

A new expected-improvement algorithm for continuous minimax optimization

Julien Marzat, Eric Walter, H el ene Piet-Lahanier

► To cite this version:

Julien Marzat, Eric Walter, H el ene Piet-Lahanier. A new expected-improvement algorithm for continuous minimax optimization. *Journal of Global Optimization*, Springer Verlag, 2016, 64 (4), pp.785-802. 10.1007/s10898-015-0344-x . hal-01183567

HAL Id: hal-01183567

<https://hal.archives-ouvertes.fr/hal-01183567>

Submitted on 17 Aug 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destin ee au d ep ot et  a la diffusion de documents scientifiques de niveau recherche, publi es ou non,  emanant des  tablissements d'enseignement et de recherche fran ais ou  trangers, des laboratoires publics ou priv es.

A new expected-improvement algorithm for continuous minimax optimization

Julien Marzat · Eric Walter ·
Hélène Piet-Lahanier

Received: date / Accepted: date

Abstract Worst-case design is important whenever robustness to adverse environmental conditions must be ensured regardless of their probability. It leads to minimax optimization, which is most often treated assuming that prior knowledge makes the worst environmental conditions obvious, or that a closed-form expression for the performance index is available. This paper considers the important situation where none of these assumptions is true and where the performance index must be evaluated via costly numerical simulations. Strategies to limit the number of these evaluations are then of paramount importance. One such strategy is proposed here, which further improves the performance of an algorithm recently presented that combines a relaxation procedure for minimax search with the well-known Kriging-based EGO algorithm. Expected Improvement is computed in the minimax optimization context, which allows to further reduce the number of costly evaluations of the performance index. The interest of the approach is demonstrated on test cases and a simple engineering problem from the literature, which facilitates comparison with alternative approaches.

Keywords continuous minimax · EGO · Expected Improvement · Kriging · robust optimization · worst-case analysis

1 Introduction

Robust optimization [1–4] looks for a decision that is optimal with respect to some performance index while taking into account uncontrolled sources of variation. For a wide class of problems, a design vector \mathbf{x}_c must be tuned to achieve the best

J. Marzat · H. Piet-Lahanier
ONERA – The French Aerospace Lab, F-91123 Palaiseau, France
Tel.: +33 1 80 38 66 50
E-mail: julien.marzat@onera.fr, helene.piet-lahanier@onera.fr

E. Walter
L2S, CNRS-SUPELEC-Univ Paris-Sud, F-91192 Gif-sur-Yvette, France
E-mail: eric.walter@centralesupelec.fr

performance possible while protecting oneself against the potentially adverse effects of an environmental vector \mathbf{x}_e . Such problems are important in the context of engineering design, robust control or estimation. In computer-aided design, for instance, \mathbf{x}_c may correspond to design parameters, and \mathbf{x}_e may describe uncertainty on the value of \mathbf{x}_c in mass production. In robust control, \mathbf{x}_c may correspond to the tuning parameters of a controller, and \mathbf{x}_e may describe uncertainty on the process to be controlled. In fault detection and isolation, \mathbf{x}_c may correspond to the tuning parameters of a bank of Kalman filters and statistical decision tests, and \mathbf{x}_e may include parameters describing environmental perturbations and degrees of freedom of the tests on which some performance index is computed.

The approaches available for addressing such robust design problems can be classified as stochastic or deterministic [5]. With stochastic approaches [6], one may search for \mathbf{x}_c that optimizes the mathematical expectation with respect to \mathbf{x}_e of some performance index [7,8]. This requires the availability of the probability distribution of \mathbf{x}_e , and may result in a design that is good on average but unsatisfactory in low-probability regions of the \mathbf{x}_e -space. Minimax (or worst-case) approaches [9], on the other hand, give equal consideration to all possible values of \mathbf{x}_e . This is the strategy considered here, where we look for

$$\{\widehat{\mathbf{x}}_c, \widehat{\mathbf{x}}_e\} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} J(\mathbf{x}_c, \mathbf{x}_e), \quad (1)$$

with $J(\cdot, \cdot)$ a scalar performance index, $\mathbf{x}_c \in \mathbb{X}_c$ a vector of design parameters and $\mathbf{x}_e \in \mathbb{X}_e$ a vector of perturbation parameters. $\mathbb{X}_c \subset \mathbb{R}^{d_c}$ and $\mathbb{X}_e \subset \mathbb{R}^{d_e}$ are assumed to be known compact sets. Sometimes, the value of the worst vector \mathbf{x}_e in \mathbb{X}_e is easy to guess based on prior problem knowledge. If, for instance, we are trying to minimize the worst-case injury in an automotive crash simulation, common sense tells us that the worst case will be when the speed of the vehicle is at its highest possible value. In such a case, solving (1) boils down to a classical minimization problem where the environmental variables take their worst-case values. This is not the situation considered here. It is assumed instead that the worst-case values of the environmental variables may depend on the settings of the design variables, in a way that is not intuitively obvious (see, for instance, the vibration absorber example of Section 4.2). This paper is thus devoted to the approximate computation of a minimax solution (1) for problems with the four following characteristics:

- the worst-case value of the environmental vector cannot be deduced from prior problem knowledge and may depend on the value taken by the design vector,
- the feasible sets \mathbb{X}_c and \mathbb{X}_e are continuous,
- there is no closed-form expression for $J(\mathbf{x}_c, \mathbf{x}_e)$,
- $J(\mathbf{x}_c, \mathbf{x}_e)$ can only be evaluated through possibly very costly numerical simulations.

Section 2 recalls the state of the art in this context and in particular a Kriging-based strategy that we presented in [10]. Section 3 presents a new strategy to further decrease the number of evaluations of the performance index. In Section 4, these two approaches are compared on six test functions of the literature and one classical benchmark problem in mechanics (a vibration damper).

2 Existing tools for continuous minimax optimization of black-box functions

There are two brute-force approaches one may think of. The first is nested optimization, where each evaluation of the cost function for the outer minimization is carried out by an inner maximization. The second one evaluates performance for each design variable setting at a *very large* discrete set \mathcal{R}_e of points in the continuous set of environmental variables, and takes the worst value found. With both approaches one must expend a lot of effort to find the worst-case performance for each setting of the design variables. These approaches are thus inappropriate when dealing with expensive simulations. Computationally-intensive techniques based on evolutionary algorithms [11, 12] are inapplicable for the same reason. A number of less expensive methods [9, 13, 14] require instead a closed-form expression for the performance index, which is assumed unavailable here.

The fact that the worst-case value of the environmental vector is not obvious makes the procedure proposed in [15] particularly adequate. In this procedure, a *small* discrete set \mathcal{R}_e of points in the space of the environmental variables, which are associated with a high value of the performance index, is updated. Replacing a maximization over the continuous set \mathbb{X}_e by a much easier one over the discrete set \mathcal{R}_e is a relaxation of the original problem. This relaxation procedure is generic and does not specify the optimization algorithms to be used.

The cost involved in the evaluation of the performance index suggests using an optimization strategy based on surrogate modeling such as EGO [16], a well-established method for expensive black-box functions with no derivatives. The origins of EGO can be traced back to [17, 18], see [19] for an historical account and more details. However, EGO cannot handle minimax optimization. A natural approach for dealing with all four characteristics of the problems considered is thus to use the procedure of [15] with optimizations performed by EGO. This was done in [10], where the algorithm MiMaReK (called MiMaReK 1 in what follows) was introduced.

To facilitate the understanding of the new results presented in Section 3 and make the paper self-contained, we now summarize the relaxation procedure for minimax optimization described in [15] and EGO before recalling the main features of MiMaReK 1.

