Noname manuscript No.
(will be inserted by the editor)

Unconstrained high-dimensional Bayesian optimization of an airfoil shape design via multi-fidelity surrogate modeling

M. Meliani

the date of receipt and acceptance should be inserted later

Abstract In engineering, science, and technology, predictions and design decisions can be made or informed by a variety of information sources that range from experimental data to computer models. These information sources typically encompass different mathematical formulations, different grid resolutions, different physics, or different modeling assumptions that simplify the problem. This leads to information sources with varying degrees of fidelity, with varying associated accuracy and querying costs. In this paper we propose a novel and flexible way to use multi-fidelity information sources optimally in the context of airfoil shape optimization using both a Reynolds averaged Navier-Stokes (RANS) solver and a low fidelity approximation based on a simplified physical formulation. The new developments based on Bayesian optimization and kriging metamodeling allow the aerodynamic optimization to be sped up and divide for example (on a 15-design-variable unconstrained optimization problem) the total cost by at least two compared to a single fidelity optimization.

1 Introduction

The use of multiple fidelities is particularly interesting for aircraft design decisions where the high-dimensionality of the design space and the cost of the high fidelity analyses make global optimization near impossible. Many works exist on using multiple information sources for surrogate modeling (different accuracies for the same quantity of interest) [1] or optimization [2,3]. Works also exist on modeling the differences in fidelity in an additive way [4,5]. However these approaches fail to work in the general case where different assumptions about the physics of the problem exist between fidelity levels. In this context, the work of [6] provided powerful tools for multi-fidelity surrogate modeling which [7] used for a multi-fidelity extension of the Efficient Global Optimization algorithm (EGO) [8] for

M. Meliani
ISAE-Supaero, DMSM, Toulouse, France
E-mail: meliani@kth.se
Present address: KTH Royal Institute of Teknology
The Centre for ECO2 Vehicle Design, Teknikringen 8, SE-100 44 Stockholm, Sweden

unconstrained problems. [9] proposed an alternative formulation of multi-fidelity co-kriging with explicit contributions of each fidelity level to the overall model. In 2002, EGO was extended to constrained problems with the development of SEGO [10]. More recently SEGO was coupled to MOE (Mixture Of Experts) methodology in order to solve high-dimensional and constrained optimization problems such as wing shape aerodynamic optimization in the SEGOMOE framework [11, 12]. SEGOMOE is used as a reference for surrogate-based optimizations with a single fidelity (HF). Although it was developed for constrained optimization problems, in practice, SEGOMOE framework is able to treat either unconstrained or constrained problems with or without Mixture of Experts. In this paper these works are used to improve the efficiency of global optimization algorithms which suffer from the curse of dimensionality. We demonstrate that the use of multiple fidelities helps alleviate this curse and show that we can perform high-dimensional Bayesian optimization. We begin by reviewing kriging and multi-fidelity co-kriging formulations; we then present the multi-fidelity extension of EGO called MFEGO. We provide a theoretical analysis of the properties of the proposed algorithm including its global convergence. Finally, we present the results of an unconstrained optimization of the design of a subsonic airfoil with 15 design variables and compare the performance of MFEGO to standard methodologies such as EGO and the standard gradient-based optimizer SNOPT [13].

2 Kriging and Multi-Fidelity co-kriging description

One way to build an approximation of different functions of interest is kriging. The idea behind kriging is that the surrogate model $\hat{y}(x) = m(x)+Z(x)$, is comprised of two parts: 'regression' term $m(x)$ and a functional departure from that regression: $Z(x)$ [10]. We can write the regression term as the following:

$$m(x) = \sum_{j=1}^{k} \beta_j f_j, \tag{1}$$

where $\{f_1, \dots, f_k\}$ are basis functions of $m(x)$ and β_j are coefficients weighting these basis functions. *Ordinary kriging* denotes the special case where $k = 1$ and $f_1 = 1$, meaning the regression function takes the form of a constant, leaving most of the prediction work to $Z(x)$. $Z(x)$ on the other hand is a Gaussian process defined by the process variance σ^2 and a spatial correlation function known as the kernel of the process $R(\cdot, \cdot)$. The choice of the kernel influences greatly the way data is fitted. For the rest of this paper we use the squared exponential correlation kernel:

$$R(w, x) = \exp\left(- \sum_{k=1}^{d} \theta_{(k)} (w_{(k)} - x_{(k)})^2 \right) \tag{2}$$

where $x \in R^d$, $w \in R^d$ and $\theta \in R^d$ is a vector of hyperparameters of the kriging model, denoting the correlation along the different axes of space. Once we have defined the Kernel, and "fitted" the vector of hyperparameters θ using the training vector $X_T = \{x_1, \dots, x_n\}$ (with $x_i \in R^d$) yielding the responses $Y_T = \{y_1, \dots, y_n\}$

