A new strategy for worst-case design from costly numerical simulations

Julien Marzat, Éric Walter, Hélène Piet-Lahanier

***Abstract*— Worst-case design is important whenever robustness to adverse environmental conditions should be ensured regardless of their probability. It leads to minimax optimization, which is most often considered assuming that a closed-form expression for the performance index is available. In this paper, we consider the important situation where this is not the case and where evaluation of the performance index is via costly numerical simulations. In this context, strategies to limit the number of these evaluations are of paramount importance. This paper describes one such strategy, which further improves the performance of an algorithm recently presented that combines the use of a relaxation procedure for minimax search and Kriging-based efficient global optimization. Test cases from the literature demonstrate the interest of the approach.**

***Index Terms*— computer experiments, Kriging, minimax, optimization, robust design, surrogate models, worst case.**

I. Introduction and problem statement

For a wide class of design problems, a design vector $\mathbf{x}_\mathrm{c}$ must be tuned to achieve the best performance possible while protecting oneself against the potentially adverse effects of an environment vector $\mathbf{x}_\mathrm{e}$. Such problems are important in the context of robust control, estimation and decision. A few examples are as follows:

- in fault detection and isolation, $\mathbf{x}_\mathrm{c}$ may correspond to the tuning parameters of a bank of Kalman filters and of statistical decision tests, and $\mathbf{x}_\mathrm{e}$ may include parameters describing environmental perturbations and degrees of freedom of the tests on which the performance index is computed,
- in robust control, $\mathbf{x}_\mathrm{c}$ may correspond to the tuning parameters of a controller, and $\mathbf{x}_\mathrm{e}$ may describe uncertainty on the process to be controlled,
- in particle filtering, $\mathbf{x}_\mathrm{c}$ may correspond to parameters describing the strategy for managing the number of particles, whereas $\mathbf{x}_\mathrm{e}$ makes it possible to consider a large family of test cases in order to widen the scope of the resulting tuning,
- in computer-aided design, $\mathbf{x}_\mathrm{c}$ may correspond to design parameters, and $\mathbf{x}_\mathrm{e}$ may describe uncertainty on the value of $\mathbf{x}_\mathrm{c}$ in mass production.

The approaches available for addressing such robust design problems can be classified as stochastic or deterministic. With stochastic approaches, one may optimize with respect to $\mathbf{x}_\mathrm{c}$ the mathematical expectation with respect to $\mathbf{x}_\mathrm{e}$ of some performance index. This requires the availability of the probability distribution of $\mathbf{x}_\mathrm{e}$, and may result in a design that is good on average but unsatisfactory in low-probability regions of the $\mathbf{x}_\mathrm{e}$-space. Minimax (or worst-case) approaches, on the other hand, give equal consideration to all possible values of $\mathbf{x}_\mathrm{e}$. This is the approach considered here, where we want to compute

$$\{\widehat{\mathbf{x}}_\mathrm{c}, \widehat{\mathbf{x}}_\mathrm{e}\} = \arg \min_{\mathbf{x}_\mathrm{c} \in \mathbb{X}_\mathrm{c}} \max_{\mathbf{x}_\mathrm{e} \in \mathbb{X}_\mathrm{e}} J\left(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}\right), \tag{1}$$

with $J(\cdot,\cdot)$ a scalar performance index, $\mathbf{x}_\mathrm{c} \in \mathbb{X}_\mathrm{c}$ a vector of design parameters and $\mathbf{x}_\mathrm{e} \in \mathbb{X}_\mathrm{e}$ a vector of perturbation parameters. $\mathbb{X}_\mathrm{c}$ and $\mathbb{X}_\mathrm{e}$ are assumed to be known compact sets. Any pair $\{\widehat{\mathbf{x}}_\mathrm{c}, \widehat{\mathbf{x}}_\mathrm{e}\}$ such that (1) is satisfied is a minimax (or worst-case) solution of the problem.

Depending on how J is described, different approaches can be considered. Most often, a closed-form expression for J is assumed to be available [1]–[3]. Unfortunately, in real-life complex design problems, this is not the case, and J can only be evaluated numerically through possibly very costly simulations. The methodology developed in this paper is dedicated to this important class of difficult problems.

In this context, the relaxation procedure proposed in [4] is particularly useful. This procedure is generic and does not specify the optimization algorithms to be used. For costly simulations, specific tools are needed. Most of the available techniques use evolutionary algorithms [5], [6], which are known to be computationally expensive and thus inapplicable in our context. An interesting attempt combining the use of a surrogate model with a heuristic optimization strategy has been reported in [7]. In [8], [9], we have proposed MiMaReK, a robust design approach combining Kriging-based optimization, one of the most efficient tools in the context of costly evaluations, with Shimizu and Aiyoshi's relaxation procedure. The new algorithm described in the present paper improves MiMaReK by further reducing the number of evaluations required. The presentation is organized as follows. Section II briefly recalls the original MiMaReK framework. Section III describes the new strategy, which is evaluated and compared to the previous one on test cases in Section IV.

II. Minimax optimization via relaxation and Kriging

A. Relaxation procedure

Equation (1) translates into the following optimization problem with an infinite number of constraints,

$$\begin{cases} \min\limits_{\mathbf{x}_\mathrm{c} \in \mathbb{X}_\mathrm{c}} \tau, \\ \text{subject to } J(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}) \leq \tau, \ \forall \mathbf{x}_\mathrm{e} \in \mathbb{X}_\mathrm{e}. \end{cases} \tag{2}$$

J. Marzat and H. Piet-Lahanier are with ONERA – The French Aerospace Lab, F-91123 Palaiseau, France, firstname.lastname@onera.fr

É. Walter is with the Laboratoire des Signaux et Systèmes (L2S), CNRS-SUPELEC-Univ-Paris-Sud, France, firstname.lastname@lss.supelec.fr

The Shimizu and Aiyoshi procedure (Algorithm 1) relaxes these constraints iteratively to compute an approximate minimax solution with proved convergence to an exact solution when ε_{R} under reasonable technical conditions [4].

