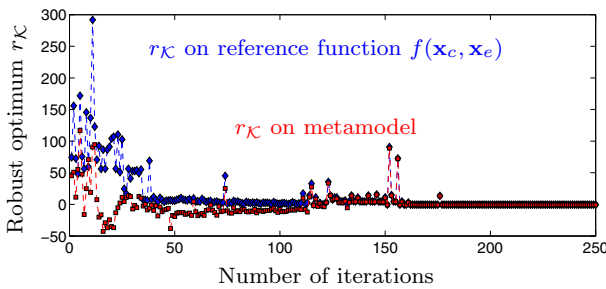


Fig. 13 The robust optimum found on the metamodel at each iteration of the algorithm is plotted for problem P4. The metamodel is initially constructed using 100 initial samples. The plot also shows the objective value for the robust optimum on the reference function

Steadily, from iteration 51 to iteration 100, the metamodel starts giving a more accurate picture of the reference function. By the 150th iteration, the two plots are practically indistinguishable. It is clear that by the 200th iteration, the algorithm has converged to the robust optimum, and the same result is given by the reference function, as exhibited by the overlap of the plots.

5.2.4 Comparison with deterministic optimization

In this subsection, the benefit of performing robust optimization over deterministic optimization for uncertain problems involving constraints is illustrated. The performance in the presence of uncertainties is shown for the deterministic optimum for problems P1 and P5. We choose these two problems since P1 is a problem affected by parameter uncertainties while P5 is affected by implementation error.

The global deterministic optimum is found on the analytical problems P1 and P5. It would be interesting to find whether the deterministic solution is feasible in the presence of uncertainties. *If the worst-case uncertainties are realized*, then the deterministic solution for both P1 and P5 is *always infeasible*. However, it could be

argued that the worst-case uncertainties may not always be realized. Then the question is, what is the yield for the deterministic solution for different realizations of uncertainties. In order to investigate this aspect, it is *assumed* that the uncertainties follow a uniform distribution. Monte-Carlo sampling is performed with 10000 sample points in the uncertainty space using the uniform distribution. The feasibility of the deterministic optimum is evaluated for these different realizations of the uncertainties.

Table 3 shows the results of this investigation. The second column shows the optimal \mathbf{x}_c for the deterministic case for Problem P1 and P5. Column 3, shows that if the worst-case uncertainties are realized then the deterministic solution is always infeasible. The last column shows the results of performing Monte-Carlo simulation with 10000 sample points in the uncertainty space using a uniform distribution. For problem P1, 69.209% of the designs are feasible. This means that the yield for this problem would be close to 70%. For problem P5, the percentage of feasible designs are less than 30%, 28.71% to be exact. For this problem, if the global deterministic optimum is chosen by the designer as a solution, the yield would be less than 30% in the presence of uncertainties that follow a uniform distribution.

In comparison, as Table 2 showed, for the robust optimum only 2 out of the 100 designs were infeasible even for the worst-case realization of the uncertainties for problem P1. In the case of P5, only 3% of the solutions found were infeasible in the worst-case. The rest of the 98 designs for P1 and 97 designs for P2 have 100% yield when we perform Monte-Carlo sampling with 10000 points using uniform distribution in the uncertainty space.

This comparison illustrates that, for constrained problems, the yield can be improved dramatically by incorporating uncertainties in the design process and by performing robust optimization on the problem. While robust optimization is inherently more expensive to perform than deterministic optimization, the presented algorithm mitigates that problem by finding the solution efficiently with the aid of metamodels and intelligent infill sampling of the original expensive simulation. The performance of the algorithm is now evaluated on an engineering problem.

5.3 Engineering case study: robust optimization of an optical filter

The algorithm is applied on an engineering problem that is very sensitive to uncertainties. The problem involves an optical filter that is fabricated as part of an

Table 3 The performance of the deterministic optimum, in terms of feasible designs, is shown for problem P1 and P5 for the worst-case realization of the uncertainties and in the case of Monte-carlo sampling with 10000 points using uniform distribution in the uncertainty space

Problem no	\mathbf{x}_c	Feasibility in the worst-case	Percentage of feasible solutions for Monte-Carlo with uniform distribution (%)
P1	-0.500	Always infeasible	69.209
	-0.300		
P5	0.2591	Always infeasible	28.7100
	1.8112		

optical integrated circuit. A schematic of the integrated device is shown in Fig. 14. Light is input at the in-port. Light is guided, via the principle of total internal reflection, in a relatively higher refractive index layer of SiN, black path in Fig. 14, that is embedded in a lower refractive index medium SiO₂. The path through which the light is guided is known as a waveguide. The device is affected by fabrication defects that directly impact the geometry of the cross section of the waveguide. These fabrication defects, in turn, affect the optical performance of the device.

