Struct Multidisc Optim (2015) 52:319–336
DOI 10.1007/s00158-015-1236-x

RESEARCH PAPER

Efficient global robust optimization of unconstrained problems affected by parametric uncertainties

Samee ur Rehman[1] · **Matthijs Langelaar**[1]

Received: 19 May 2014 / Revised: 21 November 2014 / Accepted: 7 March 2015 / Published online: 13 May 2015

Abstract A novel technique for efficient global robust optimization of problems affected by parametric uncertainties is proposed. The method is especially relevant to problems that are based on expensive computer simulations. The globally robust optimal design is obtained by searching for the best worst-case cost, which involves a nested min-max optimization problem. In order to reduce the number of expensive function evaluations, we fit response surfaces using Kriging and use adapted versions of expected improvement to direct the search for the robust optimum. The numerical performance of the algorithm is compared against other techniques for min-max optimization on established test problems. The proposed approach exhibits reliable convergence, is more efficient than previous methods and shows strong scalability.

Keywords Kriging · Min-max optimization · Expected improvement · Robust optimization · Parametric uncertainties · Worst-case design

1 Introduction

Most practical design problems have some degree of uncertainty associated with them. Problems that are sensitive to even slight perturbations may give rise to suboptimal or even infeasible solutions when optimized without incorporating uncertainties. The uncertainties may be present in the parameters or in the design variables. Uncertainty in the parameters is known as parametric uncertainty while uncertainty in the design variables is referred to as implementation error. Often the uncertainties are bounded-but-unknown (Gurav et al. 2005) and the probability distribution of the uncertainties is not available. In such a scenario, robust optimization (Ben-Tal et al. 2009; Beyer and Sendhoff 2007) has to be applied to find a robust solution.

The basic idea of robust optimization is 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. Other terms such as best worst-case optimization or min-max optimization are used to describe the same concept.

Due to its wide ranging applications, robust optimization has been a topic of intense research in several different fields. Tackling robustness has been a fairly established concept in robust control, please refer to (Zhou et al. 1996) and the references therein for more detail. From a purely engineering perspective, the pioneering paper was written by (Taguchi 1984). Considerable progress in robust optimization has been made in the field of mathematical programming in recent years (Ben-Tal et al. 2009). However, the focus has been limited to solving convex problems of varying complexity. Min-max optimization has also received a lot of attention in decision and game theory (Aghassi and Bertsimas 2006). Work in this field has shown that for certain convex-concave functions, the global robust optimum can be found by searching for a saddle point solution (Rustem and Howe 2002). Therefore, saddle point optimization has been used extensively to find the robust optimum of such unconstrained min-max problems (Rustem and Howe 2002).

✉ Samee ur Rehman
S.U.Rehman@tudelft.nl

[1] Precision and Microsystems Engineering,
Delft University of Technology,
Delft, the Netherlands

However, many practical robust optimization problems are not convex. Additionally, the underlying simulation of these unconstrained continuous min-max problems could be expensive to evaluate. There is considerably less work on finding the global robust optimum of such non-convex unconstrained problems. Unconstrained continuous min-max problems have many applications including, but not limited to, aerodynamic shape design (Ong et al. 2006), finance (Rustem and Howe 2002), game theory (Zuhe et al. 1990), fault detection (Frank and Ding 1997) and signal processing (Kassam and Poor 1985). Evolutionary algorithms have been used to find the global solution of such unconstrained continuous min-max problems for which no special structure has been assumed (Cramer et al. 2009; Shi and Krohling 2002; Ong et al. 2006; Zhou and Zhang 2010; Lung and Dumitrescu 2011). However, using evolutionary algorithms to find the min-max solution of such problems leads to extremely high computational costs since these algorithms usually require very high number of function evaluations.

Since robust optimization involves solving a nested min-max problem, applying robust optimization directly on a problem based on an expensive computer simulation usually leads to prohibitively high computational costs. Instead, one can sample the expensive function at carefully chosen points and build a response surface or surrogate (Simpson et al. 2001; Forrester and Keane 2009). However, the best sample placement depends on the characteristics of the underlying function, which makes the sampling strategy itself a challenging problem. The ultimate goal is to reduce the number of expensive function evaluations required to find the optimum. Many different kinds of response surfaces can be used to build an approximate model. These include non-interpolating methodologies such as polynomial and regression models as well as interpolating approaches, e.g. radial basis functions, Kriging and splines.

Amongst the different techniques for response surface building, Kriging (Krige 1951; Sacks et al. 1989) holds distinctive appeal due to its statistical framework which enables an estimation for the error in the interpolation between the sample points. This provides the groundwork for the development of expected improvement (Jones 2001) and Efficient Global Optimization (EGO) (Jones et al. 1998; Franey et al. 2011) procedures, which allow the adaptive placement of sample points at locations most likely to lead to the global optimum. EGO has successfully been applied in deterministic unconstrained optimization, e.g. (Forrester et al. 2008), while the convergence properties of expected improvement have been established in (Vazquez and Bect 2010). A drawback of Kriging is that the correlation matrix that contains the underlying basis functions may tend to suffer from ill-conditioning (Jones et al. 1998). Furthermore, it has also been shown that Kriging underestimates the potential error in the interpolation (den Hertog et al. 2006).

In recent work (Marzat et al. 2012, 2013), expected improvement has also been applied to perform unconstrained continuous min-max optimization of black-box functions. The robust solution is found by transforming the min-max problem into a nominal constrained optimization problem with constraint relaxation. Although the method uses less expensive function evaluations compared to evolutionary algorithms (Lung and Dumitrescu 2011), the number of expensive function calls is still quite high.

This work also deals with global robust optimization of unconstrained continuous min-max problems, where the uncertainties are considered to be bounded-but-unknown. The black box function is assumed to be continuous and to be based on an expensive simulation. To solve the min-max problem, we propose a technique that extends the established Efficient Global Optimization (EGO) algorithm for deterministic optimization to the non-deterministic case. At each iteration of the proposed algorithm, the problem is divided into a separate search for the next control variables sample location and the next parametric uncertainties sample location. The EI criteria are suitably adapted to find the robust optimum instead of the nominal optimum using a small number of expensive function evaluations. We compare the efficiency of our approach with other methods using a set of standard test problems. Since the proposed algorithm is based on surrogate construction, it is not suitable for solving very high-dimensional problems. For a more in-depth discussion on challenges involved in applying surrogate-based optimization on high-dimensional problems we refer readers to (Wang and Shan 2007).

This paper is organized as follows. We introduce the problem in Section 2 and provide the mathematical background of robust optimization for problems involving parametric uncertainties. Section 3 contains a brief description of Kriging and Efficient Global Optimization. In Section 4 a detailed description of the robust optimization algorithm based on EGO is provided. Finally, Sections 5 and 6 contain the results and conclusions, respectively.

2 Robust optimization of unconstrained problems affected by parametric uncertainties

The considered problem may formally be stated as

$$\min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} f(\mathbf{x}_c, \mathbf{x}_e), \tag{1}$$

where $\mathbf{x}_c$ is the set of control variables while $\mathbf{x}_e$ is the set of parametric uncertainties or environment variables. No special structure, such as convexity or monotonicity, *etc.* is assumed for the function $f(\mathbf{x}_c, \mathbf{x}_e)$. However,

$f(\mathbf{x}_c, \mathbf{x}_e)$ is assumed to be continuous. We seek to find the global min-max solution of $f(\mathbf{x}_c, \mathbf{x}_e)$. The function is assumed to be expensive to evaluate and under this constraint, our aim is to develop a method that finds the location of the global best worst-case cost using a small number of function evaluations. This is done by estimating the robust optimum location on a relatively much cheaper surrogate model built using Kriging. The problem may be written as,

$$\min_{\mathbf{x_c}\in\mathbb{X}_c} \max_{\mathbf{x_e}\in\mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e), \tag{2}$$

where $\mathcal{K}_f$ is the continuously differentiable Kriging model of the expensive to evaluate function $f(\mathbf{x}_c, \mathbf{x}_e)$. The primary contribution in this work is to provide an adaptive sampling scheme, dedicated to the robust optimization setting, such that the Kriging function models the behaviour of the reference function $f(\mathbf{x}_c, \mathbf{x}_e)$ very accurately in regions of interest, i.e. potential robust optimum locations, using relatively few expensive function calls of $f(\mathbf{x}_c, \mathbf{x}_e)$.

In principle, any global optimization approach can be used to perform the min-max optimization in (2). However, since the Kriging model is available and the Jacobian, Hessian information can be obtained cheaply as well, optimizers that leverage this information would be preferable. In this context there are several promising algorithms that could be used to perform robust optimization on the surrogate model. It should be noted that solving (2) is equivalent to solving a semi-infinite optimization problem (López and Still 2007) since (2) can be written as a constrained minimization problem over an infinite number of constraints. For a survey of the state of the art methods that solve (2) via a semi-infinite programming approach please refer to (Stein 2012). Many of the algorithms reviewed in (Stein 2012) require first and second order derivative information for the specified function, which can easily be provided for the continuously differentiable Kriging function.

A strategy that does not require gradient information is that of redefining (2) as a constrained minimization problem (Shimizu et al. 1997) and using constraint relaxation to reduce the continuous set $\mathbb{X}_e$ to a finite discrete set. This method was used for min-max optimization in combination with Kriging by (Marzat et al. 2012).

In this work, our focus is on the adaptive sampling strategy instead of on the particular method chosen to solve the optimization problems on the resulting surrogate model. Nevertheless, in the optimization process we have opted to make full use of the availability of the Jacobian and Hessian information. Details on the applied optimizers are given in Appendix B.

Throughout this work, the 11$^{\text{th}}$ test problem from Appendix A will be used to illustrate the steps taken by the proposed algorithm,

$$f_{11}(\mathbf{x_c}, \mathbf{x_e}) = \frac{\cos\left(\sqrt{x_{c1}^2 + x_{e1}^2}\right)}{\sqrt{x_{c1}^2 + x_{e1}^2} + 10}. \tag{3}$$

The problem is a damped cosine wave in two dimensions with $\mathbb{X}_c \in [0, 10]$ and $\mathbb{X}_e \in [0, 10]$. As shown by Fig. 1, the function is non-convex and multimodal. Both x_c and x_e have been rescaled such that the design domain is in the range $[0, 1]$. The oscillatory behaviour of the function in both dimensions suggests that it is a non-trivial problem even for nominal global optimization. Therefore, finding the global min-max solution is also a challenging exercise. The problem will be used in Section 4 to visualize the steps involved in the algorithm. In Section 5 it will be employed to demonstrate the evolution of the algorithm as it searches for the global robust optimum.

3 Kriging and efficient global optimization

3.1 Kriging

Kriging is an interpolation method that uses a stochastic process approach to construct a cheap model of the expensive function. In this work, Kriging is basically used as a fitting technique for the deterministic simulation data. A detailed derivation of Kriging surrogate construction and prediction can be found in (Sacks et al. 1989). In this section, a concise description of Kriging and EGO is provided, while omitting details of the derivation.