Algorithm 1 Minimax optimization via relaxation

1: Pick $\mathbf{x}_{\mathrm{e}}^{(1)} \in \mathbb{X}_{\mathrm{e}}$, and set $\mathcal{R}_{\mathrm{e}} = \left\{\mathbf{x}_{\mathrm{e}}^{(1)}\right\}$ and $i = 1$.
2: Compute $\mathbf{x}_{\mathrm{c}}^{(i)} = \arg\min_{\mathbf{x}_{\mathrm{c}} \in \mathbb{X}_{\mathrm{c}}} \left\{ \max_{\mathbf{x}_{\mathrm{e}} \in \mathcal{R}_{\mathrm{e}}} J(\mathbf{x}_{\mathrm{c}}, \mathbf{x}_{\mathrm{e}}) \right\}$
3: Compute $\mathbf{x}_{\mathrm{e}}^{(i+1)} = \arg\max_{\mathbf{x}_{\mathrm{e}} \in \mathbb{X}_{\mathrm{e}}} J(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e}})$
4: If $J(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e}}^{(i+1)}) - \max_{\mathbf{x}_{\mathrm{e}} \in \mathcal{R}_{\mathrm{e}}} J(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e}}) < \varepsilon_{\mathrm{R}}$ then return $\left\{\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e}}^{(i+1)}\right\}$ as an approximate solution to the initial minimax problem (1).
Else, append $\mathbf{x}_{\mathrm{e}}^{(i+1)}$ to $\mathcal{R}_{\mathrm{e}}$, increment i by 1 and go to Step 2.

Constraint relaxation is achieved at Step 2, where the function to be minimized is the maximum of the performance index over the *finite* set $\mathcal{R}_{\mathrm{e}}$. Steps 2 and 3 leave open the choice of the algorithm to be employed to compute the optima required. We use Kriging-based optimization, which makes it possible to save on the simulation budget.

B. Kriging

Consider a black-box function $f(\mathbf{x})$, known only through numerical evaluations, to be minimized over a known compact set $\mathbb{X}$. Assume that the value of the function has already been evaluated at n points, $\mathcal{X}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and denote by $\mathbf{f}_n = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^{\mathrm{T}}$ the vector of the corresponding function values. *Kriging* makes it possible to predict the value of f over the continuous space $\mathbb{X}$ by modelling it as a zero-mean Gaussian Process $Z(\mathbf{x})$, whose covariance function is expressed as

$$\mathrm{cov}\left(Z(\mathbf{x}_i, \mathbf{x}_j)\right) = \sigma_Z^2 R(\mathbf{x}_i, \mathbf{x}_j), \tag{3}$$

where σ_Z^2 is the process variance and $R(\cdot,\cdot)$ a correlation function, possibly parameterized by a vector $\boldsymbol{\theta}$. In this paper, we use the correlation function

$$R(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(- \sum_{k=1}^{\dim \mathbb{X}} \left| \frac{x_i(k) - x_j(k)}{\theta_k} \right|^2 \right), \tag{4}$$

where $x_i(k)$ is the k-th component of $\mathbf{x}_i$ and the positive coefficients θ_k are scale factors. It should be kept in mind that other correlation functions may be appropriate [10]. For any value of $\mathbf{x} \in \mathbb{X}$, the Kriging prediction is Gaussian and thus entirely characterized by its mean and variance. The mean of the prediction is given by

$$\widehat{f}(\mathbf{x}) = \mathbf{r}\left(\mathbf{x}\right)^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{f}_n, \tag{5}$$

where

$$\begin{aligned} &\mathbf{R}|_{i,j} = R(\mathbf{x}_i, \mathbf{x}_j), \ \{i,j\} = 1, ..., n, \\ &\mathbf{r}(\mathbf{x}) = \left[R(\mathbf{x}_1, \mathbf{x}), ..., R(\mathbf{x}_n, \mathbf{x})\right]^{\mathrm{T}}. \end{aligned} \tag{6}$$

The variance of the prediction is

$$\widehat{\sigma}^2\left(\mathbf{x}\right) = \sigma_Z^2 \left(1 - \mathbf{r}\left(\mathbf{x}\right)^{\mathrm{T}} \mathbf{R}^{-1} \mathbf{r}\left(\mathbf{x}\right)\right). \tag{7}$$

The process variance σ_Z^2 and the vector of parameters $\boldsymbol{\theta}$ of the correlation function (if any) can be estimated, for instance, by maximum likelihood. The fact that the probability distribution of the Kriging prediction is available is an important feature that will be extensively used below.

C. Efficient Global Optimization

Algorithm 2 describes the Efficient Global Optimization (EGO) procedure [11], which exploits the distribution of Kriging prediction to search for a global minimizer of f.