The filter is realized by placing a ring shaped waveguide in between two straight waveguides. When the waveguides are within a certain proximity, light at particular wavelengths is coupled from the straight waveguide into the ring-shaped waveguide and from thereon it couples again into the other straight waveguide. Details related to the physics and operation of ring resonators may be found in the work by Bogaerts et al. (2012). The response, Fig. 15 at the drop port shows that the filter resonates at a certain frequency and drops the power at other frequencies. The device can basically be described as two couplers separated by a ring section.

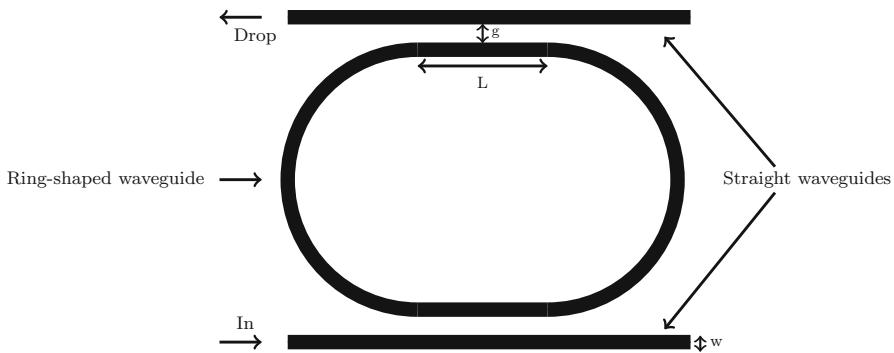


Fig. 14 Top-view schematic of an optical ring resonator. The area occupied by this integrated photonic device is less than 1 mm². Reproduced with permission from ur Rehman and Langelaar 2015

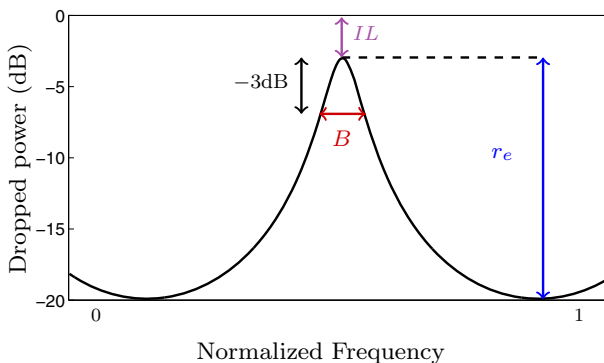


Fig. 15 Spectral response at the drop port of a ring resonator. The Bandwidth (*B*), Insertion Loss (*IL*) and Extinction Ratio (*r_e*) of the ring resonator are also indicated on the plot

There has been previous work on deterministic optimization of integrated optical filters (Ahmed et al. 2011, 2012). These sets of work have been focused on performing deterministic optimization of analytical expressions of single or multiple ring resonators with respect to the coupling ratio instead of the geometrical parameters of the device. But in order to generate a design that can actually be fabricated, the geometry needed to realize the coupling ratio has to be simulated using electromagnetic simulations of the device. Furthermore, for robust optimization, we need to perform optimization w.r.t. the explicit geometry of the waveguide since, in that scenario, the uncertainties in the geometry can also be incorporated. Therefore, the use of expensive electromagnetic simulations to realize the geometry is typically unavoidable if the user wants to find a robust optical filter design that can be fabricated.

In this work, we are interested in optimizing the bandwidth, B , of the filter at $-3dB$, Fig. 15. The problem involves two constraints related to the Insertion loss IL and the extinction ratio r_e , Fig. 15. A commercial software package (Phoenix Software 2014) is used to simulate the optical filter. The cost of a single simulation in this case is approximately ten minutes. In general, certain expensive simulations can take much more time, for example, several hours to days for a single response. The proposed algorithm is also meant to address such problems.