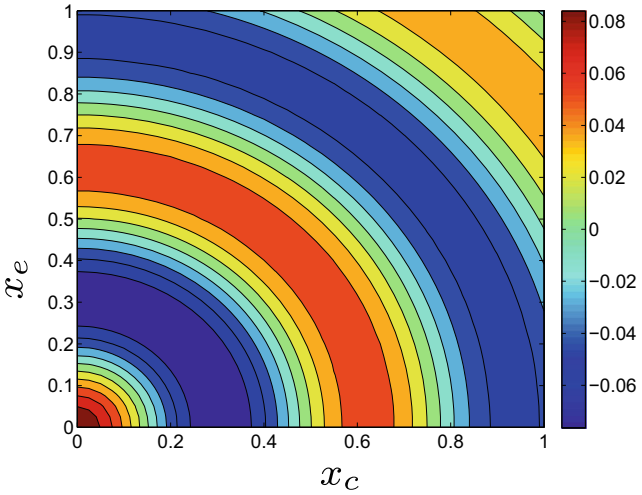

Fig. 1 Contour lines of the reference function $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. Used to illustrate the choices made by the proposed algorithm at different stages

The function response at any position $\mathbf{x}$ in the domain is assumed to be a normally distributed random variable $Y(\mathbf{x})$ with mean μ and variance σ^2. Each random variable in this stochastic process is assumed to be correlated to other random variables via the following parameterized Gaussian correlation function,

$$\text{Corr}\left[Y(\mathbf{x}_i), Y(\mathbf{x}_j)\right] = \exp\left(-\sum_{q=1}^{k} \theta_q \left|x_{iq} - x_{jq}\right|^p\right) \quad (4)$$

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. We set p to a constant value of 2. The correlation is therefore governed by the parameter θ_q and the distance between the points. The correlation is 1 when $\mathbf{x}_i = \mathbf{x}_j$ and it drops as the distance between the points increases. θ_q determines how active the q^{th} dimension is in shaping the cheap response. A higher value of θ_q means that the correlation will fall relatively quickly with respect to the q^{th} dimension as the distance between the points increases. A lower value of θ_q for the q^{th} dimension, on the other hand, will result in a relatively slower fall in correlation with increasing distance. This would also lead to a comparatively flatter response with respect to the q^{th} dimension.

The maximum likelihood estimator for a normal distribution is used to find values for θ_q, μ and σ^2 such that the likelihood of the observed data is maximized.

Once these model parameters are found, Kriging estimates the interpolation between the sample points that is most consistent with the observed data. The value of the response at these locations is found by maximizing the combined likelihood of the observed data and the Kriging prediction. The Maximum Likelihood Estimate (MLE) for the prediction $\hat{y}$ is given by

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}), \quad (5)$$

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. We find the values for the correlation vector $\mathbf{r}$ and the correlation matrix $\mathbf{R}$ via (4). It can be noted from (5) that the Kriging predictor is simply a linear combination of basis functions and a constant.

One of the advantages of the stochastic process assumption in Kriging is that the error in the predicted response $\hat{y}(\mathbf{x})$ can be estimated. The mean squared error (MSE) in the prediction is given by

$$s^2(\mathbf{x}) = \hat{\sigma}^2\left[1 - \mathbf{r}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r} + \frac{1 - \mathbf{1}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{r}}{\mathbf{1}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{1}}\right], \quad (6)$$

where $\hat{\sigma}^2$ is the maximum likelihood estimate of the variance σ^2. The fraction term in (6) represents the estimated error in using a maximum likelihood estimate $\hat{\mu}$ for the mean to compute the error instead of using the true mean. An intuitive explanation of the mean squared error in terms of the combined log-likelihood between the observed data and the prediction can be found in Jones et al. (1998), while the full derivation of (6) is available in Sacks et al. (1989).

3.2 Efficient global optimization

The estimated MSE enables adaptive sampling of the constructed surrogate. Therefore, it proves useful in order to efficiently reach the global deterministic optimum of an expensive to evaluate function. The adaptive sampling strategy of expected improvement is also based on the MSE (Jones et al. 1998). To formulate the expected improvement metric, entirely new assumptions have to be made. In this formulation, the uncertainty in the predicted value $\hat{y}(\mathbf{x})$ at any position $\mathbf{x}$ in the domain is described as a random variable $Y(\mathbf{x})$ having a normal distribution. The mean and the variance of this distribution are assumed to be given by the Kriging prediction $\hat{y}(\mathbf{x})$ and the mean squared error $\hat{s}^2(\mathbf{x})$ respectively. Let $y_{\min}$ denote the minimum objective value in the observed responses. We may be able to improve on $y_{\min}$ at a position $\mathbf{x}$ if a part of the distribution $Y(\mathbf{x})$ lies below the current minimum. We find the expectation of this improvement I by computing the expectation $E[I(\mathbf{x})] = E[\max(y_{\min} - Y, 0)]$. Using integration by parts, the expected improvement can be written as

$$E[I(\mathbf{x})] = (y_{\min} - \hat{y})\Phi\left(\frac{y_{\min} - \hat{y}}{s}\right) + s\phi\left(\frac{y_{\min} - \hat{y}}{s}\right) \quad (7)$$

where $\Phi(.)$ is the normal cumulative distribution function and $\phi(.)$ is the normal probability density function. By finding the position in the design domain where EI is maximum we get an indication of where adding a new point would be most beneficial. In this way, the Kriging prediction $\hat{y}$ and the error estimate $\hat{s}^2$ enables the development of the methodology of expected improvement (EI) and efficient global optimization (EGO) (Jones et al. 1998). Based on the Kriging prediction and MSE, EI gives an indication of the best location to sample in order to improve on $y_{\min}$. Using the EI strategy, EGO is able to converge to the global optimum of the problem by adding new data points until the maximum EI is sufficiently low.

The flowchart in Fig. 2 shows how EGO makes use of EI to find the global optimum. $\mathrm{N_T}$ is the total number of sample points available. This is determined by the simulation time and cost. The initial number of samples, n, is chosen through uniform sampling for single design variable problems or through a space-filling strategy, e.g., Latin hypercube sampling (LHS) (Morris and Mitchell 1995) for two or more design variables. The Kriging metamodel of $f(\mathbf{x})$ is denoted by $\mathcal{K}_f$. The algorithm terminates when either $\mathrm{EI_{max}}$ falls

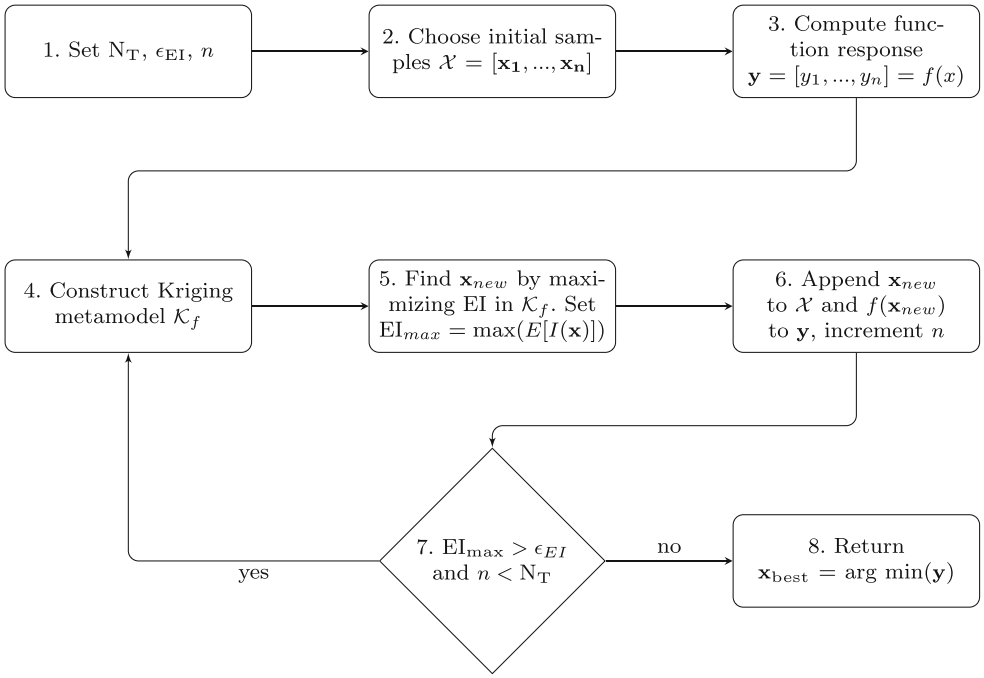

Fig. 2 Flowchart of the efficient global optimization algorithm (Jones et al. 1998). The algorithm uses Kriging and expected improvement to find the nominal optimum of an unconstrained problem with relatively few function calls of an expensive to evaluate function

below the threshold ϵ_{EI} or the total number of samples N_T is consumed.

4 Efficient global robust optimization of parametric uncertainties affected problems

4.1 Main concept

We propose an efficient global robust optimization algorithm for finding the best worst-case cost of unconstrained problems affected by parametric uncertainties. The primary goal of this method is to find the global robust solution using a relatively small number of expensive function evaluations. Before expounding the algorithm, the main concept behind the technique will be discussed.

The proposed algorithm is similar to the efficient global optimization technique in that both methods employ an initialization phase followed by an iterative surrogate-based optimization approach, where the optimal sampling location at each iteration is chosen via adaptive sampling. We refer to the proposed method as Efficient Global Robust Optimization (EGRO).

During the initialization phase, the underlying expensive simulation is sampled in the control and environment variable space, $(\mathbb{X}_c, \mathbb{X}_e)$, using an appropriate design of experiments strategy. Based on the sampled set, an approximate model that encompasses the combined space $(\mathbb{X}_c, \mathbb{X}_e)$ is constructed using Kriging.

This is followed by the iteration phase, in which the expensive function is adaptively sampled by evaluating it at the location that is most expected to improve the current robust optimum. The sampling location is searched for in two stages, firstly, in the control variable space (Section 4.3) and secondly in the environment variable space (Section 4.4). For both stages, the control variables and environment variables location is found by using modified versions of the expected improvement criterion. Thereafter, the expensive function is evaluated at the new location. The Kriging metamodel is then rebuilt with the augmented set of sampling locations and responses. The process of finding the new sampling location is repeated until convergence.

4.2 Algorithm

Figure 3 shows a flowchart that helps to visualize the main steps involved in the proposed algorithm. Comparing EGO (Fig. 2) and EGRO (Fig. 3), we observe that four steps, highlighted by a bold border for the flowchart boxes, have been added.