2.1 Relaxation procedure for minimax optimization

Minimax optimization searches for the vector \mathbf{x}_c of design variable settings that minimizes the worst-case performance $\max_{\mathbf{x}_e \in \mathbb{X}_e} J(\mathbf{x}_c, \mathbf{x}_e)$. This problem can be relaxed by performing the maximization on a finite set of points \mathcal{R}_e of \mathbb{X}_e instead of carrying out the maximization over all possible \mathbf{x}_e in \mathbb{X}_e . By a suitable increase of the size of the set \mathcal{R}_e , the approximate problem will gradually become equivalent to the original problem. The basic idea of the algorithm is to start with an \mathcal{R}_e consisting of just one randomly chosen point \mathbf{x}_e^* in the space of the environmental variables. The approximate problem is then solved to come up with the best set of design variables \mathbf{x}_c^* and the corresponding minimax value $J^* = J(\mathbf{x}_c^*, \mathbf{x}_e^*)$ by *minimizing* $J(\mathbf{x}_c, \mathbf{x}_e^*)$ over \mathbb{X}_c . Given \mathbf{x}_c^* , the next step searches for a point \mathbf{x}_e^* in the environmental space that makes the performance as bad as possible, by *maximizing*

$J(\mathbf{x}_c^*, \mathbf{x}_e)$ over \mathbb{X}_e . If this reverse optimization in the entire environmental space does not worsen performance too much, that is, if $J(\mathbf{x}_c^*, \mathbf{x}_e)^* - J^* < \varepsilon_R$ for some positive threshold $\varepsilon_R > 0$, then \mathbf{x}_c^* is considered as a good enough approximation of a minimax solution and the algorithm is stopped. Otherwise, the point \mathbf{x}_e^* is added to the set \mathcal{R}_e and the procedure is iterated. This strategy, proposed in [15], is summarized as Algorithm 1. Under reasonable technical conditions, it has been proven to converge to an exact solution when $\varepsilon_R \rightarrow 0$.

Algorithm 1 Minimax optimization via relaxation [15]

Set ε_R .

1. Pick $\mathbf{x}_e^* \in \mathbb{X}_e$, and set $\mathcal{R}_e = \{\mathbf{x}_e^*\}$.
 2. Compute $\mathbf{x}_c^* = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \left\{ \max_{\mathbf{x}_e \in \mathcal{R}_e} J(\mathbf{x}_c, \mathbf{x}_e) \right\}$ and $J^* = \max_{\mathbf{x}_e \in \mathcal{R}_e} J(\mathbf{x}_c^*, \mathbf{x}_e)$.
 3. Compute $\mathbf{x}_e^* = \arg \max_{\mathbf{x}_e \in \mathbb{X}_e} J(\mathbf{x}_c^*, \mathbf{x}_e)$.
 4. If $J(\mathbf{x}_c^*, \mathbf{x}_e^*) - J^* < \varepsilon_R$ then return $\{\mathbf{x}_c^*, \mathbf{x}_e^*\}$ as an approximate solution to the initial minimax problem (1).
Else, append \mathbf{x}_e^* to \mathcal{R}_e and go to Step 2.
-

Note that each new iteration is computationally more expensive than the previous one, because evaluating the worst-case performance at a trial point \mathbf{x}_c requires evaluating $J(\mathbf{x}_c, \mathbf{x}_e)$ for all \mathbf{x}_e in \mathcal{R}_e . On the examples treated so far, it turned out that only a few iterations of the outer loop are necessary before the convergence criterion is satisfied.

Steps 2 and 3 leave open the choice of the algorithm to be employed to compute the optimizers required. In MiMaReK, both optimizations – first to find \mathbf{x}_c^* and then to find \mathbf{x}_e^* – are carried out via the response-surface-based EGO algorithm, which makes it possible to limit the simulation budget.

2.2 EGO

EGO (the acronym of Efficient Global Optimization [16]) exploits the distribution of Kriging prediction [20] to search for a global minimizer $\hat{\mathbf{x}} \in \mathbb{X}$ of a cost function $f(\mathbf{x})$, known only through numerical evaluations. Assume that the value of the function has already been evaluated at n points, $\mathcal{X}_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and denote by $\mathbf{f}_n = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^T$ the vector of the corresponding function values. In Kriging, the value of f over \mathbb{X} is predicted by modeling it as the Gaussian process (GP)

$$F(\mathbf{x}) = \mathbf{p}(\mathbf{x})^T \mathbf{b} + Z(\mathbf{x}). \quad (2)$$

In this model, $\mathbf{p}(\mathbf{x})$ is some known vector of regressors (usually chosen constant or polynomial in \mathbf{x}) and \mathbf{b} is a vector of unknown regression coefficients to be estimated, e.g. by maximum likelihood. $Z(\mathbf{x})$ is a zero-mean Gaussian Process whose covariance function is expressed as

$$\text{cov}(Z(\mathbf{x}_i, \mathbf{x}_j)) = \sigma_Z^2 R(\mathbf{x}_i, \mathbf{x}_j), \quad (3)$$

with σ_Z^2 the process variance and $R(\cdot, \cdot)$ a correlation function, possibly parameterized by a vector $\boldsymbol{\theta}$. For any given $\mathbf{x} \in \mathbb{X}$, the Kriging prediction is Gaussian and thus entirely characterized by its mean and variance. The mean of the prediction is given by

$$\hat{f}(\mathbf{x}) = \mathbf{p}(\mathbf{x})^T \mathbf{b} + \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1}(\mathbf{f}_n - \mathbf{P}\mathbf{b}), \quad (4)$$

where

$$\begin{aligned} \mathbf{R}_{|i,j} &= R(\mathbf{x}_i, \mathbf{x}_j), \quad \{i, j\} = 1, \dots, n, \\ \mathbf{r}(\mathbf{x}) &= [R(\mathbf{x}_1, \mathbf{x}), \dots, R(\mathbf{x}_n, \mathbf{x})]^T \\ \mathbf{P} &= [\mathbf{p}(\mathbf{x}_1), \dots, \mathbf{p}(\mathbf{x}_n)]^T. \end{aligned} \quad (5)$$

The variance of the prediction is

$$\hat{\sigma}^2(\mathbf{x}) = \sigma_Z^2 \left(1 - \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1} \mathbf{r}(\mathbf{x})\right). \quad (6)$$

In this paper, the following correlation function is used

$$R(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(- \sum_{k=1}^{\dim \mathbb{X}} \left| \frac{x_i(k) - x_j(k)}{\theta_k} \right|^2 \right), \quad (7)$$

where $x_i(k)$ is the k -th component of \mathbf{x}_i and the positive coefficients θ_k are scale factors. Other correlation functions may be employed [21]. 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 [22].

EGO (Algorithm 2) is initialized by sampling n points in \mathbb{X} , e.g., with Latin Hypercube Sampling (LHS) [23], and 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

$$\varphi(z, \mathbf{x}) \triangleq \frac{d}{dz} (\Phi(z, \mathbf{x})). \quad (8)$$

Define improvement [22] as

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

where f_{\min}^n is the smallest value in \mathbf{f}_n . The Expected Improvement (EI) based on the Kriging prediction is

$$\text{EI}(\mathbf{x}) = \mathbb{E}[I(z)] = \int_{-\infty}^{+\infty} (f_{\min}^n - z)_+ \varphi(z, \mathbf{x}) dz = \int_{-\infty}^{f_{\min}^n} (f_{\min}^n - z) \varphi(z, \mathbf{x}) dz, \quad (10)$$

which can be computed in closed-form using (4) and (6) as

$$\text{EI}(\mathbf{x}, f_{\min}^n, \hat{f}, \hat{\sigma}) = \hat{\sigma}(\mathbf{x}) [u \Phi_{\mathcal{N}}(u) + \varphi_{\mathcal{N}}(u)], \quad (11)$$

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

$$u = \frac{f_{\min}^n - \hat{f}(\mathbf{x})}{\hat{\sigma}(\mathbf{x})}. \quad (12)$$

EGO achieves an iterative search for the global minimum of f and an associated global minimizer. Since $\text{EI}(\mathbf{x})$ is simple and fast to evaluate using (4) and (6), it can be optimized at each step via an auxiliary algorithm to be chosen (one may for instance choose DIRECT [24] as recommended in [25], but many other algorithms could be considered including [26]). The maximizer of $\text{EI}(\mathbf{x})$ is then used to run a single costly simulation of the black-box function $f(\cdot)$. The resulting data are appended in the sets $\{\mathcal{X}_n, \mathbf{f}_n\}$ used for updating the GP model at the next iteration. LHS is used to initialize \mathcal{X}_n at Step 1, a usual heuristic being to draw ten points per dimension of \mathbb{X} . EGO stops when the number of evaluations reaches the budget n_{\max} allotted for the evaluation of $f(\cdot)$, or when Expected Improvement falls below the threshold ε_{EI} [22].