The quantities of interest are very sensitive to deviations caused by fabrication defects. Therefore, performing robust optimization on the device can lead to significant improvement in the overall yield. The problem is affected by both implementation error type uncertainties and parametric uncertainties. The design variables of the problem are $\mathbf{x}_c \in [w \ g \ L]$ where w is the width of the waveguides while g is the gap between the straight and the ring section, Fig. 14. L is the length of the straight coupling section in the ring. An implementation error, $\Delta w \in \mathcal{U}$ affects both the gap g and the width w . The parametric uncertainty Δt is the uncertainty in the out-of-plane thickness of the waveguide. The robust optimization problem is defined as

$$\begin{aligned} & \min_{w,g,L} \max_{\Delta t, \Delta w \in \mathcal{U}} -B, \\ \text{s.t. } & \max_{\Delta t, \Delta w \in \mathcal{U}} -r_e + 10 \text{ dB} \leq 0 \\ & \max_{\Delta t, \Delta w \in \mathcal{U}} IL + 20 \text{ dB} \leq 0 \end{aligned} \quad (20)$$

where $w \in [0.9, 1.27] \mu\text{m}$, $g \in [0.9, 1.4] \mu\text{m}$ and $L \in [100, 300] \mu\text{m}$. The range of the implementation error uncertainty set \mathcal{U} is $[-0.1, 0.1] \mu\text{m}$ while the parametric uncertainty set is given by $\Delta t \in [-3, 3] \text{nm}$. The deterministic waveguide thickness is $t = 32 \text{nm}$ and the radius of the ring section is $600 \mu\text{m}$.

The proposed method is applied to identify the robust optimum of the filter. The result is compared to the deterministic optimum of the problem that is determined by applying the deterministic optimization using constrained expected improvement (Parr et al. 2012a). The optimization problem without uncertainties is defined as

Table 4 Comparison of the robust optimum found by the proposed approach with the deterministic optimum for the filter bandwidth. Higher values for B indicated better performance

Optimum	w	g	L	Δw	Δt	Deterministic B	Worst-case B	Expensive simulations
Deterministic	1.11	1.259	220.48	-0.1	-0.003	6.16 GHz	Infeasible	225
Robust	1.17	1.088	108.70	0	-0.003	3.42 GHz	2.19 GHz	225

$$\begin{aligned}
 & \min_{w,g,L} -B, \\
 \text{s.t. } & -r_e + 10 \text{ dB} \leq 0 \\
 & IL + 20 \text{ dB} \leq 0.
 \end{aligned} \tag{21}$$

Both optimization runs are allowed a total of 225 expensive simulations, equivalent to 37.5 hours of simulation time. A large number of total simulations is used since the problem is quite nonlinear. Table 4 shows a comparison of the robust optimum estimated by the proposed approach with the deterministic optimum found via constrained expected improvement (Parr et al. 2012a). Both, the deterministic and the robust optimization algorithms are initialized with $n = 10 \times n_d$ samples chosen via LHS. It should be noted that n is different for the two algorithms since the total number of variables (due to the presence of parametric uncertainties) is greater for the robust optimization algorithm. Columns 2 to 4 provide the optimal locations for W , g and L for both algorithms. Columns 5 and 6 give the worst-case location with respect to the objective in ΔW and Δt . The next two columns show the deterministic and worst-case bandwidth for the two methods. Apart from the worst-case location for the deterministic optimum all other solutions were found to be feasible since neither of the two constraints was violated. The feasibility as well as the actual value of the bandwidth B for the deterministic worst-case location was found on the simulator as a post processing step. The deterministic optimum is found at the boundary of the first constraint, therefore the worst-case bandwidth at the deterministic location turns out to be infeasible.

It is interesting to note that the worst-case location with respect to the objective for the robust optimum does not occur at the boundary of the set Δw . This shows that the behavior of the function inside the uncertainty set is non-convex. While the nominal performance of the robust optimum is suboptimal compared to the deterministic optimum, the robust solution has the advantage that it remains feasible even in the worst case.