After initialization in Step 1, n initial sampling locations are chosen in the combined control and environment variable space through a space-filling technique in Step 2. The response at these sampling locations is evaluated on the expensive function in Step 3. Thereafter, a Kriging metamodel $\mathcal{K}_f$ of $f(\mathbf{x}_c, \mathbf{x}_e)$ is constructed using the samples and responses. Step 5 is divided into three sub-steps. In Step 5a, a reference robust optimum solution $r_{\mathcal{K}}$ is searched for on the constructed metamodel $\mathcal{K}_f$,

$$r_{\mathcal{K}} = \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e). \tag{8}$$

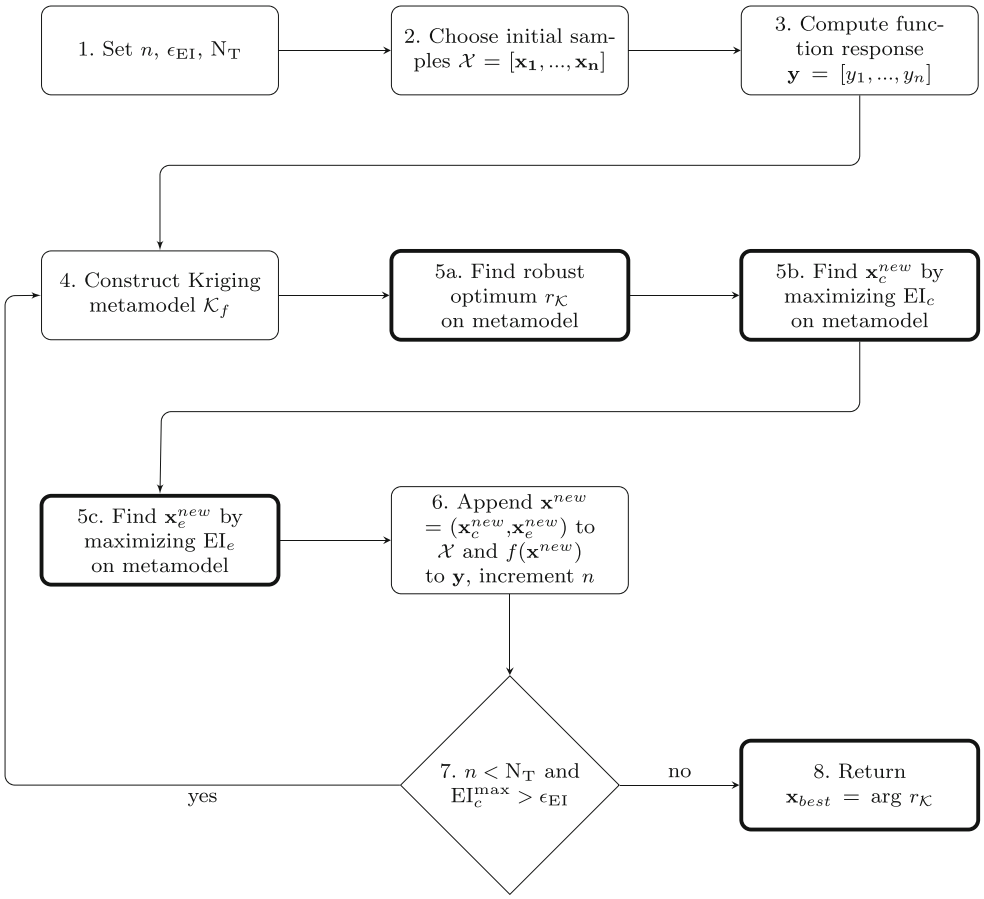

Fig. 3 Flowchart shows the algorithm for efficient global robust optimization of parametric uncertainties affected problems (EGRO). The steps with the bold borders represent the changes that have been made to the EGO algorithm in Fig. 2 in order to incorporate robust optimization in the presence of parametric uncertainties

In Step 5b and 5c, the algorithm identifies the next location $\mathbf{x}^{new}$ at which the expensive function should be evaluated. The location corresponding to the highest expectation of improvement over the current robust optimum $r_{\mathcal{K}}$ is chosen as $\mathbf{x}^{new}$. The search for $\mathbf{x}^{new}$ is performed by consecutively finding the control variables $\mathbf{x}_c^{new}$ and the environment variables $\mathbf{x}^{new}$. The key steps are treated in the next subsections.

4.3 Optimal sampling location in $\mathbb{X}_c$

In order to find $\mathbf{x}_c^{new}$ we limit the search space to the control variable space $\mathbb{X}_c$ only. $\mathbf{x}_c^{new}$ should be the location in the control variable space that is expected to give the highest improvement over the current robust optimum $r_{\mathcal{K}}$. To compute the expectation of improvement we need to know the worst-case Kriging prediction across the whole control variable space:

$$\hat{y}_{max}(\mathbf{x}_c) = \max_{\mathbf{x_e} \in \mathbb{X}_e} \mathcal{K}_f(\mathbf{x}_c, \mathbf{x}_e) \tag{9}$$

and the corresponding maximizer in the environment variable space $\mathbb{X}_e$ is denoted by $\mathbf{x}_e max$.

Figure 4 visualizes the steps described so far. Figure 4a shows a Kriging metamodel of the damped cosine function of one control variable and one environment variable plotted in Fig. 1. The worst-case Kriging prediction $\hat{y}_{max}$ is plotted in Fig. 4b. The robust optimum $r_{\mathcal{K}}$ is the minimum value on this curve. $r_{\mathcal{K}}$ is determined in Step 5a.

As in deterministic expected improvement, we assume that the uncertainty in the value of the worst-case Kriging prediction, $\hat{y}_{\max}$, at any point $\left(\mathbf{x}_c, \mathbf{x}_e^{max}\right)$ can be modelled using a normally distributed random variable Y_{max} with mean $\hat{y}_{max}$ and variance $s^2\left(\mathbf{x}_c, \mathbf{x}_e^{max}\right)$. The term $s^2\left(\mathbf{x}_c, \mathbf{x}_e^{max}\right)$ represents the Kriging mean squared error at $\left(\mathbf{x}_c, \mathbf{x}_e^{max}\right)$.

An improvement of the worst-case Kriging prediction $\hat{y}_{max}$ over the current robust optimum $r_{\mathcal{K}}$ occurs when $Y_{max} < r_{\mathcal{K}}$. The expected improvement is found by computing the expected value of the improvement $I_c = \max(r_{\mathcal{K}} - Y_{max}, 0)$ under the normal distribution setting,

$$\underbrace{E[I_c(\mathbf{x}_c)]}_{\mathrm{EI}_c} = \int_{I_c=0}^{I_c=\infty} I_c \frac{\exp\left(-\frac{t_c^2}{2}\right)}{\sqrt{2\pi}s} dI_c, \tag{10}$$

with

$$t_c = \frac{r_{\mathcal{K}} - I_c - \hat{y}_{max}}{s}, \; s = s\left(\mathbf{x}_c, \mathbf{x}_e^{max}\right) \tag{11}$$

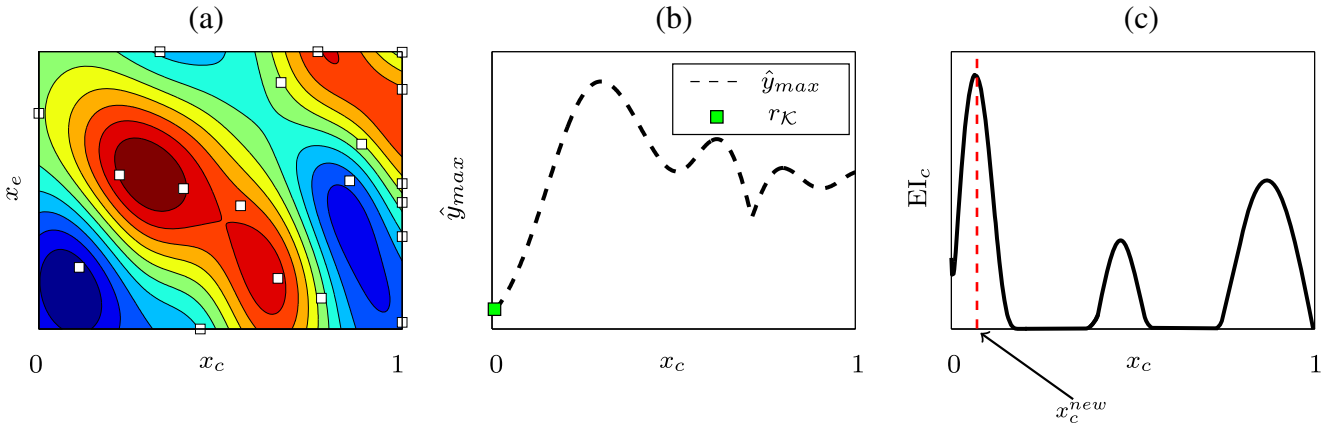

Fig. 4 The Kriging metamodel of a two-dimensional function is shown in (**a**). The worst-case Kriging prediction, $\hat{y}_{max}$, is plotted in (**b**). The reference robust optimum $r_{\mathcal{K}}$ is the minimum value obtained on $\hat{y}_{max}$. EI$_c$ is plotted in (**c**). $\mathbf{x}_c^{new}$ is the global maximizer of EI$_c$

Since the standard normal probability density function is defined as

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-z^2}{2}\right), \tag{12}$$

the modified expected improvement criterion EI$_c$ can be simplified as

$$E[I_c(\mathbf{x}_c)] = (r_{\mathcal{K}} - \hat{y}_{max}) \int_{t_c=-\infty}^{t_c=\frac{r_{\mathcal{K}}-\hat{y}_{max}}{s}} \phi(t_c)dt_c - s \int_{t_c=-\infty}^{t_c=\frac{r_{\mathcal{K}}-\hat{y}_{max}}{s}} t_c\phi(t_c)dt_c. \tag{13}$$

We recognize the first integral in (13) as the normal cumulative distribution function $\Phi\left(\frac{r_{\mathcal{K}}-\hat{y}_{max}}{s}\right)$. The second integral in (13) can be solved by using the substitution $z = \frac{-t_c{}^2}{2}$. The final expression for EI$_c$ is

$$E[I_c(\mathbf{x}_c)] = (r_{\mathcal{K}} - \hat{y}_{max})\Phi\left(\frac{r_{\mathcal{K}}-\hat{y}_{max}}{s}\right) + s\phi\left(\frac{r_{\mathcal{K}}-\hat{y}_{max}}{s}\right). \tag{14}$$

The global maximizer of (14) is the new control variable location $\mathbf{x}_c^{new}$ at which the expensive function should be evaluated since this is the control variables location that gives the highest expectation of improvement. The flowchart in Fig. 3 shows that we search for this global maximizer in Step 5b. Figure 4c shows the expected improvement EI$_c$ as a function of x_c, along with the location of the maximizer x_c^{new}. Computing $r_{\mathcal{K}}$ and EI$_c$ could be computationally expensive since they both involve estimation of $\hat{y}_{max}$. For a more in-depth discussion on calculation of $r_{\mathcal{K}}$ and EI$_c$ please refer to Appendix B.

4.4 Optimal sampling location in $\mathbb{X}_e$

Once $\mathbf{x}_c^{new}$ has been identified, we need to find the location $\mathbf{x}_e$, in the environment variable space $\mathbb{X}_e$, at which the expensive function should be evaluated. In this space, the worst-case cost (maximum) is of interest. Let the deterministic maximum on the metamodel, with respect to $\mathbf{x}_e$, at the new control variable location $\mathbf{x}_c^{new}$ be given by

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

Figure 5a shows the same Kriging metamodel of the damped cosine problem as Fig. 4a. The new sampling location in the control variable space, $\mathbf{x}_c^{new}$, is indicated. The Kriging surface along the red line at $\mathbf{x}_c^{new}$ is reproduced as a function of $\mathbf{x}_e$ in Fig. 5b. The maximum on this slice of the Kriging surface gives the reference worst-case cost $g_{\mathcal{K}}$.