Algorithm 2 Efficient Global Optimization

1: Choose an initial sampling $\mathcal{X}_n = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ in $\mathbb{X}$
2: Compute $\mathbf{f}_n = \left[f\left(\mathbf{x}_1\right), ..., f\left(\mathbf{x}_n\right)\right]^{\mathrm{T}}$
3: **while** $\max_{\mathbf{x} \in \mathbb{X}} \mathrm{EI}(\mathbf{x}) > \varepsilon_{\mathrm{EI}}$ **and** $n < n_{\max}$ **do**
4: Fit the Kriging model on the known data points $\{\mathcal{X}_n, \mathbf{f}_n\}$ with (5)-(7)
5: Find $f_{\min}^n = \min_{i=1,\ldots,n} \{f\left(\mathbf{x}_i\right)\}$
6: Find $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x} \in \mathbb{X}} \mathrm{EI}(\mathbf{x})$
7: Compute $f(\mathbf{x}_{n+1})$, append it to $\mathbf{f}_n$ and append $\mathbf{x}_{n+1}$ to $\mathcal{X}_n$
8: $n \leftarrow n + 1$
9: **end while**

EGO is initialized by sampling n points in $\mathbb{X}$, e.g., with Latin Hypercube Sampling (LHS), and by computing the corresponding values of the function to be minimized. Let $\Phi(z, \mathbf{x})$ be the (Gaussian) cumulative distribution of the Kriging prediction at z, when the vector of parameters takes the value $\mathbf{x}$. The corresponding probability density is given by

$$\varphi(z, \mathbf{x}) \triangleq \frac{\mathrm{d}}{\mathrm{d}z}\left(\Phi(z, \mathbf{x})\right). \tag{8}$$

Define improvement [12] as

$$(f_{\min}^n - z)_+ = \begin{cases} (f_{\min}^n - z) & \text{if positive} \\ 0 & \text{otherwise} \end{cases}, \tag{9}$$

where $f_{\min}^n$ is the smallest value in $\mathbf{f}_n$. The expected value of the improvement (EI) based on the Kriging prediction is defined as

$$\begin{aligned} \mathrm{EI}(\mathbf{x}) &= \int_{-\infty}^{+\infty} (f_{\min}^n - z)_+ \varphi(z, \mathbf{x}) \mathrm{d}z \\ &= \int_{-\infty}^{f_{\min}^n} (f_{\min}^n - z) \varphi(z, \mathbf{x}) \mathrm{d}z, \end{aligned} \tag{10}$$

which can be computed in closed-form using (5) and (7) as

$$\mathrm{EI}\left(\mathbf{x}, f_{\min}^n, \widehat{f}, \widehat{\sigma}\right) = \widehat{\sigma}\left(\mathbf{x}\right)\left[u \Phi_{\mathcal{N}}\left(u\right) + \varphi_{\mathcal{N}}\left(u\right)\right], \tag{11}$$

where $\Phi_{\mathcal{N}}$ is the cumulative distribution function and $\varphi_{\mathcal{N}}$ the probability density function of the normalized Gaussian distribution $\mathcal{N}\left(0, 1\right)$, and where

$$u = \frac{f_{\min}^n - \widehat{f}\left(\mathbf{x}\right)}{\widehat{\sigma}\left(\mathbf{x}\right)}. \tag{12}$$

EGO achieves an iterative search for the global minimum of f and an associated global minimizer. Note that the optimization of EI is carried out at a much lower computational cost than required by the original problem, as no evaluation of the performance index is necessary. By optimizing EI, EGO strikes a compromise between local search (in the neighborhood of the current estimate of the minimizer) and global search (where prediction uncertainty is high). Convergence results are reported in [13], [14].

D. MiMaReK

MiMaReK (for Minimax optimization via relaxation and Kriging) searches for the solution of (1) by combining Algorithms 1 and 2. The resulting procedure, presented in more detail in [8] and [9], is Algorithm 3.

Even if the first version of MiMaReK (called MiMaReK 1 in what follows) turned out to be quite efficient for an economical determination of the solutions of minimax problems, it presents the following drawback. Each iteration of its outer loop requires building two dedicated Kriging predictors from scratch (one for predicting, for a given value of $\mathbf{x}_\mathrm{c}$, the performance index J as a function of $\mathbf{x}_\mathrm{e}$, and the other for predicting the maximum of J over a finite number of values of $\mathbf{x}_\mathrm{e}$, as a function of $\mathbf{x}_\mathrm{c}$). This entails a number of costly evaluations of J, some of which could hopefully be avoided by using a single Kriging predictor for J in the entire procedure, updated whenever a new evaluation of J is carried out.

III. New strategy for saving evaluations

A first simple step towards decreasing the number of evaluations of J is to use a single Kriging predictor for all maximizations of $J(\mathbf{x}_\mathrm{c}^{(i)}, \mathbf{x}_\mathrm{e})$ with respect to $\mathbf{x}_\mathrm{e}$ (Step 3 of Algorithm 3), whatever the value of i. This Kriging predictor is based on all past evaluations of the performance index, and each execution of the outer loop increases the number of its training data.

Using the same Kriging predictor for the minimization of $\max_{\mathbf{x}_\mathrm{e}\in\mathcal{R}_\mathrm{e}} J(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e})$ is significantly more complex. An easy-to-implement idea would be to approximate the mean of this process by

$$\hat{\mu}(\mathbf{x}_\mathrm{c}) = \max_{\mathbf{x}_\mathrm{e}\in\mathcal{R}_\mathrm{e}} \hat{J}(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}) \triangleq \hat{J}(\mathbf{x}_\mathrm{c}, \check{\mathbf{x}}_\mathrm{e}) \tag{13}$$

and its variance by

$$\hat{\sigma}^2(\mathbf{x}_\mathrm{c}) = \hat{\sigma}^2(\mathbf{x}_\mathrm{c}, \check{\mathbf{x}}_\mathrm{e}), \tag{14}$$

with $\hat{J}$ and $\hat{\sigma}^2$ computed by Kriging. It would then become trivial to compute EI as needed by EGO. However, this is a daring approximation, as the mean of the maximum is not the maximum of the means and the distribution of the maximum is not Gaussian. Preliminary tests have confirmed that this approach is not viable.