The theoretical properties of EGO and other algorithms based on EI have been studied in [27, 28]. If the performance index is appropriately described by a GP with *known* and *fixed* covariance function, then convergence to one of the global optimizers is guaranteed as the number of evaluations tends to infinity. Convergence rates are also discussed in [28]. However, in real-life applications, it is impossible to check whether the performance index is appropriately described by a GP and even if it is so, the covariance function is usually unknown. Despite these limitations, EGO and other Kriging-based optimization algorithms have demonstrated their ability to find an estimate of the global optimum and a global optimizer on both analytical and real-world examples [22, 25, 29, 30].

Algorithm 2 EGO [16]

Set $\varepsilon_{\text{EI}}, n_{\max}$.

1. Choose an initial sampling $\mathcal{X}_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ in \mathbb{X}
2. Compute $\mathbf{f}_n = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^T$

while $\max_{\mathbf{x} \in \mathbb{X}} \text{EI}(\mathbf{x}) > \varepsilon_{\text{EI}}$ **and** $n < n_{\max}$

3. Fit the Kriging model on the known data points $\{\mathcal{X}_n, \mathbf{f}_n\}$ with (4)-(6)
4. Find $f_{\min}^n = \min_{i=1, \dots, n} \{f(\mathbf{x}_i)\}$
5. Find $\mathbf{x}_{n+1} = \arg \max_{\mathbf{x} \in \mathbb{X}} \text{EI}(\mathbf{x})$
6. Compute $f(\mathbf{x}_{n+1})$, append it to \mathbf{f}_n and append \mathbf{x}_{n+1} to \mathcal{X}_n
7. $n \leftarrow n + 1$

end while

2.3 MiMaReK 1

MiMaReK stands for *Minimax optimization via relaxation and Kriging*. It searches for the solution of (1) in the context of costly simulations by combining Algorithms 1 and 2. The initial algorithm MiMaReK 1 was presented in [10]. A simplified description of the resulting procedure is given in Algorithm 3. In this algorithm, two instances of EGO with two separate Kriging models are used, one for each of the two optimization steps of Algorithm 1.

- The first one depends only on \mathbf{x}_c , and interpolates $\max_{\mathbf{x}_e \in \mathcal{R}_e} \{J(\mathbf{x}_c, \mathbf{x}_e)\}$.

- The second one depends only on \mathbf{x}_e and interpolates $J(\mathbf{x}_c^*, \mathbf{x}_e)$.

The relaxation principle from Algorithm 1 is incorporated by evaluating and keeping in memory the values taken by the black-box function for all couples $\{\mathbf{x}_c \in \mathcal{X}_0^c, \mathbf{x}_e \in \mathcal{R}_e\}$, so that EGO can be initialized at Step 2. This means that n_c evaluations of the black-box function should be computed at each iteration of Step 2, corresponding to the n_c values of \mathbf{x}_c in the initial design \mathcal{X}_0^c associated to the new point \mathbf{x}_e^* found at the previous iteration of Step 3. This strategy uses the classical Expected Improvement expression to find \mathbf{x}_c^* , which makes it simple to implement and limits the complexity of the Kriging model to a dimension of d_c with only n_c initial points. The same scheme is employed at Step 3, where the maximization is carried out in a space of dimension d_e with n_e initial samples. However, the main drawback of this strategy is a loss of information because the evaluations carried out during previous runs of the EGO algorithms are not taken into account, which may lead to sampling repeatedly in areas of interest for determining the worst-case performance.

The tuning parameters of MiMaReK 1 are the initial numbers of evaluations n_c and n_e , the maximum numbers of evaluations n_{EI}^c and n_{EI}^e , and the thresholds $\varepsilon_{\text{EI}}^c$ and $\varepsilon_{\text{EI}}^e$ for the two instances of EGO, as well as the global relaxation threshold ε_R . An analysis performed in [10] on analytical test functions has confirmed that the smaller the thresholds $\varepsilon_{\text{EI}}^c$, $\varepsilon_{\text{EI}}^e$ and ε_R are, the more accurate the solution will be (at the price of more evaluations). Another interesting feature is that an approximate minimax solution is still obtained with higher thresholds, but with less accuracy.

To assess the improvement in computational cost achieved by the new algorithm, the number of evaluations required by MiMaReK 1 is established as follows. It is assumed that the maximum numbers of iterations (n_{EI}^c and n_{EI}^e) are reached during the optimizations by EGO. The first iteration of the outer loop starts with a minimization to find \mathbf{x}_c^* . This requires n_c evaluations for the initial design and n_{EI}^c evaluations for the iterations. Then comes the maximization to find \mathbf{x}_e^* , which requires n_e evaluations for the initial design and n_{EI}^e evaluations for the iterations. In the second loop, the values differ because the set \mathcal{R}_e now has two components. When minimizing to find \mathbf{x}_c^* , the worst-case value at each point in the initial design must be updated because there is a new element in \mathcal{R}_e , and this update still requires n_c function evaluations. During the iterations, however, evaluating a trial point \mathbf{x}_c now requires *two* evaluations of the expensive function, $J(\mathbf{x}_c, \mathbf{x}_e^{(1)})$ and $J(\mathbf{x}_c, \mathbf{x}_e^{(2)})$, because the worst value for all points in \mathcal{R}_e is searched for. The number of evaluations for the maximization to find \mathbf{x}_e^* remains unchanged. Thus we have $n_c + n_{\text{EI}}^c + n_e + n_{\text{EI}}^e$ evaluations during the first outer loop, $n_c + 2n_{\text{EI}}^c + n_e + n_{\text{EI}}^e$ during the second, $n_c + 3n_{\text{EI}}^c + n_e + n_{\text{EI}}^e$ during the third, and so forth. Assuming that N iterations of the outer loop are required, the total number of evaluations of the expensive function is finally

$$n_{\text{MM1}} = N \left(n_c + n_e + \frac{N+1}{2} n_{\text{EI}}^c + n_{\text{EI}}^e \right). \quad (13)$$

An illustration of the behavior of the relaxation procedure at Step 2 of MiMaReK 1 for the test function f_3 (see Section 4.1) is provided in Figure 1. The function displayed corresponds to the Kriging approximation of $\{\max_{\mathbf{x}_e \in \mathcal{R}_e} J(\mathbf{x}_c, \mathbf{x}_e)\}$ over \mathbb{X}_c . The dots indicate the location of the points sampled by the algorithm. The

Algorithm 3 MiMaReK 1 [10]**1. Step 1**

- (a) Choose \mathbf{x}_e^* in \mathbb{X}_e . Take $\mathcal{R}_e = \{\mathbf{x}_e^*\}$.
- (b) Choose $\mathcal{X}_0^c = \{\mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,n_c}\}$ in \mathbb{X}_c and $\mathcal{X}_0^e = \{\mathbf{x}_{e,1}, \dots, \mathbf{x}_{e,n_e}\}$ in \mathbb{X}_e .

while e is larger than some positive scalar ε_R chosen by the user

2. Step 2 Minimization with relaxation using EGO

- (a) Find $\mathbf{x}_c^* = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \left\{ \max_{\mathbf{x}_e \in \mathcal{R}_e} J(\mathbf{x}_c, \mathbf{x}_e) \right\}$ using EGO on \mathcal{X}_0^c and the previously computed values of J for $\{\mathbf{x}_c \in \mathcal{X}_0^c, \mathbf{x}_e \in \mathcal{R}_e\}$.
- (b) Compute $J^* = \max_{\mathbf{x}_e \in \mathcal{R}_e} J(\mathbf{x}_c^*, \mathbf{x}_e)$.