In order to find $\mathbf{x}_e^{new}$, a modified expected improvement criterion is employed once again. To formulate the EI measure, it is again assumed that the uncertainty in the value of the Kriging prediction $\hat{y}\left(\mathbf{x}_c^{new}, \mathbf{x}_e\right)$ at any point $\left(\mathbf{x}_c^{new}, \mathbf{x}_e\right)$ can be modelled using a normally distributed random variable Y with mean $\hat{y}$ and variance $s^2\left(\mathbf{x}_c^{new}, \mathbf{x}_e\right)$.

Note that when computing an improvement in the control variable space we were searching for the global minimum of the worst-case function. In the environment variable space we are searching for the global maximum of the function at $\mathbf{x}_c^{new}$. An improvement is sought over the current maximum $g_{\mathcal{K}}$ and this occurs when $Y > g_{\mathcal{K}}$. We define an improvement I_e over $g_{\mathcal{K}}$ as $I_e = \max(Y - g_{\mathcal{K}}, 0)$. Given the normal distribution setting, the expected improvement can be written as,

$$\underbrace{E\left[I_e\left(\mathbf{x}_c^{new}, \mathbf{x}_e\right)\right]}_{\text{EI}_e} = \int_{I_e=0}^{I_e=\infty} I_e \frac{\exp\left(-\frac{t_e{}^2}{2}\right)}{\sqrt{2\pi}s} dI_e, \tag{16}$$

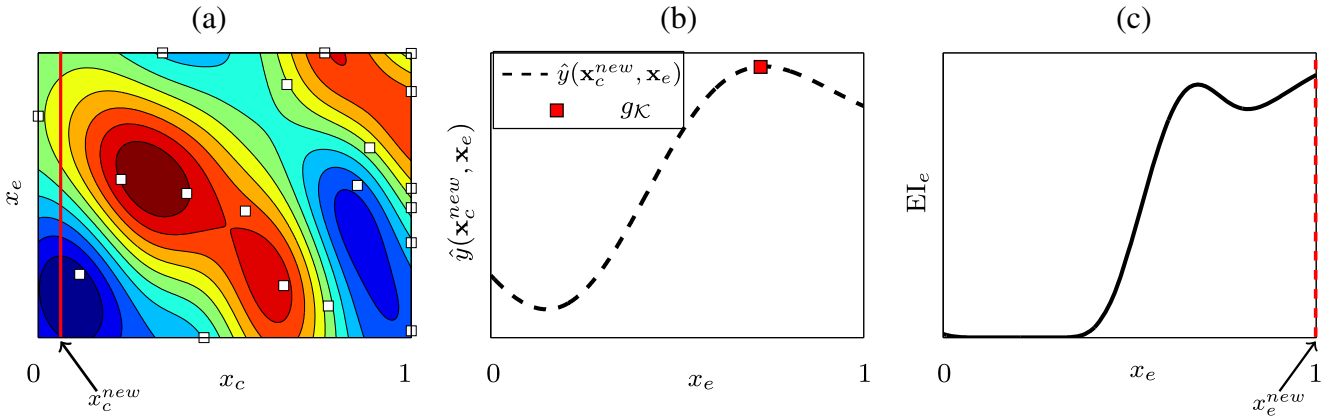

Fig. 5 The Kriging metamodel of a two-dimensional function is shown in **a** along with the location of $\mathbf{x}_c^{new}$. The Kriging prediction at $\mathbf{x}_c^{new}$, corresponding to the red line on the left plot, is plotted as a one-dimensional function with respect to $\mathbf{x}_e$ in (**b**). EI_e is plotted in (**c**). $\mathbf{x}_e^{new}$ is the global maximizer of EI_e

with

$$t_e = \frac{\hat{y} - I_e - g_{\mathcal{K}}}{s}, \; s = s\left(\mathbf{x}_c^{new}, \mathbf{x}_e\right). \tag{17}$$

Using derivation similar to the one shown for EI_c, the expected improvement simplifies to

$$\underbrace{E\left[I_e\left(\mathbf{x}_c^{new}, \mathbf{x}_e\right)\right]}_{\mathrm{EI}_e} = (\hat{y} - g_{\mathcal{K}})\Phi\left(\frac{\hat{y} - g_{\mathcal{K}}}{s}\right) + s\phi\left(\frac{\hat{y} - g_{\mathcal{K}}}{s}\right). \tag{18}$$

The global maximizer of (18) is the new environment variable location $\mathbf{x}_e^{new}$ at which the expensive function should be evaluated. Step 5c in Fig. 3 involves the search for this global maximizer. Figure 5c shows the expected improvement EI_e as a function of x_e, and the location of the maximizer x_e^{new}.

Next, in Step 6 of the flowchart, the expensive to evaluate function f is sampled at $\left(\mathbf{x}_c^{new}, \mathbf{x}_e^{new}\right)$. Step 7 is a conditional statement. If the total number of samples available is not exhausted or $\mathrm{EI}_c^{\max}$ is greater than the threshold ϵ_{EI}, the algorithm returns to Step 4 where the Kriging metamodel is reconstructed with the additional sample and the process of finding $\mathbf{x}^{new}$ is repeated. Otherwise, the algorithm terminates in Step 8 and the argument of the last robust optimum found, $r_{\mathcal{K}}$, is returned as the robust optimum location, $\mathbf{x}_{best}$.

4.5 Algorithm choices and rationale

Here we discuss two important choices made in the definition of the EGRO algorithm.

1. Use of EI_e

Instead of computing the expected improvement EI_e in the environment variable space, a simpler option could have been to sample the location of the deterministic maximum $g_{\mathcal{K}}$. However, sampling the location of the deterministic maximum at $\mathbf{x}_c^{new}$ is not viable since this often leads to the algorithm resampling the same location in the next iterations. The algorithm would stall in such a situation and would fail to converge. Furthermore, not only would the resampling be a waste of expensive function evaluations, but it would also cause the Kriging correlation matrix $\mathbf{R}$ to become ill-conditioned. The use of EI_e effectively avoids such problems because the uncertainty near sampled points is low, resulting in low EI values.

2. Formulation of EI_c

The expression for EI_c, (10), is approximate since we lack vital information to find the exact expression. In our formulation of EI_c, we first evaluated the deterministic maximum $\hat{y}_{max}$, with respect to the environment variable space $\mathbb{X}_e$, for a fixed $\mathbf{x}_c$. This evaluation was repeated until the worst-case Kriging prediction was known for the complete $\mathbb{X}_c$ space. Thereafter, we imposed the assumption that each point on the deterministically found worst-case Kriging prediction surface is actually a random variable, whose mean is given by $\hat{y}_{max}$ and variance is given by the Kriging mean squared error. This assumption enabled us to find an analytical expression for EI_c (14) similar to the one for the nominal EI (7).

To find the exact expression for EI_c, we need to first retain the assumption from deterministic EGO that each point on the Kriging surface is a random variable, whose mean is given by the Kriging prediction and variance is given by the Kriging mean squared error. Now we are interested in finding the distribution of the maximum (Arellano-Valle and Genton 2008), with respect to the environment variables, for a fixed $\mathbf{x}_c$. However, to perform this operation, additional assumptions concerning the joint distribution of the random variables are required. In

all likelihood, the distribution of the maximum would be non-Gaussian and the computation of expected improvement would require numerical integration (ur Rehman et al. 2014). The lack of joint distribution information and the high cost of performing numerical integration makes exact evaluation of EI_c less attractive. Therefore, a more pragmatic approach of computing the worst-case Kriging prediction as a deterministic quantity and imposing a Gaussian distribution assumption on $\hat{y}_{max}$ was taken to find an analytical expression for EI_c. Further details on the rationale behind this approach may be found in (ur Rehman et al. 2014). To what extent this choice affects the effectiveness of EGRO is investigated using numerical benchmark tests.

5 Results

5.1 Test problems and evaluation methodology

The algorithm is tested on a set of standard robust optimization problems from literature of varying dimensions and type. The 13 test problems are provided in the Appendix. Problems 1 to 7 have been applied in (Rustem and Howe 2002; Marzat et al. 2012) while problems 8 to 13 have been used in (Ong et al. 2006; Cramer et al. 2009; Shi and Krohling 2002; Lung and Dumitrescu 2011; Marzat et al. 2012, 2013).

To illustrate the practical value of the algorithm, an engineering case study has been added to the set of numerical examples. The engineering problem is that of an optical filter that can be affected by manufacturing uncertainties, leading to deterioration in parameters of interest.

The first seven problems, f_1 to f_7, are convex in the control variable space $\mathbf{x}_c$ and concave in the environment variables space $\mathbf{x}_e$. These problems have been optimized using saddle-point optimization in (Rustem and Howe 2002) while a Kriging-based optimization approach has been used in (Marzat et al. 2012). Problem f_1 to f_3 are 4 dimensional with 2 dimensions in $\mathbb{X}_c$ and 2 dimensions in $\mathbb{X}_e$. The largest problem is f_7 which has 5 dimensions in $\mathbb{X}_c$ and 5 dimensions in $\mathbb{X}_e$.

Test problems f_8 to f_{13} have been widely used by the evolutionary optimization community (Cramer et al. 2009; Shi and Krohling 2002; Zhou and Zhang 2010) to test evolutionary methods designed to optimize min-max problems. The same problems have also been tested by (Marzat et al. 2012, 2013). The problems have a relatively smaller size, with a maximum of 4 dimensions. Some of the test problems, such as f_{10} and f_{11}, are especially difficult to optimize due to the non-convex and highly multimodal behaviour of the functions.

All the test problems used as numerical examples in this work are established benchmark problems for robust optimization using surrogates. By using these benchmarks, the performance of the proposed algorithm can easily be compared against state of the art methods in literature in terms of number of expensive function calls needed to reach the global robust optimum. The reference results for function f_1 to f_7 have been solved to theoretically global robust optimality (Rustem and Howe 2002). Amongst the algorithms in literature, the MiMareK 1 and MiMareK 2 algorithms (Marzat et al. 2012; 2013) converge to the global robust optimum in the lowest number of expensive function evaluations. In this work, we compare the converge speed of EGRO in terms of number of expensive function evaluations required to reach the global robust optimum against currently available methods.

When constructing the metamodel, we choose the initial sampling locations using a space-filling technique such as Latin hypercube sampling. Since this type of initial sampling is non-deterministic, the algorithm is run 100 times on each test case and the results are averaged. The repeatability of the algorithm for random initial sampling is thereby tested. The average results are compared against other techniques used for min-max optimization.

For all problems, the initial number of sampling locations are chosen as $n = 10 \times n_d$, where n_d represents the number of dimensions. The threshold for termination for $EI_c^{\max}$ is fixed at $\epsilon = 10^{-7}$. The maximum number of function evaluations available is different for each function. However, this number is not allowed to exceed $N_T = 35 \times n_d$ for any problem.

5.2 Illustrative example

Before performing a detailed analysis of EGRO's numerical performance, we illustrate the evolution of the algorithm as it searches for the global robust optimum by applying it on a two-dimensional problem. The function, which is basically a damped cosine wave, is plotted in Fig. 1.