In the new version of MiMaReK (called MiMaReK 2 in what follows), we instead compute the expected improvement of $\max_{\mathbf{x}_\mathrm{e}\in\mathcal{R}_\mathrm{e}} J(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e})$ exactly.

Let X_i $(i = 1, \ldots, m)$ be m independent random variables, with pdf φ_{X_i} and cumulative distribution function Φ_{X_i} and let

$$Z = \max_i X_i. \tag{15}$$

Z is less than z, if and only if all the X_i's are less than z, so

$$\Phi_Z(z) = \prod_{i=1}^{m} \Phi_{X_i}(z). \tag{16}$$

The pdf of Z is thus

$$\varphi_Z(z) \triangleq \frac{\mathrm{d}}{\mathrm{d}z}\left(\Phi_Z(z)\right) = \sum_{i=1}^{m} \varphi_{X_i}(z) \prod_{j\neq i} \Phi_{X_j}(z). \tag{17}$$

Here, $X_i \sim \mathcal{N}(\hat{J}(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}^i), \hat{\sigma}^2(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}^i))$, where $\mathbf{x}_\mathrm{e}^i$ is the i-th $\mathbf{x}_\mathrm{e}$ vector in $\mathcal{R}_\mathrm{e}$, and $\hat{J}$ and $\hat{\sigma}^2$ are provided by Kriging, so

$$\varphi_{X_i}(z) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}^i)}} \exp\left(-\frac{1}{2}\frac{\left(z - \hat{J}(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}^i)\right)^2}{\hat{\sigma}^2(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}^i)}\right). \tag{18}$$

The values of the vectors $\mathbf{x}_\mathrm{e}^i$ are all known at Step 2, so φ_Z is parametrized by $\mathbf{x}_\mathrm{c}$ only and

$$\varphi_Z(z, \mathbf{x}_\mathrm{c}) = \sum_{i=1}^{m} \varphi_{X_i}(z, \mathbf{x}_\mathrm{c}) \prod_{j\neq i} \Phi_{X_j}(z, \mathbf{x}_\mathrm{c}). \tag{19}$$

For any given $\mathbf{x}_\mathrm{c}$ and z, $\varphi_Z(z, \mathbf{x}_\mathrm{c})$ can be evaluated numerically. It is therefore possible to evaluate the EI for any value of $\mathbf{x}_\mathrm{c}$. Note that the closed-form expression (11) for EI is no longer valid.

The new expression for expected improvement is

$$\mathrm{EI}(\mathbf{x}_\mathrm{c}) = \int_{-\infty}^{f_{\min}^n} (f_{\min}^n - z)\varphi_Z(z, \mathbf{x}_\mathrm{c})\mathrm{d}z. \tag{20}$$

In Algorithm 4, $\mathcal{R}_\mathrm{e}(l)$ stands for the l-th vector in the set $\mathcal{R}_\mathrm{e}$, and $\mathcal{J}_\mathrm{c}(l)$ is the l-th scalar value in the set $\mathcal{J}_\mathrm{c}$. The sets $\mathcal{J}_\mathrm{c}'$ and $\mathcal{X}_\mathrm{c}'$ are temporary, and used to store the data generated at Step 2d. Only the minimum value of the performance index and corresponding argument will be kept and stored in $\mathcal{J}_\mathrm{c}$ and $\mathcal{X}_\mathrm{c}$. This will save evaluations at Step 2b. Optimization at Steps 3(b)ii and 3c is carried out over the values of the performance index such that their $\mathbf{x}_\mathrm{c}$ argument is equal to $\mathbf{x}_\mathrm{c}^{(i)}$. Since EGO has been presented for minimization, the maximization of J carried out at Steps 3(b)ii and 3c is transformed into the minimization of $-J$.

Simplifying hypotheses make it possible to get a rough assessment of how many evaluations may be saved by using MiMaReK 2 rather than MiMaReK 1. Let the total number of initial samples $n_\mathrm{c} + n_\mathrm{e}$ be the same in MiMaReK 1 and MiMaReK 2. Assume that the maximum numbers of iterations (n_EI^c and n_EI^e) are reached during the optimizations by EGO. For N iterations of the outer loop, the required number of evaluations is

$$n_\mathrm{MM1} = N\left(n_\mathrm{c} + n_\mathrm{e} + \frac{N+1}{2} n_\mathrm{EI}^\mathrm{c} + n_\mathrm{EI}^\mathrm{e}\right) \tag{21}$$

Algorithm 3 MiMaReK 1, MiniMax optimization via Relaxation and Kriging Version 1

Set $\varepsilon_{\mathrm{R}}, \varepsilon_{\mathrm{EI}}^{\mathrm{c}}, n_{\mathrm{EI}}^{\mathrm{c}}, \varepsilon_{\mathrm{EI}}^{\mathrm{e}}, n_{\mathrm{EI}}^{\mathrm{e}}, n_{\mathrm{c}}, n_{\mathrm{e}}$.