3. Step 3 Maximization using EGO

- (a) Find $\mathbf{x}_e^* = \arg \max_{\mathbf{x}_e \in \mathbb{X}_e} \{J(\mathbf{x}_c^*, \mathbf{x}_e)\}$ using EGO on \mathcal{X}_0^e and append it to \mathcal{R}_e .

- 4. Step 4** Compute $e = J(\mathbf{x}_c^*, \mathbf{x}_e^*) - J^*$.

end while

true optimum is located at $\mathbf{x}_c^* = 10$ for this test function, where the function value is equal to $9.7794 \cdot 10^{-2}$ (dashed line). Recall that, at this step of Algorithm 3, the objective is to minimize this function over \mathbb{X}_c . It can be seen that at each iteration a larger portion of the function graph is located above the true optimum. This is achieved thanks to the relaxation procedure that iteratively excludes local minima by incorporating a new vector from \mathbb{X}_e (selected at Step 3 of MiMaReK 1) into the finite set \mathcal{R}_e .

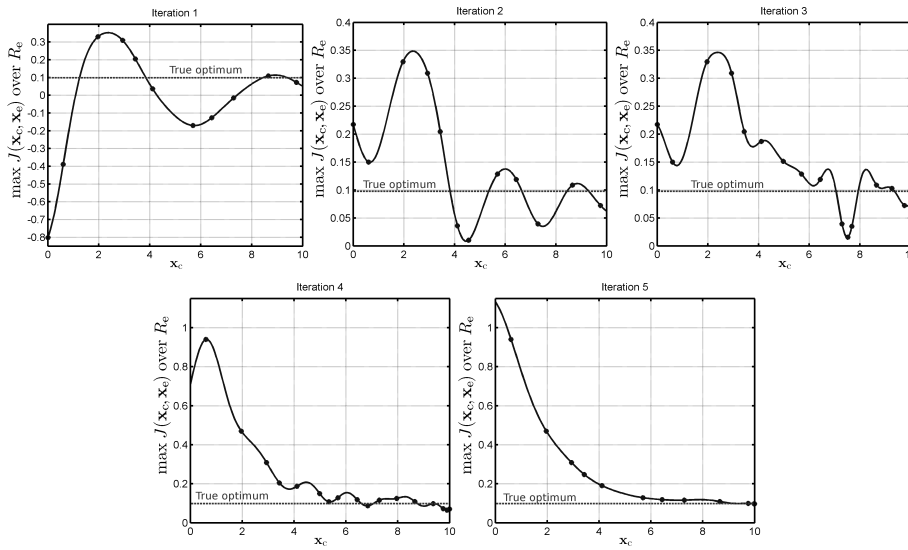


Fig. 1: Effect of iterative relaxation at Step 2 of MiMaReK 1 for f_3

3 New strategy for saving evaluations: MiMaReK 2

MiMaReK 1 is quite effective for an economical determination of approximate solutions of minimax problems (see [10,31] for test cases). It has, however, some inherent inefficiencies. Each iteration of its outer loop is carried out by building two dedicated Kriging predictors from scratch (one for predicting, for a given value of \mathbf{x}_c , the performance index J as a function of \mathbf{x}_e , and the other for predicting the maximum of J over a finite number of values of \mathbf{x}_e , as a function of \mathbf{x}_c). As the the previous set of observations is not reused, this entails a number of costly evaluations that could be avoided by reusing all past evaluations of the performance index J in the next iteration.

This is achieved with the new strategy, which works with a single response surface in the combined space of design and environmental variables. Thus, every function evaluation can be used to update the surface and no information is lost.

A first simple step towards this decrease of the number of evaluations of J is to use a single Kriging predictor at Step 3 for all maximizations of $J(\mathbf{x}_c^*, \mathbf{x}_e)$ with respect to \mathbf{x}_e . 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.

3.1 Expected Improvement at relaxation step

Using the same Kriging predictor at Step 2 for the minimization with relaxation on \mathcal{R}_e is more complex regarding the EI optimization. An easy-to-implement idea would be to approximate the mean of this process by

$$\hat{\mu}(\mathbf{x}_c) = \max_{\mathbf{x}_e \in \mathcal{R}_e} \hat{J}(\mathbf{x}_c, \mathbf{x}_e) \triangleq \hat{J}(\mathbf{x}_c, \check{\mathbf{x}}_e) \quad (14)$$

and its variance by

$$\hat{\sigma}^2(\mathbf{x}_c) = \hat{\sigma}^2(\mathbf{x}_c, \check{\mathbf{x}}_e), \quad (15)$$

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 presented in Algorithm 4 (and called MiMaReK 2 in the following), the Expected Improvement of $\{\max_{\mathbf{x}_e \in \mathcal{R}_e} J(\mathbf{x}_c, \mathbf{x}_e)\}$ is computed instead, as

$$\text{EI}_{\text{mm}}(\mathbf{x}_c) = \mathbb{E} \left[\max_{i \in \{1, \dots, m\}} \left(J_{\min} - J(\mathbf{x}_c, \mathbf{x}_e^{(i)}) \right)_+ \right], \quad (16)$$

where J_{\min} is the best performance of $\{\max_{\mathbf{x}_e \in \mathcal{R}_e} J(\mathbf{x}_c, \mathbf{x}_e)\}$ obtained so far and m the number of points in \mathcal{R}_e .

In a preliminary version of MiMaReK 2 [32], the $\mathbf{x}_e^{(i)}$ were assumed independent and an approximation of EI_{mm} computed as the result of a numerical integration with a quadrature method, based on the univariate cumulative and probability densities corresponding to each environmental input vector in \mathcal{R}_e . However, if these input points are sufficiently close to each other, a significant approximation

error is introduced by considering them as independent. The estimation method used in the present paper thus does not make this simplifying assumption.

The Kriging covariance [33] between points $(\mathbf{x}_c, \mathbf{x}_e^{(i)})$ and $(\mathbf{x}_c, \mathbf{x}_e^{(j)})$ is equal to

$$\hat{\sigma}_{ij} = \sigma_Z^2 \left(R_{ij} - \mathbf{r}_i^T \mathbf{R}^{-1} \mathbf{r}_j \right), \quad (17)$$

where $R_{ij} = R\left((\mathbf{x}_c, \mathbf{x}_e^{(i)}), (\mathbf{x}_c, \mathbf{x}_e^{(j)})\right)$ and $\mathbf{r}_i = \mathbf{r}\left(\mathbf{x}_c, \mathbf{x}_e^{(i)}\right)$. For $i = j$, the variance formula (6) is obtained. The corresponding $m \times m$ Kriging covariance matrix is denoted by $\boldsymbol{\Sigma}_m$. The Monte-Carlo computation [7] of EI_{mm} for some $\mathbf{x}_c \in \mathbb{X}_c$ proceeds as follows.

1. Compute the vector $\hat{\mathbf{f}}$ with components $\hat{f}_i = \hat{J}\left(\mathbf{x}_c, \mathbf{x}_e^{(i)}\right)$ for $i = 1, \dots, m$.
2. Compute the covariance matrix $\boldsymbol{\Sigma}_m$ and let \mathbf{L} such that $\boldsymbol{\Sigma}_m = \mathbf{L}\mathbf{L}^T$ is its Cholesky factorization.
3. Generate N_{mc} Monte-carlo samples of Gaussian random vectors $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, \mathbf{I}_m)$.
4. For each realization ($k = 1, \dots, N_{\text{mc}}$), compute

$$\hat{f}_k^{\text{mc}} = \max_{i=1, \dots, m} \hat{\mathbf{f}} + \mathbf{L}\boldsymbol{\varepsilon}_k \quad (18)$$

and the corresponding improvement

$$\text{I}_k^{\text{mc}} = \left(J_{\min} - \hat{f}_k^{\text{mc}} \right)_+ \quad (19)$$

5. A Monte-Carlo estimate of EI_{mm} is then provided by

$$\text{EI}_{\text{mm}}^{\text{mc}} = \frac{1}{N_{\text{mc}}} \sum_{k=1}^{N_{\text{mc}}} \text{I}_k^{\text{mc}} \quad (20)$$

The Tallis formula recently reported in [34] for multi-EI computation might be employed to obtain an exact formula for EI_{mm} . It involves multiple calls to multivariate density functions, which may limit its applicability to small values of m .