The problem is multimodal, non-convex and a function of one control variable and one environment variable. Since the problem is only 2-dimensional, it is relatively easy to visualize the choices made by EGRO at each iteration of the algorithm as it searches for the global robust optimum.

The initial variables of EGRO are set to $n = 20$, $\epsilon_{EI} = 10^{-7}$ and $N_T = 50$. We aim to find the robust optimum within 50 evaluations of $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. Latin hypercube sampling is used to choose $n = 20$ locations where the expensive function is evaluated. Before choosing these locations both the dimensions are scaled such that they lie in the range $[0, 1]$. Given this range for the dimensions,

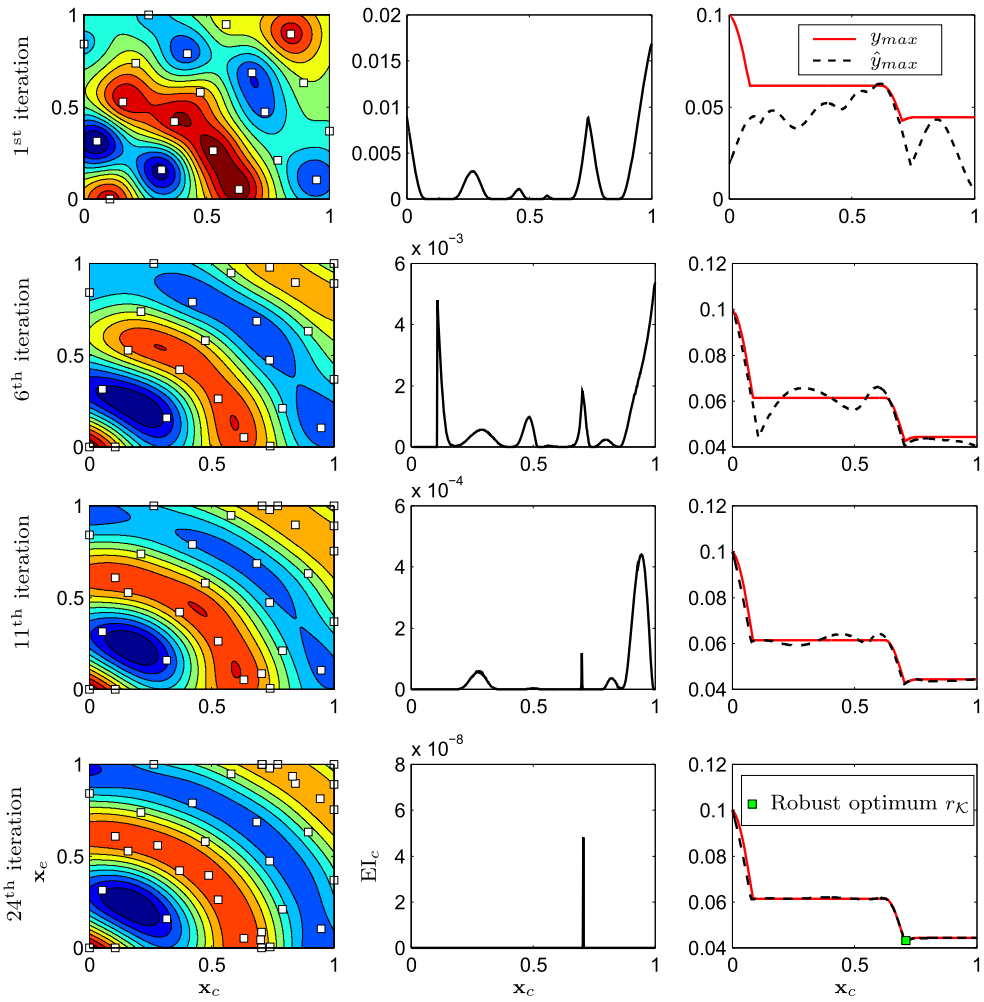

Fig. 6 Snapshots of the progress of the algorithm applied to $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. The *left column* shows the Kriging metamodel $\mathcal{K}_f$ of the function. The best worst-case expected improvement EI_c is plotted in the central column. The *right column* shows the deterministic worst-case cost $\hat{y}_{\max}$ on $\mathcal{K}_f$ and the true worst-case cost $y_{\max}$, which is provided as a reference. By the 24th iteration the robust optimum has been found

the global robust optimum of the problem is located at $\mathbf{x} = (0.7044, 1)$.

The Kriging metamodel $\mathcal{K}_f$ is built based on the sampling locations and responses. Figure 6 shows how the algorithm evolves in its search for the robust optimum of the problem. The numbers on the far left represent the iteration of the algorithm. The column on the left shows the Kriging metamodel $\mathcal{K}_f$ of the function. The expected improvement in the control variable space EI_c is plotted in the central column. The column on the right shows the deterministic worst-case cost $\hat{y}_{\max}$ on $\mathcal{K}_f$ and the true worst-case cost $y_{\max}$, computed on $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. The true worst-case cost $y_{\max}$ is provided as a reference and the algorithm does not have access to it. It is interesting to note that y_{max}, with its peaks and flat plateaus, looks slightly like a step or a staircase function. The worst-case cost experiences a very slight dip at $\mathbf{x}_c = 0.7044$. It is at this location that the worst-case cost is minimum and therefore this is the robust optimum location in $\mathbb{X}_c$.

At the first iteration, the Kriging metamodel, constructed using $n = 20$ initial samples, seems to have captured the general trend of the true function $f_{11}(\mathbf{x}_c, \mathbf{x}_e)$. However, the worst-case cost $\hat{y}_{max}$ on the metamodel does not approximate the true worst-case cost y_{max} too well at this stage. We note that for the first iteration EI_c is maximum at $\mathbf{x}_c = 1$ and this is where a new sample is added in the $\mathbb{X}_c$ domain. By the 6th iteration, there is significant improvement in the global accuracy of both the nominal Kriging metamodel $\mathcal{K}_f$ as well as the worst-case cost $\hat{y}_{max}$. However, locally, the plots show that there is significant room for improvement in some parts of the design landscape, such as between $\mathbf{x}_c = 0.1$ to $\mathbf{x}_c = 0.6$ in the worst-case cost plot. It is interesting to note that by the 6th iteration, $\hat{y}_{max}$ approximates y_{max} very well at the location of the robust optimum at $\mathbf{x}_c = 0.7044$. The two plots are also quite close to each other in the general vicinity of this robust optimum at this stage.

By the 11th iteration of EGRO, $\hat{y}_{max}$ and y_{max}, at least visually, seem to be almost the same for all values of $\mathbf{x}_c$. At this stage, the general location around $\mathbf{x}_c = 0.7044$ has also been sampled more than once. Shifting our focus to the central column of Fig. 6, we note that EI_c has also fallen to quite a low value. By the 24th iteration, the Kriging metamodel

$\mathcal{K}_f$ seems to approximate the concentric circular contours of the true function in Fig. 1 quite well. Additionally, y_{max} and $\hat{y}_{max}$ are now visually indistinguishable in the plots shown. The robust optimum, which is found by identifying the argument of $r_{\mathcal{K}}$, is returned at this point since $\mathrm{EI}_c^{\max}$ is now smaller than the threshold ϵ_{EI}.

5.3 Numerical performance evaluation

Now we turn our attention to analyzing the numerical performance of the algorithm based on the average performance of 100 runs on each test problem provided in the Appendix. Table 1 shows the reference results obtained from (Rustem and Howe 2002; Marzat et al. 2012). The table shows the robust optimum locations for the control variable and environment variable and the robust optimum object value. The number of dimensions in $\mathbb{X}_c$ and $\mathbb{X}_e$ along with their size are also provided in the second column of Table 1 for each test problem. The right-most column shows the total number of dimensions n_d for each function.

The numerical performance of EGRO on the test problems is summarized in Table 2. The first column lists the test functions. The mean robust optimum location in $\mathbb{X}_c$ and $\mathbb{X}_e$ based on 100 runs is provided in the second and third column respectively. The fourth and fifth column show the mean and standard deviation of the objective value $f_i(\mathbf{x}_c, \mathbf{x}_e)$ at the robust optimum, where i denotes the function number. Finally, the sixth column provides the total number of expensive function evaluations n_f scaled by the total number of dimensions n_d of the problem. For instance $f_7(\mathbf{x}_c, \mathbf{x}_e)$ required a total of $n_f = 288$ evaluations. The function has $n_d = 10$ total dimensions. The ratio of n_f to n_d is therefore 288/10, which is rounded up to 29 in the table.

Figure 7 shows the ratio of the average robust optimum (third column in Table 2) to the reference robust optimum (sixth column in Table 1) for all the test problems except

Table 1 Reference results of all the test problems. The functions are listed in the Appendix. The reference results have been obtained from (Rustem and Howe 2002; Marzat et al. 2012)

Test function	$\mathbb{X}_c$	$\mathbb{X}_e$	$\mathbf{x}_c$	$\mathbf{x}_e$	$f_i(\mathbf{x}_c, \mathbf{x}_e)$	n_d
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^2$	$[-5, 5]^2$	−0.4833	0.0833	−1.6833	4
			−0.3167	−0.0833		
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^2$	$[-5, 5]^2$	1.6954	0.7186	1.4039	4
			−0.0032	−0.0001		
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^2$	$[-3, 3]^2$	−1.1807	2.0985	−2.4688	4
			0.9128	2.666		
$f_4(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^2$	$[-3, 3]^3$	0.4181	0.709	−0.1348	5
			0.4181	1.0874		
				0.709		
$f_5(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^3$	$[-1, 1]^3$	0.1111	0.4444	1.345	6
			0.1538	0.9231		
			0.2	0.4		
$f_6(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^4$	$[-2, 2]^3$	−0.2316	0.6195	4.543	7
			0.2229	0.3535		
			−0.6755	1.478		
			−0.0838			
$f_7(\mathbf{x}_c, \mathbf{x}_e)$	$[-5, 5]^5$	$[-3, 3]^5$	1.4252	0.5156	−6.3509	10
			1.6612	0.8798		
			−1.2585	0.2919		
			−0.9744	0.1198		
			−0.7348	−0.1198		
$f_8(\mathbf{x}_c, \mathbf{x}_e)$	$[0, 10]$	$[0, 10]$	5	5	0	2
$f_9(\mathbf{x}_c, \mathbf{x}_e)$	$[0, 10]$	$[0, 10]$	0	0	3	2
$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	$[0, 10]$	$[0, 10]$	10	2.1257	0.0978	2
$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	$[0, 10]$	$[0, 10]$	7.0441	10	0.0425	2
$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	$[-0.5, 0.5]$	$[0, 10]^2$	0.5	0	0.25	4
	$[0, 1]$		0.25	0		
$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	$[-1, 3]^2$	$[0, 10]^2$	1	Any	1	4
			1	Any		

Table 2 The average numerical performance of EGRO based on 100 runs is evaluated on the 13 test problems provided in the Appendix