1) **Step 1**

a) Choose randomly $\mathbf{x}_{\mathrm{e}}^{(1)}$ in $\mathbb{X}_{\mathrm{e}}$. Initialize $\mathcal{R}_{\mathrm{e}} = \left\{\mathbf{x}_{\mathrm{e}}^{(1)}\right\}$. Set $i \leftarrow 1$.

b) Choose a design $\mathcal{X}_0^{\mathrm{c}} = \{\mathbf{x}_{\mathrm{c},1}, ..., \mathbf{x}_{\mathrm{c},n_{\mathrm{c}}}\}$ in $\mathbb{X}_{\mathrm{c}}$.

c) Choose a design $\mathcal{X}_0^{\mathrm{e}} = \{\mathbf{x}_{\mathrm{e},1}, ..., \mathbf{x}_{\mathrm{e},n_{\mathrm{e}}}\}$ in $\mathbb{X}_{\mathrm{e}}$.

while $e > \varepsilon_{\mathrm{R}}$

2) **Step 2**

a) Initialize $j \leftarrow n_{\mathrm{c}}$ and $\mathcal{X}_j^{\mathrm{c}} = \mathcal{X}_0^{\mathrm{c}}$.

b) Compute $\mathbf{J}_j^{\mathrm{c}} = \left\{ \max_{\mathbf{x}_{\mathrm{e}} \in \mathcal{R}_{\mathrm{e}}} \left\{J\left(\mathbf{x}_{\mathrm{c},1}, \mathbf{x}_{\mathrm{e}}\right)\right\}, ..., \max_{\mathbf{x}_{\mathrm{e}} \in \mathcal{R}_{\mathrm{e}}} \left\{J\left(\mathbf{x}_{\mathrm{c},n_{\mathrm{c}}}, \mathbf{x}_{\mathrm{e}}\right)\right\} \right\}$.

c) **while** $\max_{\mathbf{x}_{\mathrm{c}} \in \mathbb{X}_{\mathrm{c}}} \{\mathrm{EI}(\mathbf{x}_{\mathrm{c}})\} > \varepsilon_{\mathrm{EI}}^{\mathrm{c}}$ **and** $j < n_{\mathrm{EI}}^{\mathrm{c}}$

i) Fit a Kriging model on the known data points $\left\{\mathcal{X}_j^{\mathrm{c}}, \mathbf{J}_j^{\mathrm{c}}\right\}$.

ii) Find $J_{\min}^j = \min_{1...j} \left\{\mathbf{J}_j^{\mathrm{c}}\right\}$.

iii) Find the next point of interest $\mathbf{x}_{\mathrm{c},j+1}$ by maximizing $\mathrm{EI}(\mathbf{x}_{\mathrm{c}})$

iv) Append $\mathbf{x}_{\mathrm{c},j+1}$ to $\mathcal{X}_j^{\mathrm{c}}$.

v) Find $\max_{\mathbf{x}_{\mathrm{e}} \in \mathcal{R}_{\mathrm{e}}} \left\{J\left(\mathbf{x}_{\mathrm{c},j+1}, \mathbf{x}_{\mathrm{e}}\right)\right\}$ and append it to $\mathbf{J}_j^{\mathrm{c}}$.

vi) $j \leftarrow j + 1$.

end while

d) Find $\mathbf{x}_{\mathrm{c}}^{(i)} = \arg \min_{\mathbf{x}_{\mathrm{c}} \in \mathcal{X}_j^{\mathrm{c}}} \left\{\mathbf{J}_j^{\mathrm{c}}\right\}$

e) Compute $e_{\mathrm{prec}} = \max_{\mathbf{x}_{\mathrm{e}} \in \mathcal{R}_{\mathrm{e}}} J\left(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e}}\right)$

3) **Step 3**

a) Initialize $k \leftarrow n_{\mathrm{e}}$ and $\mathcal{X}_k^{\mathrm{e}} = \mathcal{X}_0^{\mathrm{e}}$.

b) Compute $\mathbf{J}_k^{\mathrm{e}} = \left\{-J\left(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e},1}\right), ..., -J\left(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e},n_{\mathrm{e}}}\right)\right\}$.

c) **while** $\max_{\mathbf{x}_{\mathrm{e}} \in \mathbb{X}_{\mathrm{e}}} \{\mathrm{EI}(\mathbf{x}_{\mathrm{e}})\} > \varepsilon_{\mathrm{EI}}^{\mathrm{e}}$ **and** $k < n_{\mathrm{EI}}^{\mathrm{e}}$

i) Fit a Kriging model on the known data points $\{\mathcal{X}_k^{\mathrm{e}}, \mathbf{J}_k^{\mathrm{e}}\}$.

ii) Find $J_{\max}^k = \min_{1...k} \{\mathbf{J}_k^{\mathrm{e}}\}$.

iii) Find the next point of interest $\mathbf{x}_{\mathrm{e},k+1}$ by maximizing $\mathrm{EI}(\mathbf{x}_{\mathrm{e}})$

iv) Append $\mathbf{x}_{\mathrm{e},k+1}$ to $\mathcal{X}_k^{\mathrm{e}}$.

v) Compute $-J\left(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e},k+1}\right)$ and append it to $\mathbf{J}_k^{\mathrm{e}}$.

vi) $k \leftarrow k + 1$.

end while

d) Find $\mathbf{x}_{\mathrm{e}}^{(i+1)} = \arg \min_{\mathbf{x}_{\mathrm{e}} \in \mathcal{X}_k^{\mathrm{e}}} \{\mathbf{J}_k^{\mathrm{e}}\}$ and append it to $\mathcal{R}_{\mathrm{e}}$