3.2 Algorithm description

With a single Kriging model on $\mathbb{X}_c \times \mathbb{X}_e$, MiMaReK 2 performs two EGO optimizations (at Steps 2 and 3): one on \mathbb{X}_c using the modified Expected Improvement (20) and one on \mathbb{X}_e using the classical Expected Improvement (11). The main algorithmic difficulties reside in the management of the global set of evaluated points, while the optimizations should be performed separately on \mathbb{X}_c or \mathbb{X}_e . The set \mathcal{X} contains all sampled points $(\mathbf{x}_c, \mathbf{x}_e)$ and \mathbf{J} stores the corresponding values of J .

At Step 2, the objective is to find

$$\mathbf{x}_c^* = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \left\{ \max_{\mathbf{x}_e \in \mathcal{R}_e} J(\mathbf{x}_c, \mathbf{x}_e) \right\}. \quad (21)$$

This is achieved by EGO, which now exploits the new expected improvement criterion (20) evaluated by Monte-Carlo to find successive points of interest.

The computation of J_{\min} at Step 2 is performed on a finite subset \mathcal{X}_c containing all \mathbf{x}_c^* s obtained at the end of each iteration of Step 2. Another set \mathcal{J}_c stores $\max_{\mathbf{x}_e \in \mathcal{R}_e} J(\mathbf{x}_c^*, \mathbf{x}_e)$ for these points. Both \mathcal{J}_c and \mathcal{X}_c each contain N data points, where N is the number of elements in \mathcal{R}_e (which is equivalent to the number of iterations of the main loop comprising Steps 2 to 4).

At the beginning of each new instance of Step 2, N evaluations of the black-box function are required to evaluate $J(\mathbf{x}_c, \mathbf{x}_e^*)$ and check whether the maximum values in \mathcal{J}_c should be updated, because a worse performance has been found for some of the previous optimal design vectors in association with a new candidate environmental vector \mathbf{x}_e^* . The initial target value J_{\min} to be used in the EGO algorithm at Step 2 is then set to the minimum of \mathcal{J}_c .

The same procedure is applied when a new \mathbf{x}_c^* is found at each iteration of the EGO algorithm at Step 2; N evaluations are performed to obtain the values of $J(\mathbf{x}_c^*, \mathbf{x}_e)$ for all $\mathbf{x}_e \in \mathcal{R}_e$ and they are appended to the joint sets $\{\mathcal{X}, \mathbf{J}\}$ (unlike in MiMaReK 1). If the maximum of these values is smaller than J_{\min} , it then becomes the new J_{\min} . After convergence of EGO, the worst-case value J^* is set to J_{\min} and \mathbf{x}_c^* is set to the corresponding argument. These values are then respectively appended to the sets \mathcal{J}_c and \mathcal{X}_c .

At Step 3, the objective is to find

$$\mathbf{x}_e^* = \arg \max_{\mathbf{x}_e \in \mathbb{X}_e} J(\mathbf{x}_c^*, \mathbf{x}_e) \quad (22)$$

Since EGO has been presented for minimization, the maximization of J carried out at Steps 3(a)ii and 3b is transformed into the minimization of $-J$. The relaxation procedure (Algorithm 1) on which MiMaReK is built requires that this maximization problem must be solved on \mathbb{X}_e with a fixed $\mathbf{x}_c = \mathbf{x}_c^*$. Since the global Kriging model is built on the joint space $\mathbb{X}_c \times \mathbb{X}_e$, the evaluation of the target maximal value J_{\max} should be carried out only for samples respecting $\mathbf{x}_c = \mathbf{x}_c^*$.

At the beginning of Step 3, it is most likely that only one point satisfies this constraint, i.e. $J_{\max} = -J_{\min}$. The number of candidates then grows during Step 3, since the maximization of the classical Expected Improvement on the joint model of J , with $\mathbf{x}_c = \mathbf{x}_c^*$ fixed, leads to new evaluation of the black-box function. When this step terminates, a new \mathbf{x}_e^* has been found and the convergence condition on $J(\mathbf{x}_c^*, \mathbf{x}_e^*) - J^*$ can be checked. Depending on the result, the algorithm either stops and provides the approximate minimax solution (up to the precision ε_R) or starts another round (N is incremented by 1).

Let us now count the number of evaluations needed by MiMaReK 2, for comparison with MiMaReK 1. It is still assumed that the maximum numbers of iterations (n_{EI}^c and n_{EI}^e) are reached during the optimizations by EGO, and additionally that the number n of initial samples is the same as the total number of initial samples in MiMaReK 1, so $n = n_c + n_e$. At the beginning of MiMaReK 2, $n_c + n_e$ evaluations are performed. Then, during Step 2, n_{EI}^c additional evaluations are required and n_{EI}^e at Step 3. Therefore, after one round of the main loop ($N = 1$), the number of evaluations is simply $n_c + n_e + n_{\text{EI}}^c + n_{\text{EI}}^e$.

When the second round begins, there are now 2 points in \mathcal{R}_e and 2 points in \mathcal{X}_c (the \mathbf{x}_c^* found after the first iteration of Step 2), therefore it is necessary to evaluate the value of $J(\mathbf{x}_c, \mathbf{x}_e^*)$ for the first \mathbf{x}_c (the other one has already been done at the end of Step 3), thus one additional evaluation is necessary. Then, during EGO at Step 2, $N = 2$ evaluations are required for each new point \mathbf{x}_c^* resulting

Algorithm 4 MiMaReK 2**1. Step 1**

- (a) Choose an initial design \mathcal{X} with n points in the joint space $\mathbb{X}_c \times \mathbb{X}_e$.
- (b) Compute the vector \mathbf{J} of the corresponding performance index values.
- (c) Pick one point $\mathbf{x}^* = (\mathbf{x}_c^*, \mathbf{x}_e^*)$ in \mathcal{X} .
- (d) Take $\mathcal{R}_e = \{\mathbf{x}_e^*\}$, $\mathcal{X}_c = \{\mathbf{x}_c^*\}$ and $\mathcal{J}_c = \{J(\mathbf{x}_c^*, \mathbf{x}_e^*)\}$.

while $e > \varepsilon_R$, with ε_R a positive scalar chosen by the user,

2. Step 2 Minimization with relaxation using EGO

- (a) Evaluate $J(\mathbf{x}_c, \mathbf{x}_e^*)$ for all $\mathbf{x}_c \in \mathcal{X}_c$ and update \mathcal{J}_c .
- (b) **while** $\text{EI}_{\text{mm}}^{\text{mc}}(\mathbf{x}_c)$ is large enough and n_{EI}^c is not reached,
 - i. Fit a Kriging model on the data $\{\mathcal{X}, \mathbf{J}\}$.
 - ii. Compute the best performance obtained so far J_{min} using \mathcal{X}_c and \mathcal{J}_c .
 - iii. Find the next point of interest $\mathbf{x}_c^* = \arg \max_{\mathbf{x}_c \in \mathbb{X}_c} \text{EI}_{\text{mm}}^{\text{mc}}(\mathbf{x}_c)$ with (20).
 - iv. Append to \mathcal{X} all the couples $\{\mathbf{x}_c^*, \mathbf{x}_e\}$ for all $\mathbf{x}_e \in \mathcal{R}_e$.
 - v. Compute the performance index at each of the points thus introduced in \mathcal{X} , and append the results to \mathbf{J} .

end while

- (c) Find J^* , the minimum of the worst performances, and its argument \mathbf{x}_c^* . Append them respectively to \mathcal{J}_c and \mathcal{X}_c .