Test function	Average $\mathbf{x}_c$	Average $\mathbf{x}_e$	Average $f_i(\mathbf{x}_c, \mathbf{x}_e)$	Standard deviation f_i	$\frac{\text{Total evaluations } n_f}{\text{Number of dimensions } n_d}$
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	−0.4830	0.0828	−1.6833	2.15×10^{-5}	24
	−0.3168	−0.0828			
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	1.6953	0.7183	1.4039	1.5×10^{-3}	27
	−0.0031	−0.0015			
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	−1.1908	2.1060	−2.4689	7.4×10^{-2}	32
	0.9282	2.6966			
$f_4(\mathbf{x}_c, \mathbf{x}_e)$	0.4181	0.7082			
	0.4156	1.0907	−0.1348	2.1685×10^{-4}	25
		0.7074			
$f_5(\mathbf{x}_c, \mathbf{x}_e)$	0.1113	0.4463			
	0.1544	0.9262	1.3453	1.8286×10^{-4}	23
	0.2008	0.4022			
$f_6(\mathbf{x}_c, \mathbf{x}_e)$	−0.2318	0.6180			
	0.2233	0.3533	4.543	3.1×10^{-3}	34
	−0.6747	1.4775			
	−0.0840				
$f_7(\mathbf{x}_c, \mathbf{x}_e)$	1.4250	0.5176			
	1.6608	0.8783			
	1.2569	0.2949	−6.3509	4.3×10^{-3}	29
	−0.9738	0.1201			
	−0.7346	−0.1218			
$f_8(\mathbf{x}_c, \mathbf{x}_e)$	5	5	0	8.9×10^{-8}	11
$f_9(\mathbf{x}_c, \mathbf{x}_e)$	0	0	3	1.49×10^{-2}	18
$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	10	2.1181	0.0978	3.47×10^{-4}	25
$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	7.0496	10	0.0425	1.40×10^{-6}	30
$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	0.5	0	0.251	2.7×10^{-3}	11
	0.25	0			
$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	0.999	–	0.997	5.6×10^{-3}	16
	0.999	–			

The table shows the mean and standard deviation of the robust optimum along with the average robust optimum locations in $\mathbb{X}_c$ and $\mathbb{X}_e$. The total number of evaluations per dimension on the expensive function is shown in the right-most column for each problem

f_8. In case of f_8, the ratio has not been plotted since the reference robust optimum is zero. Nevertheless, for f_8 there is no difference between the reference objective and the average robust optimum achieved by EGRO. The error bars in Fig. 7 indicate the standard deviation (fifth column in Table 2) around the average robust optimum. It is clear from the figure that the ratio of the average to the reference robust optimum is close to 1 for all the test problems. Furthermore, the standard deviation for the objective function is also low for most of the functions. For $f_3(\mathbf{x}_c, \mathbf{x}_e)$ the standard deviation is relatively higher, but the mean robust optimum objective is still quite accurate. The relatively higher standard deviation can be explained by the fact that the norm of the gradient at the robust optimum for $f_3(\mathbf{x}_c, \mathbf{x}_e)$ is more than two orders of magnitude larger than the corresponding value for the other functions. The overall results suggest that EGRO is consistently able to accurately find the robust optimum objective value, regardless of the function type or size.

We now turn our attention to the average robust optimum location in $\mathbb{X}_c$ and $\mathbb{X}_e$ (second and third column in Table 2) for the given test problems. The reference results are provided in the fourth and fifth column in Table 1. Again the numbers compare quite favorably and differences between the reference results and the algorithm's results surface only in the second and third decimal place. It should be noted here that there is a greater scope for difference between the reference robust optimum location and those achieved by the algorithm because it may be the case that some functions are quite flat with respect to certain dimensions. Therefore, the robust optimum could be quite close to the reference result even if the robust optimum location

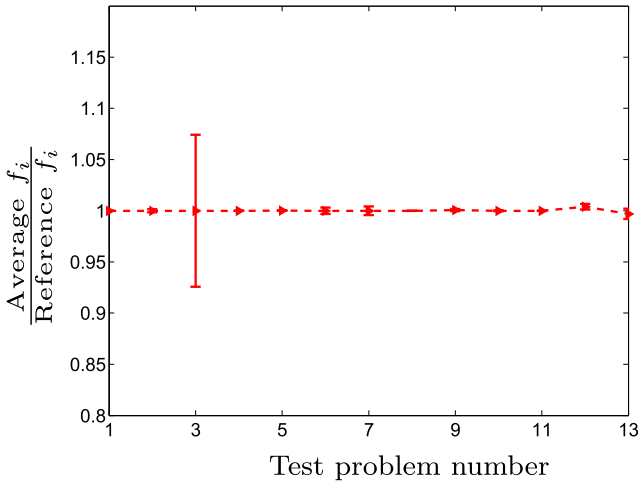

Fig. 7 Figure shows the ratio of the average robust optimum to the reference robust optimum for all the test problems. The error bars indicate the standard deviation around the average robust optimum

is relatively not as close to the reference result. Nonetheless, we note that the average robust optimum location found by the algorithm compares very well against the reference results.

The most important aspect of the algorithm is its ability to reach the global robust optimum consistently in relatively few expensive function evaluations. Finding the global robust optimum is an especially challenging problem since it involves a nested global optimization. Therefore, a large amount of expensive function evaluations could be needed to solve even quite small problems compared to the amount needed for nominal optimization.

The sixth column in Table 2 shows the total number of expensive function evaluations n_f required for the corresponding average numerical performance scaled by the total number of dimensions n_d for each test problem. The combined results of this benchmark indicate that the ratio $\frac{n_f}{n_d}$ does not increase significantly with the number of dimensions n_d since all of the ratios remain in the same vicinity of 15-35 evaluations per dimension. Even for large problems such as f_6 (7 total dimensions) and f_7 (10 total dimensions) the ratio of the total evaluations to the number of dimensions remains steady. Similarly, non-convex and highly multimodal problems such as the damped sine problem, f_{10}, and the damped cosine problem, f_{11}, do not require especially high numbers of expensive function evaluations.

5.4 Comparison with other approaches

It is pertinent here to compare the efficiency of the proposed approach against contemporary techniques for min-max optimization. Problems f_8 to f_{13} have been used for min-max optimization using evolutionary approaches (Cramer et al. 2009; Shi and Krohling 2002; Zhou and Zhang 2010). The algorithms in (Cramer et al. 2009; Shi and Krohling 2002) require a ratio of more than 10^4 total function evaluations to number of dimensions to reach the robust optimum consistently. The plots in (Zhou and Zhang 2010) suggest that the robust optimum is found based on only 110 fitness function evaluations. However, the actual expensive function evaluations are hidden within each iteration of the surrogated assisted evolutionary algorithm, resulting in total function evaluation in the same order as (Cramer et al. 2009; Shi and Krohling 2002). As shown by the last column in Table 2, EGRO is several orders faster than the evolutionary algorithms since it solves the problems f_8 to f_{13} using only a maximum ratio of 30 total function evaluations to number of dimensions.

We also compare EGRO against the recent Kriging-based min-max optimization approaches, known as MiMaReK 1 (Marzat et al. 2012) and MiMaReK 2 (Marzat et al. 2013). Marzat et al. tested MiMaReK 1 on all the test problems, f_1 to f_{13}, while MiMaReK 2 was applied on the problems f_8 to f_{13}. Marzat et al. state that under most conditions MiMaReK 2 will converge faster than MiMaReK 1. Both these algorithms were shown to use much fewer expensive function evaluations compared to evolutionary techniques in order to reach the robust optimum.

A more complete comparison can be made against MiMaReK 1, since reference average results based on 100 runs are available for it in (Marzat et al. 2012) for all the test problems. Figure 8 shows the ratio of the total function evaluations to the total number of dimensions $\frac{n_f}{n_d}$ for EGRO, MiMaReK 1 and MiMaReK 2. Comparing EGRO with MiMaReK 1 in Fig. 8, we note that EGRO uses fewer function evaluations to reach the global robust optimum for all functions. In some cases, EGRO is more than an order of magnitude faster than MiMaReK 1, such as for the two largest problems f_6 ($n_d = 10$) and f_7 ($n_d = 7$). As can be noted from Fig. 8, the difference between EGRO and MiMaReK 1 is not as dramatic for the smaller problems, but EGRO does considerably better on all functions. Figure 8 also shows the comparison of EGRO with MiMaReK 2 for the test problems f_8 to f_{13}. MiMaReK 2 uses fewer function evaluations than MiMaReK 1, but it is still slower than EGRO.

EGRO achieves average results for the robust optimum that are relatively close to the reference in Table 1 and are comparable to those found for MiMaReK 1 and MiMaReK 2 (Marzat et al. 2012, 2013). The standard deviation around the robust optimum for the 100 runs is also quite low for all problems.

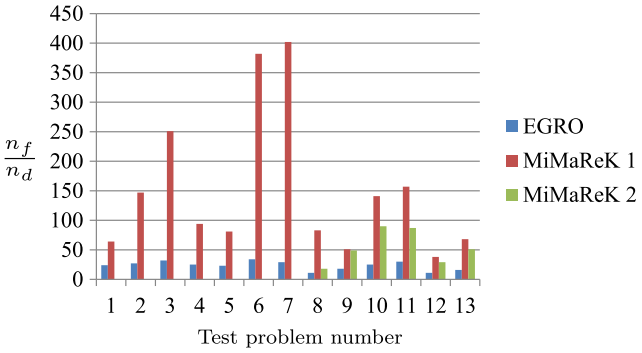

Fig. 8 Figure shows the ratio of the total function evaluations to the total number of dimensions $\frac{n_f}{n_d}$ for EGRO, MiMaReK 1 and MiMaReK 2

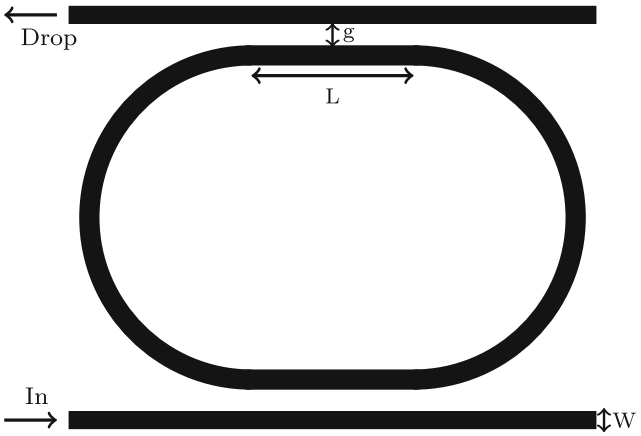

Fig. 9 Top-view schematic of an optical ring resonator

5.5 Engineering case study: robust optimization of an optical filter

In this work, we consider the robust optimization of an integrated photonic microdevice known as a ring resonator (Bogaerts et al. 2012). A top-view schematic of such a resonator is given in Fig. 9. The device consists of two straight optical paths separated by a ring-shaped optical path, all of which are integrated on a chip. These optical paths on the chip are known as waveguides. The chip consists of a single stripe of Silicon Nitride buried in Silicon Dioxide and the geometry is referred to as TripleX^{TM} (Heideman et al. 2012). When light at a certain frequency is inserted into one of the straight waveguides, it is transferred, via the ring section, into the other straight waveguide as a result of resonant coupling between the adjacent waveguides. The device operates as an optical filter that allows power to be transferred only at certain frequencies. This property motivates the use of ring resonators in many applications in optical communication and signal processing. The optical performance, however, is very sensitive to fabrication induced variations in geometry of the integrated circuit. For further details concerning the theory and application of ring resonators, please refer to (Bogaerts et al. 2012).