4) **Step 4**

a) Compute $e = J\left(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e}}^{(i+1)}\right) - e_{\mathrm{prec}}$

b) $i \leftarrow i + 1$

end while

for MiMaReK 1 and

$$n_{\mathrm{MM2}} = n_{\mathrm{c}} + n_{\mathrm{e}} + N\left(\frac{N+1}{2}\left(n_{\mathrm{EI}}^{\mathrm{c}} + 1\right) + n_{\mathrm{EI}}^{\mathrm{e}}\right) \quad (22)$$

for MiMaReK 2. Thus

$$n_{\mathrm{MM1}} - n_{\mathrm{MM2}} = (N-1)\left(n_{\mathrm{c}} + n_{\mathrm{e}}\right) - \frac{N\left(N+1\right)}{2}, \quad (23)$$

which means that MiMaReK 2 requires less evaluations than MiMaReK 1 if, for $N > 1$,

$$n_{\mathrm{c}} + n_{\mathrm{e}} > \frac{N\left(N+1\right)}{2\left(N-1\right)}. \quad (24)$$

This inequality will usually be verified, as can be seen in the examples of the next Section. A large number of iterations are indeed required to make the right-hand side larger than the total number of initial samples $n_{\mathrm{c}} + n_{\mathrm{e}}$.

IV. Comparison on test cases

In this section, we evaluate and compare the performances of MiMaReK 1 and MiMaReK 2 on six test cases. As these test cases have also been used in [5], [6] and [7], this facilitates comparisons with alternative approaches. The first

Algorithm 4 MiMaReK 2, MiniMax optimization via Relaxation and Kriging Version 2

Set $\varepsilon_{\mathrm{R}}, \varepsilon_{\mathrm{EI}}^{\mathrm{c}}, n_{\mathrm{EI}}^{\mathrm{c}}, \varepsilon_{\mathrm{EI}}^{\mathrm{e}}, n_{\mathrm{EI}}^{\mathrm{e}}, n$

1) **Step 1**
   a) Choose a design $\mathcal{X} = \left\{[\mathbf{x}_{\mathrm{c},1}^{\mathrm{T}}, \mathbf{x}_{\mathrm{e},1}^{\mathrm{T}}]^{\mathrm{T}}, ..., [\mathbf{x}_{\mathrm{c},n}^{\mathrm{T}}, \mathbf{x}_{\mathrm{e},n}^{\mathrm{T}}]^{\mathrm{T}}\right\}$ in $\mathbb{X} = \mathbb{X}_{\mathrm{c}} \times \mathbb{X}_{\mathrm{e}}$
   b) Compute $\mathbf{J}_n = [J(\mathbf{x}_{\mathrm{c},1}, \mathbf{x}_{\mathrm{e},1}), ..., J(\mathbf{x}_{\mathrm{c},n}, \mathbf{x}_{\mathrm{e},n})]^{\mathrm{T}}$
   c) Choose randomly an index $i_0 \in [1, ..., n]$
   d) Initialize $\mathbf{x}_{\mathrm{e}}^{(1)} = \mathbf{x}_{\mathrm{e},i_0}$, $\mathcal{R}_{\mathrm{e}} = \left\{\mathbf{x}_{\mathrm{e}}^{(1)}\right\}$, $\mathcal{X}_{\mathrm{c}} = \{\mathbf{x}_{\mathrm{c},i_0}\}$ and $\mathcal{J}_{\mathrm{c}} = \{J(\mathbf{x}_{\mathrm{c},i_0}, \mathbf{x}_{\mathrm{e},i_0})\}$
   e) Set $i \leftarrow 1$

while $e > \varepsilon_{\mathrm{R}}$

2) **Step 2**
   a) Initialize $j \leftarrow 0$
   b) **for** $l = 1$ to $\mathrm{card}(\mathcal{X}_{\mathrm{c}})$ **do**
   $\mathcal{J}_{\mathrm{c}}(l) = \max\left(\mathcal{J}_{\mathrm{c}}(l), J(\mathcal{X}_{\mathrm{c}}(l), \mathbf{x}_{\mathrm{e}}^{(i)})\right)$.
   end for
   c) Set $\mathcal{J}_{\mathrm{c}}' = \mathcal{J}_{\mathrm{c}}$ and $\mathcal{X}_{\mathrm{c}}' = \mathcal{X}_{\mathrm{c}}$
   d) **while** $\max\limits_{\mathbf{x}_{\mathrm{c}} \in \mathbb{X}_{\mathrm{c}}} \{\mathrm{EI}(\mathbf{x}_{\mathrm{c}})\} > \varepsilon_{\mathrm{EI}}^{\mathrm{c}}$ **and** $j < n_{\mathrm{EI}}^{\mathrm{c}}$
      i) Fit a Kriging model on the data $\{\mathcal{X}, \mathbf{J}_n\}$.
      ii) Find $J_{\min}^{j} = \min \mathcal{J}_{\mathrm{c}}'$
      iii) Find the next point of interest $\mathbf{x}_{\mathrm{c},j+1} = \arg\max\limits_{\mathbf{x}_{\mathrm{c}} \in \mathbb{X}_{\mathrm{c}}} \mathrm{EI}(\mathbf{x}_{\mathrm{c}})$ with (20)
      iv) Append $[\mathbf{x}_{\mathrm{c},j+1}^{\mathrm{T}}, \mathcal{R}_{\mathrm{e}}(1)^{\mathrm{T}}]^{\mathrm{T}}, ..., [\mathbf{x}_{\mathrm{c},j+1}^{\mathrm{T}}, \mathcal{R}_{\mathrm{e}}(i)^{\mathrm{T}}]^{\mathrm{T}}$ to the design $\mathcal{X}$
      v) Compute $J(\mathbf{x}_{\mathrm{c},j+1}, \mathcal{R}_{\mathrm{e}}(1)), ..., J(\mathbf{x}_{\mathrm{c},j+1}, \mathcal{R}_{\mathrm{e}}(i))$ and append them to the performance vector $\mathbf{J}_n$
      vi) $n \leftarrow n + i$
      vii) Compute $\max\{J(\mathbf{x}_{\mathrm{c},j+1}, \mathcal{R}_{\mathrm{e}}(1)), ..., J(\mathbf{x}_{\mathrm{c},j+1}, \mathcal{R}_{\mathrm{e}}(i))\}$ and append it to $\mathcal{J}_{\mathrm{c}}'$
      viii) Append $\mathbf{x}_{\mathrm{c},j+1}$ to $\mathcal{X}_{\mathrm{c}}'$
      ix) $j \leftarrow j + 1$
      end while
   e) Compute $e_{\mathrm{prec}} = \min\{\mathcal{J}_{\mathrm{c}}'\}$ and append it to $\mathcal{J}_{\mathrm{c}}$
   f) Append $\mathbf{x}_{\mathrm{c}}^{(i)} = \arg\min\{\mathcal{J}_{\mathrm{c}}'\}$ to $\mathcal{X}_{\mathrm{c}}$