3. Step 3 Maximization using EGO

- (a) **while** $\text{EI}(\mathbf{x}_e)$ is large enough and n_{EI}^e is not reached,
 - i. Fit a Kriging model on the data $\{\mathcal{X}, \mathbf{J}\}$.
 - ii. Find the worst performance obtained so far $J_{\text{max}} = \min_{\mathbf{x}_c = \mathbf{x}_c^*} \{-\mathbf{J}\}$.
 - iii. Find the next point of interest $\mathbf{x}_e^* = \arg \max_{\mathbf{x}_e \in \mathbb{X}_e} \text{EI}(\mathbf{x}_e)$.
 - iv. Introduce in the design \mathcal{X} the point $(\mathbf{x}_c^*, \mathbf{x}_e^*)$.
 - v. Compute $J(\mathbf{x}_c^*, \mathbf{x}_e^*)$, and append it to \mathbf{J} .

end while

- (b) Find $\mathbf{x}_e^* = \arg \min_{\mathbf{x}_c = \mathbf{x}_c^*} \{-\mathbf{J}\}$ and append it to \mathcal{R}_e .

4. Step 4 Compute $e = J(\mathbf{x}_c^*, \mathbf{x}_e^*) - J^*$

end while

from the EI_{mm} optimization and n_{EI}^c steps are performed, which makes in the end $N \cdot n_{\text{EI}}^c$ additional evaluations. During Step 3, n_{EI}^e evaluations are required. Thus, at the end of the second round of the main loop ($N = 2$), the number of additional evaluations is equal to $2n_{\text{EI}}^c + 1 + n_{\text{EI}}^e$.

At the beginning of Step 3, there are now $N = 3$ points in \mathcal{R}_e , therefore the $N - 2$ values of $J(\mathbf{x}_c, \mathbf{x}_e^*)$ for the \mathbf{x}_c that are different from the last \mathbf{x}_c^* should be evaluated, resulting in two additional evaluations. During EGO at Step 2, there are again $N \cdot n_{\text{EI}}^c$ performed (with now $N = 3$), and n_{EI}^e during EGO at Step 3. Thus, the total number of evaluations for the third round is equal to $3n_{\text{EI}}^c + 2 + n_{\text{EI}}^e$.

By adding these number of evaluations and generalizing to N iterations of the main loop, the following count is obtained

$$n_{\text{MM2}} = n_c + n_e + \left(n_{\text{EI}}^c \sum_{i=1}^N i \right) + \left(\sum_{i=1}^{N-1} i \right) + N n_{\text{EI}}^e, \quad (23)$$

$$n_{\text{MM2}} = n_c + n_e + \frac{N(N+1)}{2} n_{\text{EI}}^c + \frac{N(N-1)}{2} + N n_{\text{EI}}^e, \quad (24)$$

$$n_{\text{MM2}} = n_c + n_e + N \left(\frac{(N+1)}{2} n_{\text{EI}}^c + \frac{(N-1)}{2} + n_{\text{EI}}^e \right), \quad (25)$$

By subtracting this number of evaluations from the one obtained for MiMaReK 1 at equation (13), one gets

$$n_{\text{MM1}} - n_{\text{MM2}} = (N-1)(n_c + n_e) - \frac{N(N-1)}{2} = (N-1) \left(n_c + n_e - \frac{N}{2} \right), \quad (26)$$

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

$$n_c + n_e > \frac{N}{2}. \quad (27)$$

This inequality will usually be satisfied, as can be seen in the examples of the next section. Many iterations of the global loop are indeed required to make the right-hand side larger than the total number of initial samples $n_c + n_e$ (n_c and n_e are usually chosen equal to 10 times $\dim \mathbb{X}_c$ and $\dim \mathbb{X}_e$, according to a widely-used rule of thumb).

4 Examples

In this section, the performances of MiMaReK 1 and MiMaReK 2 are evaluated and compared on six analytical test cases [11,12,35] and a simple engineering problem [36–38]. Using such examples from the literature facilitates comparison with alternative approaches.

4.1 Analytical test cases

The first four test functions have scalar arguments

$$\begin{aligned} f_1(x_c, x_e) &= (x_c - 5)^2 - (x_e - 5)^2, \\ f_2(x_c, x_e) &= \min\{3 - 0.2x_c + 0.3x_e, 3 + 0.2x_c - 0.1x_e\}, \\ f_3(x_c, x_e) &= \frac{\sin(x_c - x_e)}{\sqrt{x_c^2 + x_e^2}}, \\ f_4(x_c, x_e) &= \frac{\cos\left(\sqrt{x_c^2 + x_e^2}\right)}{\sqrt{x_c^2 + x_e^2 + 10}}, \end{aligned}$$

while the last two have two-dimensional vector arguments

$$\begin{aligned} f_5(\mathbf{x}_c, \mathbf{x}_e) &= 100(x_{c2} - x_{c1}^2)^2 + (1 - x_{c1})^2 - x_{e1}(x_{c1} + x_{c2}^2) - x_{e2}(x_{c1}^2 + x_{c2}), \\ f_6(\mathbf{x}_c, \mathbf{x}_e) &= (x_{c1} - 2)^2 + (x_{c2} - 1)^2 + x_{e1}(x_{c1}^2 - x_{c2}) + x_{e2}(x_{c1} + x_{c2} - 2). \end{aligned}$$

Table 1 summarizes the optimal values as indicated in [12]. 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 $n = 10 \times \dim \mathbb{X}$,
- the thresholds $(\varepsilon_R, \varepsilon_{\text{EI}}^c, \varepsilon_{\text{EI}}^e)$ are set to 10^{-3} , and the maximum numbers of iterations n_{EI}^c and n_{EI}^e are set to $20 \times \dim \mathbb{X}_c$ and $20 \times \dim \mathbb{X}_e$ respectively.

Table 2 reports the numerical results obtained with MiMaReK 1 and MiMaReK 2 in terms of mean squared error (MSE) of $\hat{\mathbf{x}}_c$ with respect to the reference value \mathbf{x}_c^* , averaged on fifty random initializations, and those from [11], [12] and [35] for comparison. Table 3 gives the number of evaluations required to obtain these results. The two versions of MiMaReK are always competitive compared to other methods. Few evaluations turned out to be necessary on these test problems, while the methods proposed in [11] and [12] required about 10^5 evaluations. Results obtained with an arbitrarily fixed number of evaluations equal to 110 were reported in [35] using a very sparsely detailed method. The number of evaluations performed by MiMaReK 2 is always significantly smaller than with MiMaReK 1, and (27) is always satisfied.

Table 1: Reference solutions \mathbf{x}_c^* , \mathbf{x}_e^* and $f_i(\mathbf{x}_c^*, \mathbf{x}_e^*)$ for the six analytical functions (Section 4.1)

Function	\mathbb{X}_c	\mathbb{X}_e	\mathbf{x}_c^*	\mathbf{x}_e^*	$f_i(\mathbf{x}_c^*, \mathbf{x}_e^*)$
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	[0; 10]	[0; 10]	5	5	0
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	[0; 10]	[0; 10]	0	0	3
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	[0; 10]	[0; 10]	10	2.1257	$9.7794 \cdot 10^{-2}$
$f_4(\mathbf{x}_c, \mathbf{x}_e)$	[0; 10]	[0; 10]	7.0441	10	$4.2488 \cdot 10^{-2}$
$f_5(\mathbf{x}_c, \mathbf{x}_e)$	$[-0.5; 0.5] \times [0; 1]$	$[0; 10]^2$	0.5 0.25	0 0	0.25
$f_6(\mathbf{x}_c, \mathbf{x}_e)$	$[-1; 3]^2$	$[0; 10]^2$	1 1	any any	1

4.2 Simple engineering problem

The optimal design of a vibration absorber for a structure with an uncertain forcing frequency is a classical benchmark in mechanics, initially proposed in [36]. It can be formalized as a minimax optimization problem, for which various algorithms, ranging from analytical optimization to evolutionary strategies, have already been employed [37, 38]. The results found in these papers are very similar, which makes

Table 2: Empirical mean \pm standard deviation of $\text{MSE}(\hat{\mathbf{x}}_c)$ based on 50 runs for MiMaReK 1 and MiMaReK 2 and comparison with other methods (Section 4.1)