Figure 10 shows the spectral response at the drop port (output) of a ring resonator. In this case study, we are interested in maximizing the filter bandwidth, B, defined as the bandwidth of the response at -3dB. The design variables are the gap, g, between the straight and the ring section, the length L of the straight coupling section in the ring and the width W of the waveguides. There are two parametric uncertainties, ΔW and Δt. ΔW represents uncertainty in the width due to fabrication variations while Δt represents the uncertainty in the out-of-plane thickness of the waveguide. The robust optimization problem is defined as,

$$\min_{\mathrm{w,g,L}} \max_{\Delta \mathrm{W},\Delta \mathrm{t}} \; -B, \tag{19}$$

where $w \in [1, 1.17]\mu$m, $g \in [1, 1.3]\mu$m and $L \in [100, 300]\mu$m. The uncertainty $\Delta W \in [-0.1, 0.1]\mu$m and $\Delta t \in [-3, 3]$nm. The nominal thickness of the waveguides is $t = 32$nm. The set of control variables is $\mathbf{x}_c \in [w\ g\ L]$ and the set of environment variables consists of $\mathbf{x}_e \in [\Delta W\ \ \Delta t]$.

The ring resonator is simulated using a commercial software package, (PhoeniX Software 2014). Each simulation costs approximately 10 min. An initial Kriging metamodel is built using $10 \times n_d = 50$ samples chosen via Latin Hypercube sampling. The computational budget is set to $30 \times n_d = 150$ expensive simulations. EGRO is therefore allowed to run for 100 iterations. The robust optimum found via EGRO is compared to the nominal optimum of the problem. This nominal optimum is estimated by using the Efficient Global Optimization algorithm described in Section 3. The same total computational budget of 150 simulations is allowed for the nominal optimization and the algorithm is initialized with $10 \times n_d = 30$ samples chosen via Latin Hypercube sampling.

Table 3 shows a comparison of the robust optimum estimated by EGRO with the nominal optimum found by EGO. The optimal locations for W, g and L are provided for both optima (column 2 to 4). The worst-case location in ΔW and

Table 3 Comparison of the robust optimum and deterministic optimum for the filter bandwidth at the respective worst-case locations

	W	g	L	ΔW	Δt	Nominal B	Worst-case B	Expensive simulations
Robust optimum	1	1.032	233.64	-0.1	0.003	254.1 GHz	23.3 GHz	150
Nominal optimum	1.071	1.045	300	-0.1	-0.003	405.8 GHz	19.5 GHz	150

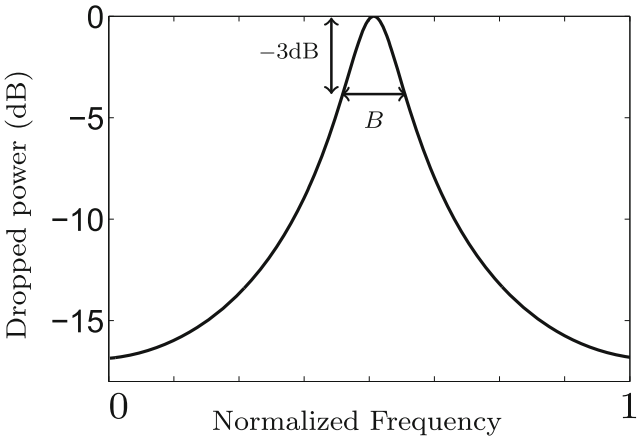

Fig. 10 Spectral response at the drop port of a ring resonator. The bandwidth of the ring resonator, B, is also indicated on the plot

Δt is also given for the two cases (column 5 and 6). Column 7 and 8 show the nominal and worst-case bandwidth found at the respective locations. It is important to note that the values given for both these quantities have been found on the expensive simulation via postprocessing after both algorithms terminate. The computational budget used by the two algorithms is given in the last column.

The performance of the robust optimum and the nominal optimum is first compared at the nominal location. The table shows that the nominal bandwidth B for the robust optimum is 254.0 GHz. On the other hand, the bandwidth at the nominal optimum is a much higher value, 405.8 GHz. However, this large difference, in reality, is not significant since even the slightest deviation from the ideal parameter values causes the bandwidth B to drop dramatically for both cases. The comparison of the nominal bandwidth to the worst-case bandwidth (second last column, Table 3) for the two optima also gives an indication of the sensitivity of the objective with respect to the parametric uncertainties. For both optima, the bandwidth falls by more than an order of magnitude when moving from the nominal to the worst-case solution.

It is therefore more instructive to compare the worst-case bandwidth found for the nominal and robust optimal locations. The worst-case bandwidth for the robust optimum is 3.8 GHz (19 percent) higher than the worst-case bandwidth for the nominal optimum. Although the robust optimum is nominally suboptimal, it will perform better than the nominal optimum in the worst-case scenario.

Interestingly, the worst-case location in ΔW and Δt for both the optima occurs at the bounds. This is purely coincidental since the objective function is, in general, non-convex with respect to ΔW and Δt for many choices of $\mathbf{x}_c$.

In addition to the final result, it is also informative to observe how the two methods, EGRO and EGO, evolve with each iteration. Figure 11 shows the worst-case bandwidth found on the metamodel at each iteration of EGRO. Observing the evolution of the worst-case cost, we note that it fluctuates from the 1st to the 35th iteration. It stays relatively constant for the next 20 iterations but starts to shift again from iteration 55 to iteration 70. Thereafter, the value remains largely steady until the algorithm terminates at the 100th iteration. An interesting observation is that the worst-case cost found on the metamodel does not exactly match the worst-case cost found on the expensive simulation, Table 3, when the algorithm terminates at the 100th iteration. This indicates that more than 150 total simulations are needed in order to improve the metamodel fidelity in the local region of the robust optimum.

The response of the expensive simulation at each newly added sample point is plotted with respect to the number of iterations of EGO in Fig. 12. The figure also shows the nominal optimum found at each iteration. Studying the

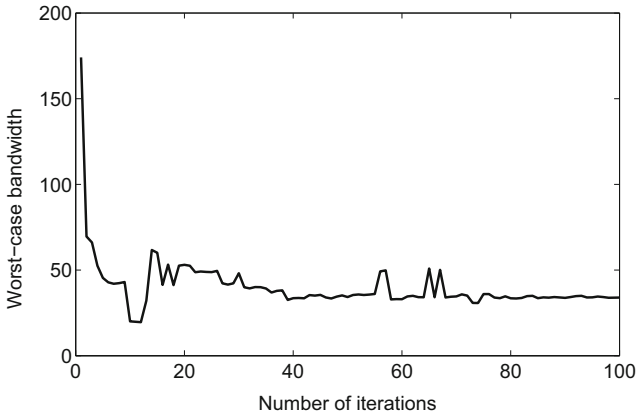

Fig. 11 The worst-case bandwidth found on the metamodel at each iteration of EGRO is plotted for the total number of iteration

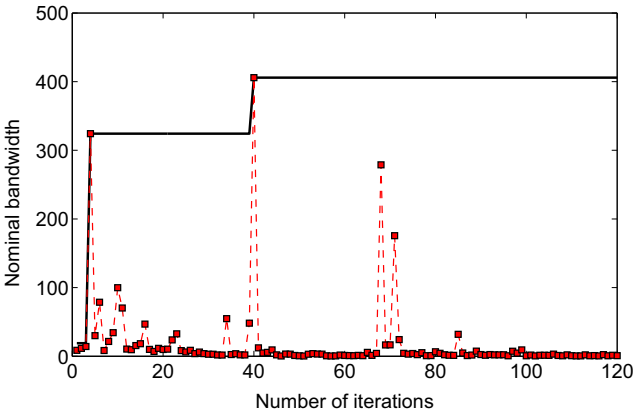

Fig. 12 The nominal optimum at each iteration of EGO is plotted as a function of the number of iterations. The figure also shows the simulated response for the new sampling location chosen at each iteration

plot, we note that the nominal optimum at any iteration in EGO is given by the maximum simulated bandwidth found till that iteration. The nominal optimum does not change after the 40th iteration since no better solution is found thereafter.

6 Conclusion

Optimization problems involving bounded-but-unknown uncertainties and expensive simulations are often encountered in practice. In this work, we have proposed a novel method for efficient global robust optimization of such problems that are affected by parametric uncertainties. To avoid extensive use of expensive function evaluations, a surrogate-based robust optimization technique was formulated. The approach depended on constructing a Kriging metamodel and adaptively sampling the resulting surrogate. The sampling locations were found using expected improvement criteria that reflected the need for finding the best worst-case cost instead of the nominal cost.

The presented algorithm was tested on several test problems found in literature. We demonstrated that the proposed approach can consistently locate the global robust optimum of these functions using relatively much fewer expensive function evaluations than the amount reported in previous work. The reproducibility of the technique was tested by observing average performance based on 100 runs on each test problem. The statistical comparison of the method against comparable approaches were also quite favorable.

In addition to the application of the algorithm on numerical problems, the technique was tested on an engineering problem as well. It was shown that the method can provide a fabrication tolerant integrated circuit design for an optical filter affected by fabrication uncertainties.

Tests on the benchmark problems showed that the proposed approach accurately found the global robust optimum. Furthermore, the algorithm scaled linearly in terms of expensive function evaluations with the number of dimensions and exhibited much faster convergence than existing techniques. Numerical results also suggest that the approximations made in the algorithm did not hinder convergence to the global robust optimum.