(with $y_i \in R$), we can express the mean and covariance of the Gaussian Process $Z(x)$:

$$\mu_Z(x) = r(x, X_T)' R^{-1}(X_T, X_T) Y_T \tag{3}$$

$$\sigma_Z^2(x) = \sigma^2(1 - r(x, X_T)' R^{-1}(X_T, X_T) r(x, X_T)) \tag{4}$$

where $x \in R^d$ is the prediction point, and X_T is the locations of training set. $R(X_T, X_T)$ is the matrix of correlations among the training points. $r(x, X_T)$, on the other hand, denotes the correlation between the prediction point and the training points. Note that Eq. (3) ensures the interpolation of the training points. In fact, if we make a prediction at training point x_i, the vector $r(x_i, X_T)$ will correspond to the i^{th} line of $R(X_T, X_T)$, so that $(r(x_i, X_T)' R^{-1}(X_T, X_T))'$ will give the i^{th} unit vector and so $\mu_Z(x_i) = y_i$. If we take into account the 'regression' term $m(x)$ the prediction becomes:

$$\mu(x) = m(x) + r(x)' R^{-1}(Y_T - m(x))$$

which we can also express using the basis introduced in Eq. (1):

$$\mu(x) = f(x)'\beta + r(x)' R^{-1}(Y_T - F\beta) \tag{5}$$

$$\begin{aligned}\sigma^2(x) = \sigma^2 \Big[&1 - r(x)' R^{-1} r(x) + \\ &\Big(f(x)' - r(x)' R^{-1} F\Big) \Big(F' R^{-1} F\Big) \Big(f(x)' - r(x)' R^{-1} F\Big)\Big]\end{aligned} \tag{6}$$

where β is the vector of coefficients β_j introduced in Eq. (1). F is the matrix of the values of the regression basis function at the positions of the training points, whereas $f(x)$ is the vector of values of these functions at the prediction point. When learning multi-fidelity models we can make assumptions to simplify the problem and inform our model (which results in a decrease of the data needed to learn the model). In general when modeling multiple fidelities that involve different assumptions on the physics of the problem (e.g., viscosity, compressibility, or turbulence in aerodynamics models), chances are that the low and high fidelities will have different scales and sometimes (in rare cases) poor correlations. [6] proposed a formulation that takes the correlation and scaling into account by introducing a factor $\rho \in R$ in the formulation above:

$$\begin{cases} f_{HF}(x) = \rho f_{LF}(x) + \delta(x) \\ \text{with} \qquad f_{LF}(\cdot) \perp \delta(\cdot) \end{cases} \tag{7}$$

where $\delta(\cdot)$ is the discrepancy function tasked with capturing the differences between the low- and high-fidelity functions (denoted resp. by $f_{LF}(x)$ and $f_{HF}(x)$) beyond scaling. The addition of the term ρ increases the robustness of the model. [9] proposed an implementation using the regression term expression of **universal kriging** introduced in Eq. (1) and extended it to take the lower fidelity model as a basis function such that the regression term becomes:

$$m(x) = \sum_{j=1}^{k} \beta_j f_j + \beta_\rho f_{LF}$$

where β_ρ is an estimation of ρ performed by a classic parameter estimation such as the likelihood maximization [14,15]. Assuming the independence of the high and low-fidelity models (two levels here), the mean and variance of the high-fidelity model are expressed:

$$\mu_{HF} = \rho\, \mu_{LF} + \mu_\delta \tag{8}$$

$$\sigma^2_{HF} = \rho^2\, \sigma^2_{LF} + \sigma^2_\delta. \tag{9}$$

The approach can be extended to l levels of fidelity. To make this explicit let us denote $f_0, \ldots, f_l$ the hierarchically ranked fidelity codes (from lowest $f_0 = f_{LF}$ to highest $f_l = f_{HF}$). Using the recursive formulation, we write:

$$\mu_k = \rho_{k-1}\, \mu_{k-1} + \mu_{\delta_k} \tag{10}$$

$$\sigma^2_k = \rho^2_{k-1}\, \sigma^2_{k-1} + \sigma^2_{\delta_k}. \tag{11}$$

The formulation by Le Gratiet [9], if satisfying the nested Design Of Experiments (DOE) requirement, offers explicit expressions of the contribution of fidelity levels to the uncertainty of the model. The nested DOE requirement states that $X_l \subseteq X_{l-1} \ldots \subseteq X_0$ where X_i is the vector of training points of the fidelity i and l is the highest fidelity. So with this assumption, a point computed at the highest fidelity has to be also computed at the lowest fidelities. We introduce the following notation in Eq. (11):

$$\begin{aligned} \sigma^2_{\delta,k} &= \sigma^2_{\delta_k} \qquad \text{for } k \in \{1, \ldots, l\} \\ \sigma^2_{\delta,0} &= \sigma^2_0 \qquad \text{for } k = 0. \end{aligned} \tag{12}$$

We express the uncertainty contribution of the fidelity level k at design point x (corrected from page 163, [9]) as:

$$\sigma^2_{cont}(k, x) = \sigma^2_{\delta,k}(x) \prod_{j=k}^{l-1} \rho^2_j, \tag{13}$$

which means that the variance contribution of the fidelity level k: $(\sigma^2_{\delta,k})$ is scaled using the recursive values of ρ_j until we get to the highest fidelity l. These contributions are essential to build Sequential Design or Optimization strategies. A Python implementation of multi-fidelity co-kriging based on Le Gratiet's work can be found in the open source Surrogate Modeling Toolbox (SMT). [1]

3 MFEGO methodology

Bayesian optimization is defined by J. Mockus [16] as an optimization technique based upon the minimization of the expected deviation from the extremum of the studied function. The objective function is treated as a black-box function. A Bayesian strategy sees the objective as a random function and places a prior over it. The prior captures our beliefs about the behavior of the function. After gathering the function evaluations, which are treated as data, the prior is updated to form the posterior distribution over the objective function. The posterior distribution, in turn, is used to construct an acquisition function (often also referred to as *infill sampling criterion*) that determines what the next query point should be.

[1] https://www.github.com/SMTorg/smt

3.1 Efficient Global Optimization: EGO

We describe here the Expected Improvement *infill sampling criterion* as well as the EGO algorithm based on [8]. Let F be an expensive black-box function to be minimized. We sample F at the different locations $X = \{x_1, x_2, \dots, x_n\}$ yielding the responses $Y = \{y_1, y_2, \dots, y_n\}$. We build a kriging model (also called Gaussian Process) with a mean function μ and a variance function σ^2 as presented in Section 2. The next step is to compute the Expected Improvement (EI) criterion. To do this, let us denote:

$$f_{min} = \min\{y_1, y_2, \dots, y_n\},$$

the EI function can be expressed:

$$E[I(x)] = E[\max(f_{min} - Y, 0)] \tag{14}$$

where Y is the random variable following the distribution $\mathcal{N}(\mu(x), \sigma^2(x))$. By expressing the right-hand side of Eq. (14) as an integral, and applying some tedious integration by parts, one can express the expected improvement in closed form:

$$E[I(x)] = \begin{cases} (f_{min} - \mu(x))\Phi\left(\frac{f_{min}-\mu(x)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{f_{min}-\mu(x)}{\sigma(x)}\right), & \text{if } \sigma > 0 \\ 0, & \text{if } \sigma = 0 \end{cases}, \tag{15}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ are respectively the cumulative and probability density functions of $\mathcal{N}(0,1)$. Next, we determine our next sampling point as:

$$x_{n+1} =_x \left(E[I(x)]\right). \tag{16}$$

We then test the response y_{n+1} of our black-box function F at x_{n+1}, rebuild the model taking into account the new information gained, and research the point of maximum expected improvement again.