	MiMaReK 1	MiMaReK 2	[11] (10^5 ev.)	[12] (10^5 ev.)	[35] (110 ev.)
f_1	0 ± 0	$3.05 \cdot 10^{-7}$ $\pm 5.43 \cdot 10^{-7}$	$1.90 \cdot 10^{-9}$ $\pm 8.04 \cdot 10^{-9}$	$2.08 \cdot 10^{-15}$ $\pm 5.71 \cdot 10^{-13}$	$4.05 \cdot 10^{-9}$ $\pm 1.38 \cdot 10^{-8}$
f_2	$3.12 \cdot 10^{-3}$ $\pm 7.13 \cdot 10^{-3}$	$3.9 \cdot 10^{-3}$ $\pm 7.2 \cdot 10^{-3}$	$1.50 \cdot 10^{-3}$ $\pm 8.90 \cdot 10^{-3}$	$1.09 \cdot 10^{-4}$ $\pm 7.20 \cdot 10^{-4}$	$5.53 \cdot 10^{-13}$ $\pm 1.04 \cdot 10^{-12}$
f_3	$1.48 \cdot 10^{-3}$ $\pm 7.30 \cdot 10^{-3}$	$1.52 \cdot 10^{-7}$ $\pm 7.65 \cdot 10^{-7}$	$3.30 \cdot 10^{-3}$ $\pm 1.14 \cdot 10^{-2}$	$5.79 \cdot 10^{-6}$ $\pm 4.98 \cdot 10^{-5}$	$1.75 \cdot 10^{-3}$ $\pm 7.34 \cdot 10^{-3}$
f_4	$2.30 \cdot 10^{-2}$ $\pm 2.79 \cdot 10^{-2}$	$5.58 \cdot 10^{-5}$ $\pm 2.73 \cdot 10^{-4}$	$4.31 \cdot 10^{-2}$ $\pm 8.34 \cdot 10^{-2}$	$2.19 \cdot 10^{-2}$ $\pm 3.79 \cdot 10^{-2}$	$8.14 \cdot 10^{-3}$ $\pm 1.97 \cdot 10^{-3}$
f_5	$1.23 \cdot 10^{-4}$ $\pm 7.58 \cdot 10^{-4}$	$1.34 \cdot 10^{-5}$ $\pm 2.80 \cdot 10^{-5}$	$4.93 \cdot 10^{-2}$ $\pm 5.90 \cdot 10^{-2}$	$8.2 \cdot 10^{-11}$ $\pm 3.60 \cdot 10^{-4}$	$6.80 \cdot 10^{-10}$ $\pm 3.50 \cdot 10^{-10}$
f_6	$1.43 \cdot 10^{-3}$ $\pm 3.88 \cdot 10^{-3}$	$1.78 \cdot 10^{-4}$ $\pm 6.59 \cdot 10^{-4}$	$1.75 \cdot 10^{-2}$ $\pm 2.25 \cdot 10^{-2}$	$3.4 \cdot 10^{-3}$ $\pm 8.32 \cdot 10^{-2}$	$4.21 \cdot 10^{-3}$ $\pm 5.50 \cdot 10^{-3}$

Table 3: Empirical mean \pm standard deviation of number of evaluations by MiMaReK 1 and MiMaReK 2 based on 50 runs (Section 4.1)

Test function	f_1	f_2	f_3	f_4	f_5	f_6
MiMaReK 1	52 ± 1	270 ± 68	281 ± 72	279 ± 89	94 ± 4	223 ± 89
MiMaReK 2	33 ± 0	150 ± 40	224 ± 32	225 ± 38	55 ± 1	143 ± 83

it possible to use them as reference solutions to assess the behavior of MiMaReK on such problems.

The system is described in Figure 2. A primary structure with mass m_1 is subjected to a sinusoidal force of amplitude X_0 and unknown frequency ω . The amplitude of the resulting harmonic motion of m_1 is denoted by X_1 . A smaller structure with mass m_2 is used to compensate for the oscillations generated by this disturbance through a viscous damping action. The design problem is to determine the characteristics of this damper so as to be robust to the worst forcing frequency.

The performance index to be optimized is the normalized maximum displacement of the primary structure, which can be expressed [36,37] as

$$J = \frac{k_1 X_1}{X_0} = \frac{1}{Z} \sqrt{\left(1 - \frac{\beta^2}{T^2}\right)^2 + 4 \left(\frac{\zeta_2 \beta}{T}\right)^2}, \quad (28)$$

where

$$Z^2 = \left[\frac{\beta^2}{T^2} (\beta^2 - 1) - \beta^2 (1 + \mu) - 4 \frac{\zeta_1 \zeta_2 \beta^2}{T} + 1 \right]^2 + 4 \left[\frac{\zeta_1 \beta^3}{T^2} + \frac{\zeta_2 \beta^3 (1 + \mu) - \zeta_2 \beta - \zeta_1 \beta}{T} \right]^2. \quad (29)$$

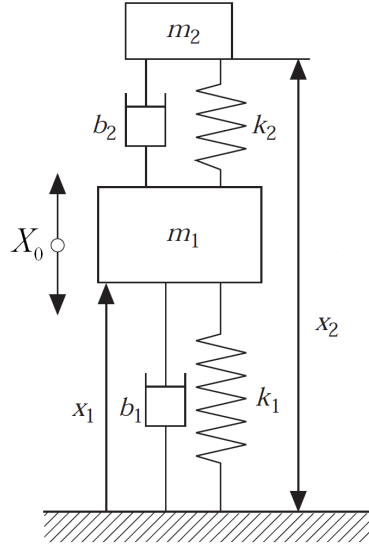


Fig. 2: Vibration absorber

These definitions involve the reduced variables

$$\omega_i = \sqrt{\frac{k_i}{m_i}}, \quad \zeta_i = \frac{b_i}{2\sqrt{k_i m_i}}, \quad \beta = \frac{\omega}{\omega_1}, \quad T = \frac{\omega_2}{\omega_1}, \quad \mu = \frac{m_2}{m_1}. \quad (30)$$

The parameters of the main system are fixed to $\mu = 0.1$, $\zeta_1 = 0.1$ and $\omega_1 = 100$. The decision variables to be determined are ζ_2 and T , while the optimization should be robust to the effect of the environmental variable β . The design problem can thus be written as the search for

$$\{\zeta_2^*, T^*, \beta^*\} = \arg \min_{\zeta_2, T} \max_{\beta} J. \quad (31)$$

Following [36–38], it is assumed that $\zeta_2 \in [0, 1]$, $T \in [0, 2]$ and $\beta \in [0, 2.5]$. In spite of the analytical character of the performance index, it is treated here as a black box.

Reference results of the literature are in Table 4. The result given by [38], which was obtained with a refined sampling grid, will be used as the reference for analyzing the results provided by the two versions of MiMaReK. The tuning parameters of MiMaReK 1 and Mimarek 2 were set to $\varepsilon_R = 10^{-4}$, $\varepsilon_{EI}^c = \varepsilon_{EI}^e = 10^{-6}$, $n_{EI}^c = 40$, $n_{EI}^e = 20$. For comparison, a similar design problem has been addressed in [39] using ant-colony optimization, and required more than 10^4 evaluations, which will be impractical if the design cost function were evaluated via costly simulations. The results averaged on 50 runs (MSE and standard deviation for \mathbf{x}_c^* , \mathbf{x}_e^* , optimum value $J_{\min\max}$ and number of evaluations) obtained with MiMaReK 1 and MiMaReK 2 are given in Table 5. They are very close to the reference, although the number of evaluations of the cost function was much smaller. At least six iterations of the global relaxation loop were required to reach the solution, which suggests that the strategy from reference [35], which uses a fixed number of 110

evaluations, would be inappropriate. MiMaReK 2 required less evaluations than MiMaReK 1 to achieve a similar performance level, which confirms the interest of this new strategy for minimax optimization of black-box functions.