Appendix A: Test problems

The 13 test functions on which EGRO is applied are provided below. Functions 1 to 7 have been employed as test problems in (Rustem and Howe 2002; Marzat et al. 2012) while functions 8 to 13 have been used in (Cramer et al. 2009; Shi and Krohling 2002; Zhou and Zhang 2010; Marzat et al. 2012, 2013).

$$f_1(\mathbf{x_c}, \mathbf{x_e}) = 5(x_{c1}^2 + x_{c2}^2) - (x_{e1}^2 + x_{e2}^2) + x_{c1}(-x_{e1} + x_{e2} + 5) + x_{c2}(x_{e1} - x_{e2} + 3). \tag{20}$$

$$f_2(\mathbf{x_c}, \mathbf{x_e}) = 4(x_{c1} - 2)^2 - 2x_{e1}^2 + x_{c1}^2 x_{e1} - x_{e2}^2 + 2x_{c2}^2 x_{e2}. \tag{21}$$

$$f_3(\mathbf{x_c}, \mathbf{x_e}) = x_{c1}^4 x_{e2} + 2x_{c1}^3 x_{e1} - x_{c2}^2 x_{e2}(x_{e2} - 3) - 2x_{c2}(x_{e1} - 3)^2 \tag{22}$$

$$f_4(\mathbf{x_c}, \mathbf{x_e}) = -\sum_{i=1}^{3}(x_{ei} - 1)^2 + \sum_{i=1}^{2}(x_{ci} - 1)^2 + x_{e3}(x_{c2} - 1) + x_{e1}(x_{c1} - 1) + x_{e2}x_{c1}x_{c2} \tag{23}$$

$$f_5(\mathbf{x_c}, \mathbf{x_e}) = -x_{e1}(x_{c1} - 1) - x_{e2}(x_{c2} - 2) - x_{e3}(x_{c3} - 1) + 2x_{c1}^2 + 3x_{c2}^2 + x_{c3}^2 - x_{e1}^2 - x_{e2}^2 - x_{e3}^2 \tag{24}$$

$$f_6(\mathbf{x_c}, \mathbf{x_e}) = x_e 1(x_{c1}^2 - x_{c2} + x_{c3} - x_{c4} + 2) + x_{e2}(-x_{c1} + 2x_{c2}^2 - x_{c3}^2 + 2x_{c4} + 1) + x_{e3}(2x_{c1} - x_{c2} + 2x_{c3} - x_{c4}^2 + 5) + (5x_{c1}^2 + 4x_{c2}^2 + 3x_{c3}^2 + 2x_{c4}^2) - \sum_{i=1}^{3}(x_{ei})^2 \tag{25}$$

$$f_7(\mathbf{x_c}, \mathbf{x_e}) = 2x_{c1}x_{c5} + 3x_{c4}x_{c2} + x_{c5}x_{c3} + 5x_{c4}^2 + 5x_{c5}^2 - x_{c4}(x_{e4} - x_{e5} - 5) + x_{c5}(x_{e4} - x_{e5} + 3) + \sum_{i=1}^{3} x_{ei}(x_{ci}^2 - 1) - \sum_{i=1}^{5}(x_{ei}^2) \tag{26}$$

$$f_8(\mathbf{x_c}, \mathbf{x_e}) = (x_{c1} - 5)^2 - (x_{e1} - 5)^2 \tag{27}$$

$$f_9(\mathbf{x_c}, \mathbf{x_e}) = \min(3 - 0.2x_{c1} + 0.3x_{e1}, 3 + 0.2x_{c1} - 0.1x_{e1}) \tag{28}$$

$$f_{10}(\mathbf{x_c}, \mathbf{x_e}) = \frac{\sin(x_{c1} - x_{e1})}{\sqrt{x_{c1}^2 + x_{e1}^2}} \tag{29}$$

$$f_{11}(\mathbf{x_c}, \mathbf{x_e}) = \frac{\cos(\sqrt{x_{c1}^2 + x_{e1}^2})}{\sqrt{x_{c1}^2 + x_{e1}^2} + 10} \tag{30}$$

$$f_{12}(\mathbf{x_c}, \mathbf{x_e}) = 100(x_{c2} - x_{c1}^2)^2 + (1 - x_{c1})^2 - x_{e1}(x_{c1} + x_{c2}^2) - x_{e2}(x_{c1}^2 + x_{c2}) \tag{31}$$

$$f_{13}(\mathbf{x_c}, \mathbf{x_e}) = (x_{c1} - 2)^2 + (x_{c2} - 1)^2 + x_{e1}(x_{c1}^2 - x_{c2}) + x_{e2}(x_{c1} + x_{c2} - 2). \quad (32)$$

Appendix B: Numerical choices and optimizers

The calculation of EI_c is more computationally demanding than calculating the deterministic EI, (7), since $\hat{y}_{max}$ has to be computed for each evaluation of EI_c. In this work, Matlab implementation of the multistart framework for global optimization (Ugray et al. 2007) is used to estimate $\hat{y}_{max}$. The algorithm is a global optimization method that chooses the best solution after performing local optimization runs from multiple starting points. The Matlab implementation fmincon, of the interior point algorithm for non-linear programming (Byrd et al. 1999), was used as the local solver in this work. Since $\hat{y}_{max}$ is computed on a known analytical function, i.e. the Kriging predictor, the Jacobian and Hessian of the Kriging predictor can very cheaply and easily be computed as well. In order to enhance the efficiency, the Jacobian and Hessian were also provided to the local solver.

Estimating the robust optimum $r_{\mathcal{K}}$ on the metamodel, (8), is also a computationally intensive part of the algorithm since it requires computation of $\hat{y}_{max}$ as well. EI_c and $r_{\mathcal{K}}$ use the same strategy, described above, for estimating $\hat{y}_{max}$.

It should be noted here, that any type of global optimizer can be used to estimate $\hat{y}_{max}$. However, because $\hat{y}_{max}$, (9), has to found each time EI_c needs to be evaluated, it is best to use an approach that can use analytically supplied Jacobian and Hessian of the Kriging predictor, since they can be computed very cheaply.

The global optimization of EI_c and the outer global minimization for estimating the optimal $r_{\mathcal{K}}$ in (8) are performed via the Matlab implementation of the Simulated Annealing algorithm (Ingber 1996).

The computational cost of evaluating EI_e is much less than the calculation of EI_c, (18), since there is no internal optimization that has to be performed each time EI_e is evaluated. However, EI_e requires a precomputed value for the reference global worst-case cost $g_{\mathcal{K}}$. Matlab implementation of the Simulated Annealing algorithm (Ingber 1996) is again used for this global optimization.

References

Aghassi M, Bertsimas D (2006) Robust game theory. Math Program 107(1–2):231–273. doi:10.1007/s10107-005-0686-0

Arellano-Valle RB, Genton MG (2008) On the exact distribution of the maximum of absolutely continuous dependent random variables. Stat Probabil Lett 78(1):27–35. doi:10.1016/j.spl.2007.04.021

Ben-Tal A, El Ghaoui L, Nemirovski A (2009) Robust optimization. Princeton University Press

Beyer HG, Sendhoff B (2007) Robust optimization A comprehensive survey. Comput Method Appl M 196(33–34):3190–3218. doi:10.1016/j.cma.2007.03.003

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

Byrd R, Hribar M, Nocedal J (1999) An interior point algorithm for large-scale nonlinear programming. SIAM J Optim 9(4):877–900

Cramer AM, Sudhoff SD, Zivi EL (2009) Evolutionary Algorithms for Minimax Problems in Robust Design. IEEE Trans Evol Comput 13(2):444–453. doi:10.1109/TEVC.2008.2004422

den Hertog D, Kleijnen JPC, Siem AYD (2006) The correct kriging variance estimated by bootstrapping. J Oper Res Soc 57(4):400–409. http://www.jstor.org/stable/4102391

Forrester A, Andras S, Keane AJ (2008) Engineering design via surrogate modelling : a practical guide. Wiley

Forrester AIJ, Keane AJ (2009) Recent advances in surrogate-based optimization. Prog Aerosp Sci 45(1–3):50–79. doi:10.1016/j.paerosci.2008.11.001

Franey M, Ranjan P, Chipman H (2011) Branch and bound algorithms for maximizing expected improvement functions. J Stat Plan Infer 141(1):42–55. doi:10.1016/j.jspi.2010.05.011

Frank PM, Ding X (1997) Survey of robust residual generation and evaluation methods in observer-based fault detection systems. J Proc Contr 7(6):403–424. doi:10.1016/S0959-1524(97)00016-4

Gurav SP, Goosen JFL, VanKeulen F (2005) Bounded-But-Unknown uncertainty optimization using design sensitivities and parallel computing: Application to MEMS. Comp Struct 83(14):1134–1149. doi:10.1016/j.compstruc.2004.11.021

Heideman R, Hoekman M, Schreuder E (2012) TriPleX-based integrated optical ring resonators for lab-on-a-chip and environmental detection. IEEE J Quantum Electron 18(5):1583–1596. doi:10.1109/JSTQE.2012.2188382

Ingber L (1996) Adaptive simulated annealing (ASA): Lessons learned. Control Cybern 0001018

Jones DR (2001) A taxonomy of global optimization methods based on response surfaces. J Glob Optim 21(4):345–383

Jones DR, Schonlau M, Welch WJ (1998) Efficient Global Optimization of Expensive Black-Box Functions. J Glob Optim 13(4):455–492

Kassam S, Poor H (1985) Robust techniques for signal processing: A survey. Proc IEEE 73(3)

Krige DG (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand. J Chem Metall Min Soc S Afr 52(6):119–139

López M, Still G (2007) Semi-infinite programming. Eur J Oper Res 180(2):491–518. doi:10.1016/j.ejor.2006.08.045

Lung RI, Dumitrescu D (2011) A new evolutionary approach to minimax problems. In: 2011 IEEE Congress on Evolutionary computation (CEC), pp 1902–1905

Marzat J, Walter E, Piet-Lahanier H (2012) Worst-case global optimization of black-box functions through Kriging and relaxation. J Glob Optim:1–21. doi: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), 2013, pp 3991–3996

Morris M, Mitchell TJ (1995) Exploratory designs for computational experiments. J Stat Plan Infer 43(3):381–402. doi:10.1016/0378-3758(94)00035-T

Ong YS, Nair PB, Lum KY (2006) Max-min surrogate-assisted evolutionary algorithm for robust design. IEEE Trans Evol Comput 10(4):392–404. doi:10.1109/TEVC.2005.859464

PhoeniX Software (2014) PhoeniX Software 4.8.4. www.phoenixbv.com

Rustem B, Howe M (2002) Algorithms for worst-case design and applications to risk management. Princeton UniversityPress

Sacks J, Welch W, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. Stat Sci 4(4):409–435

Shi Y, Krohling RA (2002) Co-evolutionary particle swarm optimization to solve min-max problems. In: Proceedings of the evolutionary computation on 2002. CEC '02. Proceedings of the 2002 Congress - Volume 02, IEEE Computer Society, Washington, DC, USA, CEC '02, (pp 1682–1687)

Shimizu K, Ishizuka Y, Bard J (1997) Nondifferentiable and two-level mathematical programming. Kluwer Academic Publishers

Simpson TW, Poplinski JD, Koch PN, Allen JK (2001) Metamodels for computer-based engineering design: survey and recommendations. Eng Comput 17(2):129–150. doi:10.1007/PL00007198

Stein O (2012) How to solve a semi-infinite optimization problem. Eur J Oper Res 223(2):312–320. doi:10.1016/j.ejor.2012.06.009

Taguchi G (1984) Quality engineering through design optimization. Global Telecommunications Conference

Ugray Z, Lasdon L, Plummer J, Glover F, Kelly J, Martí R (2007) Scatter search and local NLP solvers: a multistart framework for global optimization. INFORMS J Comput 19(3):328–340. doi:10.1287/ijoc.1060.0175

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

Vazquez E, Bect J (2010) Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. J Stat Plan Infer 140(11):3088–3095. doi:10.1016/j.jspi.2010.04.018

Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. J Mech Des 129(4):370–380. doi:10.1115/1.2429697

Zhou A, Zhang Q (2010) A surrogate-assisted evolutionary algorithm for minimax optimization. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp 1–7. doi:10.1109/CEC.2010.5586122

Zhou K, Doyle JC, Glover K (1996) Robust and optimal control. Prentice-Hall, Inc., Upper Saddle River

Zuhe S, Neumaier A, Eiermann M (1990) Solving minimax problems by interval methods. BIT Numer Math 30(1):742–751