3) **Step 3**
   a) Initialize $k \leftarrow 0$
   b) **while** $\max\limits_{\mathbf{x}_{\mathrm{e}} \in \mathbb{X}_{\mathrm{e}}} \{\mathrm{EI}(\mathbf{x}_{\mathrm{e}})\} > \varepsilon_{\mathrm{EI}}^{\mathrm{e}}$ **and** $k < n_{\mathrm{EI}}^{\mathrm{e}}$
      i) Fit a Kriging model on the data $\{\mathcal{X}, \mathbf{J}_n\}$
      ii) Find $J_{\max}^{k} = \min\limits_{\mathbf{x}_{\mathrm{c}} = \mathbf{x}_{\mathrm{c}}^{(i)}} \{-\mathbf{J}_n\}$
      iii) Find the next point of interest $\mathbf{x}_{\mathrm{e},k+1} = \arg\max\limits_{\mathbf{x}_{\mathrm{e}} \in \mathbb{X}_{\mathrm{e}}} \mathrm{EI}(\mathbf{x}_{\mathrm{e}})$
      iv) Append $[\mathbf{x}_{\mathrm{c}}^{(i)\mathrm{T}}, \mathbf{x}_{\mathrm{e},k+1}^{\mathrm{T}}]^{\mathrm{T}}$ to $\mathcal{X}$
      v) Compute $J\left(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e},k+1}\right)$ and append it to $\mathbf{J}_n$
      vi) $n \leftarrow n + 1$
      vii) $k \leftarrow k + 1$
      end while
   c) Find $\mathbf{x}_{\mathrm{e}}^{(i+1)} = \arg\min\limits_{\mathbf{x}_{\mathrm{c}} = \mathbf{x}_{\mathrm{c}}^{(i)}} \{-\mathbf{J}_n\}$ and append it to $\mathcal{R}_{\mathrm{e}}$

4) **Step 4**
   a) Compute $e = J\left(\mathbf{x}_{\mathrm{c}}^{(i)}, \mathbf{x}_{\mathrm{e}}^{(i+1)}\right) - e_{\mathrm{prec}}$
   b) $i \leftarrow i + 1$

end while

TABLE I
RESULTS FOR THE TESTS FUNCTIONS WITH MIMAREK 1
(OBTAINED FROM 50 RUNS)

Function	MiMaReK 1			
	Percentage of deviation from theoretical optimum		Number of evaluations of the performance index	
	Mean	Std. dev.	Mean	Std. dev.
f_1	0	0	52	1
f_2	0.17	$7.4 \cdot 10^{-3}$	270	68
f_3	0.12	$4 \cdot 10^{-4}$	281	72
f_4	3.5	0.02	279	89
f_5	0.76	0.01	94	4
f_6	2.51	0.23	223	89

TABLE II
RESULTS FOR THE TESTS FUNCTIONS WITH MIMAREK 2
(OBTAINED FROM 50 RUNS)

Function	MiMaReK 2			
	Percentage of deviation from theoretical optimum		Number of evaluations of the performance index	
	Mean	Std. dev.	Mean	Std. dev.
f_1	$7.1 \cdot 10^{-3}$	$8 \cdot 10^{-5}$	35	3
f_2	0.06	$1 \cdot 10^{-3}$	98	29
f_3	0.88	$1.6 \cdot 10^{-3}$	189	38
f_4	2.9	0.04	174	12
f_5	0.99	$5 \cdot 10^{-3}$	58	4
f_6	0.4	0.01	101	11

four test functions have scalar arguments

$$f_1(x_\mathrm{c}, x_\mathrm{e}) = (x_\mathrm{c} - 5)^2 - (x_\mathrm{e} - 5^2),$$
$$f_2(x_\mathrm{c}, x_\mathrm{e}) = \min\{3 - 0.2x_\mathrm{c} + 0.3x_\mathrm{e}, 3 + 0.2x_\mathrm{c} - 0.1x_\mathrm{e}\},$$
$$f_3(x_\mathrm{c}, x_\mathrm{e}) = \frac{\sin(x_\mathrm{c} - x_\mathrm{e})}{\sqrt{x_\mathrm{c}^2 + x_\mathrm{e}^2}},\ f_4(x_\mathrm{c}, x_\mathrm{e}) = \frac{\cos\left(\sqrt{x_\mathrm{c}^2 + x_\mathrm{e}^2}\right)}{\sqrt{x_\mathrm{c}^2 + x_\mathrm{e}^2} + 10},$$