Table 4: Reference minimax results obtained for damper design (Section 4.2)

	ζ_2^*	T^*	β^*	J_{minimax}
Randall [36]	0.204	0.861	1.038	2.6271
Pennestri [37]	0.202	0.861	1.04	2.6272
Brown/Singh [38]	0.1986	0.8619	1.043	2.6227

Table 5: Empirical mean \pm standard deviation of $\text{MSE}(\mathbf{x}_c^*)$, $\text{MSE}(\mathbf{x}_e^*)$, $\text{MSE}(J_{\text{minimax}})$ with respect to reference solution [38], and number of evaluations for MiMaReK 1 and MiMaReK 2 based on 50 runs (Section 4.2)

	$\text{MSE}(\mathbf{x}_c^*)$	$\text{MSE}(\mathbf{x}_e^*)$	$\text{MSE}(J_{\text{minimax}})$	Nb. evaluations
MiMaReK 1	$1.14 \cdot 10^{-4}$ $\pm 1.13 \cdot 10^{-4}$	$7.76 \cdot 10^{-5}$ $\pm 8.53 \cdot 10^{-5}$	$4.57 \cdot 10^{-4}$ $\pm 2.92 \cdot 10^{-4}$	1452 ± 566
MiMaReK 2	$2.92 \cdot 10^{-4}$ $\pm 2.37 \cdot 10^{-4}$	$1.75 \cdot 10^{-4}$ $\pm 1.73 \cdot 10^{-4}$	$2.4 \cdot 10^{-3}$ $\pm 3 \cdot 10^{-3}$	883 ± 326

5 Conclusions and perspectives

Continuous minimax optimization problems for functions evaluated via costly numerical simulations is a difficult problem on which most existing algorithms are not applicable, either because they require a closed-form expression for the function or because too many evaluations are necessary. The MiMaReK algorithm presented in [10] is able to handle such problems under a restricted simulation budget by combining two relaxations tools: an iterative relaxation procedure first described in [15] and EGO, a global optimization procedure based on Kriging and Expected Improvement [16].

Following this framework, a new strategy based on a the evaluation of Expected Improvement in a minimax context has been proposed for further reducing the number of evaluations of the performance index for black-box minimax optimization. The performance of the methods have been evaluated on a collection of analytical test functions for comparison with other approaches and on a classical benchmark in mechanics. In all cases, the new algorithm significantly reduced the number of required evaluations of the performance function.

Further improvement in the reduction of the number of evaluations might be achieved by sampling strategies in the joint control and environmental space. If

interesting results have been obtained with probabilistic uncertainty [40,41], this is still an open topic for minimax optimization.

Constrained minimax optimization with MiMaReK remains to be investigated, in particular when the constraints are costly to evaluate and involve coupling between the design and environmental variables. Higher-dimensional engineering design problems will also be investigated in the near future. Since the new algorithm builds a single Kriging predictor on all the data collected throughout the minimax optimization procedure, the fitting of this model might become cumbersome when the dimension of the problem and the number of evaluations grow. In this regard, the use of sparse representations [42] may be quite beneficial.

Acknowledgments

The authors thank the reviewers for their comments and suggestions, which led to major improvements of the original submission.

References

1. D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53:464–501, 2011.
2. A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, Princeton, NJ, 2009.
3. C. Zang, M. I. Friswell, and J. E. Mottershead. A review of robust optimal design and its application in dynamics. *Computers & structures*, 83(4):315–326, 2005.
4. H. G. Beyer and B. Sendhoff. Robust optimization – a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, 2007.
5. X. Du and W. Chen. Towards a better understanding of modeling feasibility robustness in engineering design. *Journal of Mechanical Design*, 122:385–394, 2000.
6. R. Jin, X. Du, and W. Chen. The use of metamodeling techniques for optimization under uncertainty. *Structural and Multidisciplinary Optimization*, 25:99–116, 2003.
7. J. Janusevskis and R. Le Riche. Simultaneous Kriging-based estimation and optimization of mean response. *Journal of Global Optimization*, 55(2):313–336, 2013.
8. B. J. Williams, T. J. Santner, and W. I. Notz. Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica*, 10(4):1133–1152, 2000.
9. B. Rustem and M. Howe. *Algorithms for Worst-Case Design and Applications to Risk Management*. Princeton University Press, 2002.
10. J. Marzat, E. Walter, and H. Piet-Lahanier. Worst-case global optimization of black-box functions through Kriging and relaxation. *Journal of Global Optimization*, 55(4):707–727, 2013.
11. A. M. Cramer, S. D. Sudhoff, and E. L. Zivi. Evolutionary algorithms for minimax problems in robust design. *IEEE Transactions on Evolutionary Computation*, 13(2):444–453, 2009.
12. 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*, pages 1902–1905, 2011.
13. D. Du and P. M. Pardalos. *Minimax and Applications*. Kluwer Academic Publishers, Norwell, 1995.
14. P. Pappas and B. Rustem. An algorithm for the global optimization of a class of continuous minimax problems. *Journal of Optimization Theory and Applications*, 141(2):461–473, 2009.
15. K. Shimizu and E. Aiyoshi. Necessary conditions for min-max problems and algorithms by a relaxation procedure. *IEEE Transactions on Automatic Control*, 25(1):62–66, 1980.
16. D. R. Jones, M. J. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

17. J. Mockus. *Bayesian Approach to Global Optimization: Theory and Applications*. Kluwer Academic Publishers, Dordrecht, 1989.
18. H. J. Kushner. A versatile stochastic model of a function of unknown and time varying form. *Journal of Mathematical Analysis and Applications*, 5(1):150–167, 1962.
19. D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.
20. G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.
21. T. J. Santner, B. J. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, Berlin-Heidelberg, 2003.
22. M. Schonlau. *Computer Experiments and Global Optimization*. PhD thesis, University of Waterloo, Canada, 1997.
23. D. C. Montgomery. *Design and Analysis of Experiments, 7th Edition*. Wiley, Hoboken, 2008.
24. D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.
25. M. J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, USA, 2002.
26. Y. D. Sergeyev and D. E. Kvasov. Global search based on efficient diagonal partitions and a set of lipschitz constants. *SIAM Journal on Optimization*, 16(3):910–937, 2006.
27. E. Vazquez and J. Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference*, 140(11):3088–3095, 2010.
28. A. D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.
29. D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, 34(3):441–466, 2006.
30. J. Villemonteix, E. Vazquez, and E. Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009.
31. 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*, pages 398–403, 2012.
32. J. Marzat, E. Walter, and H. Piet-Lahanier. A new strategy for worst-case design from costly numerical simulations. In *Proceedings of the American Control Conference, Washington DC, USA*, pages 3991–3996, 2013.
33. J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.
34. C. Chevalier and D. Ginsbourger. Fast computation of the multi-points expected improvement with applications in batch selection. In G. Nicosia and P. Pardalos, editors, *Learning and Intelligent Optimization*, volume 7997 of *Lecture Notes in Computer Science*, pages 59–69. Springer Berlin Heidelberg, 2013.
35. 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*, pages 1–7, 2010.
36. S. E. Randall. Optimum vibration absorbers for linear damped systems. *ASME Journal of Mechanical Design*, 103:908–913, 1978.
37. E. Pennestri. An application of Chebyshev’s min-max criterion to the optimal design of a damped dynamic vibration absorber. *Journal of Sound and Vibration*, 217(4):757–765, 1998.
38. B. Brown and T. Singh. Minimax design of vibration absorbers for linear damped systems. *Journal of Sound and Vibration*, 330(11):2437–2448, 2011.
39. F. A. C. Viana, G. I. Kotinda, D. A. Rade, and V. Steffen Jr. Tuning dynamic vibration absorbers by using ant colony optimization. *Computers & Structures*, 86(13-14):1539–1549, 2008.
40. V. Dubourg, B. Sudret, and J.-M. Bourinet. Reliability-based design optimization using Kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44(5):673–690, 2011.

-
41. C. Chevalier. *Fast uncertainty reduction strategies relying on Gaussian process models*. PhD thesis, University of Bern, 2013.
 42. J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.