6 Conclusion

In this work, we have presented a novel technique for efficient global robust optimization of expensive simulation based problems involving constraints. The efficiency of the approach derives from the surrogate-based optimization strategy

employed. Kriging was chosen as the surrogate due to the availability of an estimate of the error in the interpolator.

We extended the applicability of the Kriging-based constrained optimization framework to the non-deterministic case, where the problem was affected by uncertainties in its parameters. Adapted infill sampling criteria for expected improvement and probability of feasibility were developed that enabled fast convergence to the global robust optimum of constrained problems. The proposed technique provides a viable alternative to fixed Design of Experiments based approaches. The efficiency of the adaptive sampling scheme is particularly important for higher dimensional problems, for which typical space-filling techniques fail to obtain a reasonable solution using a limited number of simulations. In addition to its robustness against the error arising from parametric uncertainties, the method was also made robust against metamodel error using the strategy suggested by Stinstra and den Hertog (2008). It is pertinent to point out here that the applicability of the method described in this work is not limited to Kriging. Instead, any interpolation strategy that also provides an error estimate in the interpolation can replace the Kriging framework employed in the proposed technique.

Several benchmark problems were used to analyze the numerical performance of the algorithm. Due to the random nature of the initial sampling, the algorithm was run 100 times on each test problem and the statistical results were investigated. It was shown that the algorithm exhibited efficient reliable convergence to the global robust optimum for all test problems. In addition, the algorithm was also applied on an engineering problem where the bandwidth of an optical filter was optimized. It was shown that while the deterministic optimum of the problem gives an infeasible solution in the worst-case, the robust optimum always remains feasible even when considering the uncertainties.

While the method was applied on cheap models developed via Kriging, the approach can equally easily be applied using any other interpolation method that provides an error estimate in the interpolation. The proposed technique is therefore widely applicable and presents a novel opportunity to efficiently investigate robust optimization of different expensive computer simulation based problems affected by uncertainties.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix 1: Kriging

The basis function in Kriging is given by the following Gaussian correlation function:

$$\text{Corr}[Y(\mathbf{x}_i), Y(\mathbf{x}_j)] = \exp\left(-\sum_{q=1}^k \theta_q (x_{iq} - x_{jq})^2\right) \tag{22}$$

where \mathbf{x}_i and \mathbf{x}_j are any two locations in the domain and k represents the total number of dimensions of the problem. θ_q , μ and σ^2 are varied such that the likelihood of the observed data is maximized. The Maximum Likelihood Estimate (MLE) for the prediction \hat{y} is given by

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T \mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{23}$$

where $\hat{\mu}$ is the estimated value for the mean, \mathbf{R} is the $N \times N$ correlation matrix between the N sample points, \mathbf{r} is the vector of correlations between the observed data and the new prediction, while \mathbf{y} is the observed response. The correlation vector \mathbf{r} and the correlation matrix \mathbf{R} may be found via eq. (22).

The Mean Squared Error (MSE) in the Kriging prediction is given by

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left[1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right], \tag{24}$$

The MSE $s^2(\mathbf{x})$ is zero at sample point locations because the true function value is known at these locations.

Appendix 2: Test problems

The following test problems have been used to test the ability of the algorithm to converge to the global robust optimum. The objective functions, **P1** to **P4**, were originally used as example problems for unconstrained min-max optimization in Rustem and Howe (2002).

P1

$$\begin{aligned} f(\mathbf{x}_c, \mathbf{x}_e) &= 5(x_{c1}^2 + x_{c2}^2) - (x_{e1}^2 + x_{e2}^2) \\ &\quad + x_{c1}(-x_{e1} + x_{e2} + 5) + x_{c2}(x_{e1} - x_{e2} + 3). \end{aligned} \tag{25}$$

$$h(\mathbf{x}_c, \mathbf{x}_e) = -x_{c1}^2 + 5x_{c2} - x_{e1} + x_{e2}^2 - 1.$$

P2

$$\begin{aligned} f(\mathbf{x}_c, \mathbf{x}_e) &= 4(x_{c1} - 2)^2 - 2x_{e1}^2 + x_{c1}^2 x_{e1} - x_{e2}^2 + 2x_{c2}^2 x_{e2}. \end{aligned} \tag{26}$$