3.2 EGO extension to Multi-Fidelity: MFEGO

We extend the EGO algorithm to work with multiple fidelities. In practice, as the SEGOMOE framework proposed by [11,12] is capable of handling both unconstrained and constrained problems and this with or without Mixture of Experts, we choose to note in this paper the unconstrained version of SEGOMOE without Mixture of Experts as EGO. The multi-fidelity is defined as MFEGO. The main idea of the proposed algorithm is that the search for the most promising sample and the choice of level of enrichment can be seen as problems to be tackled sequentially. Indeed, we can consider that given a Gaussian Process or kriging model ($m_k GP(\mu, \sigma^2)$) and a current best point (f_{min}), EI (or another infill sampling criterion) can be trusted to find the next *most promising point.* The choice of the *fidelity level of enrichment* is a different question that can be formulated thus: given the uncertainty at a chosen point, is it more interesting to query it at the highest-fidelity level or at the lower ones? This choice of a two-stage decision process (fix *most promising point* then *fidelity level of enrichment*) offers the advantage of greatly reducing the time of *infill sampling criterion* optimization without being restrictive in any way. The choice of the *fidelity level of enrichment* translates an idea stating that one should favor the use of low-fidelity samples for exploration, and high-fidelity ones for exploitation, while making sure that there is no resampling at the same location. Let $f_0, \dots, f_l$ be the lowest- to highest-fidelity of a quantity of interest, with querying costs $c_0, \dots, c_l$. Using the recursive formulation [9] with a constant ρ, we know that:

$$f_k = \rho_{k-1}\, f_{k-1} + \delta_k \qquad \text{for } k \in \{1, \dots, l\} \tag{17}$$

$$\rho_{k-1} = \mathrm{corr}(f_k, f_{k-1}) \frac{\mathrm{std}(f_k)}{\mathrm{std}(f_{k-1})} \tag{18}$$

$$\sigma_k^2 = \rho_{k-1}^2\, \sigma_{k-1}^2 + \sigma_{\delta_k}^2 \tag{19}$$

where the notations are std(·) for the standard deviation and corr(·, ·) for the correlation. Using the notations introduced in Eq. (12), the variance contribution of the fidelity level k at design point x^* defined by Eq. (13) is recalled here:

$$\sigma_{cont}^2(k, x^*) = \sigma_{\delta,k}^2(x^*) \prod_{j=k}^{l-1} \rho_j^2.$$

Due to the necessity of nested DOEs, all lower fidelities must be enriched at the same time.Thus, the uncertainty reduction becomes:

$$\sigma_{red}^2(k, x^*) = \sum_{i=0}^{k} \sigma_{\delta,i}^2(x^*) \prod_{j=i}^{l-1} \rho_j^2.$$

The corresponding cost to the enrichment of level of fidelities 0 through k is:

$$\mathrm{cost}_{\mathrm{total}}(k) = \sum_{i=0}^{k} c_i.$$

We propose the level of enrichment criterion as follows:

$$t =_{k \in (0, \dots, l)} \quad \frac{\sigma_{red}^2(k, x^*)}{\mathrm{cost}_{\mathrm{total}}(k)^2}$$

where t is the highest fidelity level to be added (the nested DOE imposing to enrich all lower fidelities). It is questionable whether it is possible to always find a common unit of measurement for the cost of an observation and the variance reduction, but as the variance scales with the square of the correlation (Eqs. (18) and (19)), it is reasonable to penalize the uncertainty reduction by the square of the cost. The correlation translates the quantity of information shared between multiple functions. Note that Le Gratiet [9] introduced a Sequential Design strategy where the uncertainty is simply penalized by the cost. Tests were realized with both approaches (penalization with simply the cost or the square of the cost). The penalization with the square of the cost constantly gave better results.