while the last two have two-dimensional vector arguments

$$\begin{aligned} f_5(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}) &= 100(x_{\mathrm{c}2} - x_{\mathrm{c}1}^2)^2 + (1 - x_{\mathrm{c}1})^2 \\ &\quad - x_{\mathrm{e}1}(x_{\mathrm{c}1} + x_{\mathrm{c}2}^2) - x_{\mathrm{e}2}(x_{\mathrm{c}1}^2 + x_{\mathrm{c}2}), \\ f_6(\mathbf{x}_\mathrm{c}, \mathbf{x}_\mathrm{e}) &= (x_{\mathrm{c}1} - 2)^2 + (x_{\mathrm{c}2} - 1)^2 + x_{\mathrm{e}1}(x_{\mathrm{c}1}^2 - x_{\mathrm{c}2}) \\ &\quad + x_{\mathrm{e}2}(x_{\mathrm{c}1} + x_{\mathrm{c}2} - 2). \end{aligned}$$

For each of the test cases and both versions of MiMaReK, the following applies:

- the selection of the n initial sample points is carried out by LHS, with the usual rule of thumb $n = 10 \times \dim \mathbb{X}$,
- maximization of the EI criterion is performed by the DIRECT optimization procedure, as recommended in [15],
- the thresholds $(\varepsilon_\mathrm{R}, \varepsilon_\mathrm{EI}^\mathrm{c}, \varepsilon_\mathrm{EI}^\mathrm{e})$ are set to 10^{-3}, and the maximum numbers of iterations n_EI^c and n_EI^e are set to $20 \times \dim \mathbb{X}_\mathrm{c}$ and $20 \times \dim \mathbb{X}_\mathrm{e}$ respectively.

For each version of MiMaReK, Tables I and II give the percentage of deviation of the value of the minimax performance index $f_i(\widehat{\mathbf{x}}_\mathrm{c}, \widehat{\mathbf{x}}_\mathrm{e})$ from its true value and the number of evaluations of f_i $(i = 1, ..., 6)$. The mean (and standard deviation) of these results have been obtained by averaging 50 runs for each function. The number of evaluations performed by MiMaReK 2 is always significantly smaller (and sometime very significantly smaller) than that of MiMaReK 1, and condition (24) is always satisfied. The estimated values of $f_i(\widehat{\mathbf{x}}_\mathrm{c}, \widehat{\mathbf{x}}_\mathrm{e})$ are always close to one another and to the actual value.

In [5] and [6], between 10^4 and 10^5 evaluations of the functions were required to achieve similar performance. In [7], the authors set the number of evaluations of the performance index *a priori* to 110 for each of the six test-cases, which did not allow them to obtain a suitable solution for f_6.

V. CONCLUSIONS AND PERSPECTIVES

In this paper, a new strategy for further reducing the number of evaluations of the performance index of a worst-case design problem has been presented. On reference test cases of the literature, it has been shown to be very effective. More complex, real-life design problems are currently being investigated.

REFERENCES

[1] D. Du and P. M. Pardalos, *Minimax and Applications*. Kluwer Academic Publishers, Norwell, 1995.

[2] B. Rustem and M. Howe, *Algorithms for Worst-Case Design and Applications to Risk Management*. Princeton University Press, 2002.

[3] P. Parpas and B. Rustem, "An algorithm for the global optimization of a class of continuous minimax problems," *Journal of Optimization Theory and Applications*, vol. 141, no. 2, pp. 461–473, 2009.

[4] K. Shimizu and E. Aiyoshi, "Necessary conditions for min-max problems and algorithms by a relaxation procedure," *IEEE Transactions on Automatic Control*, vol. 25, no. 1, pp. 62–66, 1980.

[5] A. M. Cramer, S. D. Sudhoff, and E. L. Zivi, "Evolutionary algorithms for minimax problems in robust design," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 444–453, 2009.

[6] R. I. Lung and D. Dumitrescu, "A new evolutionary approach to minimax problems," in *Proceedings of the 2011 IEEE Congress on Evolutionary Computation, New Orleans, USA*, 2011, pp. 1902–1905.

[7] A. Zhou and Q. Zhang, "A surrogate-assisted evolutionary algorithm for minimax optimization," in *Proceedings of the 2010 IEEE Congress on Evolutionary Computation, Barcelona, Spain*, 2010, pp. 1–7.

[8] J. Marzat, E. Walter, and H. Piet-Lahanier, "Min-max hyperparameter tuning with application to fault detection," in *Proceedings of the 18th IFAC World Congress, Milan, Italy*, 2011, pp. 12 904–12 909.

[9] J. Marzat, E. Walter, F. Damongeot, and H. Piet-Lahanier, "Robust automatic tuning of diagnosis methods via an efficient use of costly simulations," in *Proceedings of the 16th IFAC Symposium on System Identification, Brussels, Belgium*, 2012, pp. 398–403.

[10] T. J. Santner, B. J. Williams, and W. Notz, *The Design and Analysis of Computer Experiments*. Springer-Verlag, Berlin-Heidelberg, 2003.

[11] D. R. Jones, M. J. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[12] M. Schonlau, *Computer Experiments and Global Optimization*. PhD Thesis, University of Waterloo, Canada, 1997.

[13] E. Vazquez and J. Bect, "Convergence properties of the expected improvement algorithm with fixed mean and covariance functions," *Journal of Statistical Planning and Inference*, vol. 140, no. 11, pp. 3088–3095, 2010.

[14] A. D. Bull, "Convergence rates of efficient global optimization algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2879–2904, 2011.

[15] M. J. Sasena, *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD Thesis, University of Michigan, USA, 2002.