$$h(\mathbf{x}_c, \mathbf{x}_e) = 5x_{c1} - x_{c2}^2 + x_{e1} + x_{e2} - 2.$$

P3

$$\begin{aligned} f(\mathbf{x}_c, \mathbf{x}_e) &= 4(x_{c1} - 2)^2 - 2x_{e1}^2 + x_{c1}^2 x_{e1} - x_{e2}^2 + 2x_{c2}^2 x_{e2}. \end{aligned} \tag{27}$$

$$h_1(\mathbf{x}_c, \mathbf{x}_e) = 5x_{c1} - x_{c2}^2 + x_{e1} + x_{e2} - 2.$$

$$h_2(\mathbf{x}_c, \mathbf{x}_e) = -2x_{c2} + x_{e1}.$$

P4

$$\begin{aligned}
 f(\mathbf{x}_c, \mathbf{x}_e) &= 2x_{c1}x_{c5} + 3x_{c4}x_{c2} + x_{c5}x_{c3} + 5x_{c4}^2 \\
 &\quad + 5x_{c5}^2 - x_{c4}(x_{e4} - x_{e5} - 5) \\
 &\quad + x_{c5}(x_{e4} - x_{e5} + 3) + \sum_{i=1}^3 x_{ei}(x_{ci}^2 - 1) - \sum_{i=1}^5 (x_{ei}^2). \quad (28)
 \end{aligned}$$

$$\begin{aligned}
 h(\mathbf{x}_c, \mathbf{x}_e) &= 5x_{c1} - x_{c2} + x_{c3} + x_{c4} \\
 &\quad - x_{c5} + x_{e1} - x_{e2} + x_{e3} + x_{e4} - x_{e5}.
 \end{aligned}$$

P5

$$\begin{aligned}
 f(\mathbf{x}_c) &= 2x_{c1}^6 - 12.2x_{c1}^5 + 21.2x_{c1}^4 + 6.2x_{c1} - 6.4x_{c1}^3 \\
 &\quad - 4.7x_{c1}^2 + x_{c2}^6 - 11x_{c2}^5 + 43.3x_{c2}^4 - 10x_{c2} - 74.8x_{c2}^3 + 56.9x_{c2}^2 \\
 &\quad - 4.1x_{c1}x_{c2} - 0.1x_{c2}^2x_{c1}^2 + 0.4x_{c2}^2x_{c1} + 0.4x_{c1}^2x_{c2}. \quad (29)
 \end{aligned}$$

$$\begin{aligned}
 h_1(\mathbf{x}_c) &= (x_{c1} - 1.5)^4 + (x_{c2} - 1.5)^4 - 10.125. \\
 h_2(\mathbf{x}_c) &= -(2.5 - x_{c1})^3 - (x_{c2} + 1.5)^3 + 15.75.
 \end{aligned}$$

Problem P5 is affected by implementation error $\Delta = [\Delta x_{c1} \ \Delta x_{c2}] \in \mathcal{U}$ such that $\|\Delta\|_2 \leq 0.5$.

References

- Ahmed OS, Swillam MA, Bakr MH, Li X (2011) Efficient design optimization of ring resonator-based optical filters. *J Lightwave Technol* 29(18):2812–2817. doi:[10.1109/JLT.2011.2162818](https://doi.org/10.1109/JLT.2011.2162818)
- Ahmed OS, Swillam MA, Bakr MH, Li X (2012) A perturbation approach for the design of coupled resonator optical waveguides (CROWs). doi:[10.1117/12.907407](https://doi.org/10.1117/12.907407),
- Apley DW, Chen W (2013) Objective-oriented sequential sampling for simulation based robust design considering multiple sources of uncertainty. *J Mech Des* 135(5):51,005
- Ben-Tal A, El Ghaoui L, Nemirovski A (2009) *Robust optimization*. Princeton University Press, Princeton
- Bertsimas D, Nohadani O, Teo KM (2010) Nonconvex robust optimization for problems with constraints. *INFORMS J Comput* 22(1):44–58. doi:[10.1287/joc.1090.0319](https://doi.org/10.1287/joc.1090.0319)
- Birge JR, Louveaux F (1997) *Introduction to stochastic programming*. Springer Verlag, New York
- Bogaerts W, De Heyn P, Van Vaerenbergh T, De Vos K, Kumar Selvaraja S, Claes T, Dumon P, Bienstman P, Van Thourhout D, Baets R (2012) Silicon microring resonators. *Laser Photonics Rev* 6(1):47–73. doi:[10.1002/lpor.201100017](https://doi.org/10.1002/lpor.201100017)
- Elishakoff I, Ohsaki M (2010) *Optimization and anti-optimization of structures under uncertainty*. Imperial College Press, London
- Gurav SP, Goosen JEL, VanKeulen F (2005) Bounded-but-unknown uncertainty optimization using design sensitivities and parallel computing: application to MEMS. *Comput Struct* 83(14):1134–1149. doi:[10.1016/j.compstruc.2004.11.021](https://doi.org/10.1016/j.compstruc.2004.11.021)
- den Hertog D, Kleijnen JPC, Siem AYD (2006) The correct Kriging variance estimated by bootstrapping. *J Op Res Soc* 57(4):pp. 400–409, URL <http://www.jstor.org/stable/4102391>.
- Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. *J Glob Optim* 21(4):345–383. doi:[10.1023/A:1012771025575](https://doi.org/10.1023/A:1012771025575)
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13(4):455–492. doi:[10.1023/A:1008306431147](https://doi.org/10.1023/A:1008306431147)

- Marzat J, Walter E, Piet-Lahanier H (2012) Worst-case global optimization of black-box functions through Kriging and relaxation. *J Glob Optim* pp 1–21, doi:[10.1007/s10898-012-9899-y](https://doi.org/10.1007/s10898-012-9899-y)
- Marzat J, Walter E, Piet-Lahanier H (2013) A new strategy for worst-case design from costly numerical simulations. In: *American Control Conference (ACC)*, pp 3991–3996
- Morris M, Mitchell TJ (1995) Exploratory designs for computational experiments. *J Stat Plan Inference* 43(3):381–402. doi:[10.1016/0378-3758\(94\)00035-T](https://doi.org/10.1016/0378-3758(94)00035-T)
- Parr J, Forrester A, Keane aJ, Holden CM (2012a) Enhancing infill sampling criteria for surrogate-based constrained optimization. *J Comput Method Sci Eng* 12:25–45. doi:[10.3233/JCM-2012-0402](https://doi.org/10.3233/JCM-2012-0402)
- Parr JM, Keane aJ, Forrester aI, Holden CM (2012b) Infill sampling criteria for surrogate-based optimization with constraint handling. *Eng Optim* 44(10):1147–1166. doi:[10.1080/0305215X.2011.637556](https://doi.org/10.1080/0305215X.2011.637556)
- Phoenix Software (2014) Phoenix Software 4.8.4. URL www.phoenixbv.com
- Rustem B, Howe M (2002) *Algorithms for worst-case design and applications to risk management*. Princeton University Press, Princeton
- Sacks J, Welch W, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. *Stat Sci* 4(4):409–435
- Schonlau M (1997) *Computer experiments and global optimization*. PhD thesis, University of Waterloo, ON
- Schonlau M, Welch W (1996) Global optimization with nonparametric function fitting. In: *Proceedings of the section on physical and engineering sSciences, American Statistical Association*, pp 183–186
- Stinstra E, den Hertog D (2008) Robust optimization using computer experiments. *Euro J Op Res* 191(3):816–837. doi:[10.1016/j.ejor.2007.03.048](https://doi.org/10.1016/j.ejor.2007.03.048)
- ur Rehman S, Langelaar M (2015) Efficient global robust optimization of unconstrained problems affected by parametric uncertainties. *Struct Multidiscip Optim* pp 1–18, doi:[10.1007/s00158-015-1236-x](https://doi.org/10.1007/s00158-015-1236-x).
- ur Rehman S, Langelaar M, van Keulen F (2014) Efficient Kriging-based robust optimization of unconstrained problems. *J Comput Sci* 5(6):872–881. doi:[10.1016/j.jocs.2014.04.005](https://doi.org/10.1016/j.jocs.2014.04.005)
- Zhu P, Zhang S, Chen W (2015) Multi-point objective-oriented sequential sampling strategy for constrained robust design. *Eng Optim* 47(3):287–307. doi:[10.1080/0305215X.2014.887705](https://doi.org/10.1080/0305215X.2014.887705)