Algorithm 1 MFEGO Algorithm

```
 1: procedure ENRICH_LEVEL($model, x^*, costs$)                ▷ Which fidelities to query
 2:     compute $\sigma^2_{red,0}(x^*, model)$      ▷ Uncertainty reduction by querying at $x^*$ at level 0
 3:     $crit_0 \leftarrow \sigma^2_{red,0}(x^*, model)/costs[0]^2$
 4:     enrich Fidelity 0                   ▷ LF has to be computed because of nested DOEs
 5:     update datasets $X_0$ and $Y_0$
 6:     for $k \in \{1, \dots, l\}$ do
 7:         compute $\sigma^2_{red,k}(x^*, model)$
 8:         $crit_k \leftarrow \sigma^2_{red,k}(x^*, model)/(\sum_{i=0}^{k} costs[i])^2$
 9:         if $crit_k \geq crit_{k-1}$ or $\sigma^2_{red,k-1} \leq \epsilon$ then         ▷ $\epsilon$: machine resolution
10:             enrich Fidelity $k$
11:             update datasets $X_k$ and $Y_k$
12:         else
13:             break;
14:     return updated datasets
15:
16: procedure RUN(F, $n_{iter}, costs$)            ▷ Find the best minimum of F in $n_{iter}$ iterations
17:     while $i \leq n_{iter}$ do
18:         $mod \leftarrow$ model($\{X_0, \dots, X_l\}, \{Y_0, \dots, Y_l\}$)          ▷ Multi-fidelity surrogate model
19:         $f_{min} \leftarrow \min Y_l$
20:         $x_{i+1} \leftarrow$ EI($mod, f_{min}$)                           ▷ choose $x$ that maximizes EI
21:         ENRICH_LEVEL($mod, x_{i+1}, costs$)
22:         $i \leftarrow i + 1$
23:     $f_{min} \leftarrow \min Y_l$
24:     return $f_{min}$             ▷ This is the best known HF solution after $n_{iter}$ iterations
```

3.3 Algorithm

Now that the choice of level heuristic has been presented, we can summarize the proposed strategy in Algorithm 1. We note $\{X_0, \dots, X_l\}$ the DOEs of the fidelity levels 0 through l. $\{Y_0, \dots, Y_l\}$ are the corresponding responses.

As can be seen in line 19 of the MFEGO algorithm, we only update the value of f_{min} by the best HF value. That is because other fidelity codes and datasets are only considered to help MFEGO and are not the objective of optimization. By only updating the best solution when a high fidelity sample is requested, the optimization process is made more robust. Indeed, low-fidelity can only be used to reduce some amount of the expected improvement, which is an amount due to uncertainty (Exploration), rather than knowledge (Exploitation). This property ensures that MFEGO converges to the global optimum of the high-fidelity function (in the same sense that EGO converges to the global optimum of a function). Note that the criterion integrates the correlation of the fidelity levels and thus makes the algorithm more robust to cases where there is a poor correlation between fidelity levels. Indeed, when $\rho_k \to 0$, $\frac{\sigma^2_{red}(k,x^*)}{cost_{total}(k)^2} \to 0$, prompting the algorithm to move to the higher fidelities. It is important that the algorithm will not resample high-fidelity points as the EI at these points is zero. It will also not resample low-fidelity points as **the choice of level criterion** will become zero for the low fidelity points, pushing the algorithm to query the high fidelity ones, thus enhancing the model and solution.

3.4 Illustration on 1-D analytic problem

The proposed strategy is illustrated on the following 1-D analytic problem [7]:

$$f_{HF}(x) = (6x-2)^2 \times \sin(2(6x-2))$$
$$f_{LF}(x) = 0.5 f_{HF} + 10(x-0.5) - 5$$

In Figure 1 we represent graphically these two fidelity levels: f_{HF} is the expensive function and f_{LF} is the cheap one. We make the assumption that the cost ratio between the fidelity levels is 1/1000. We start the optimization with 3 high-fidelity samples and 6 low-fidelity ones. We define the low- and high-fidelity datasets as:

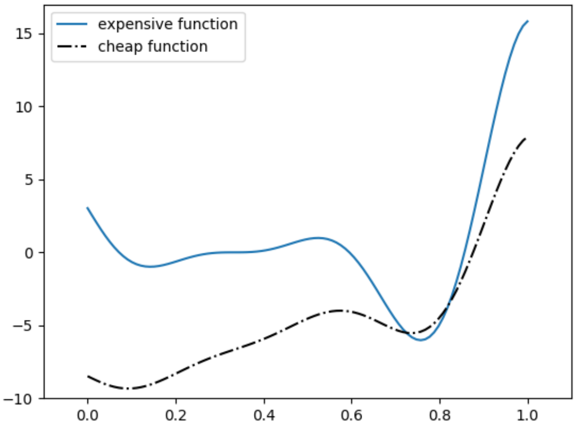


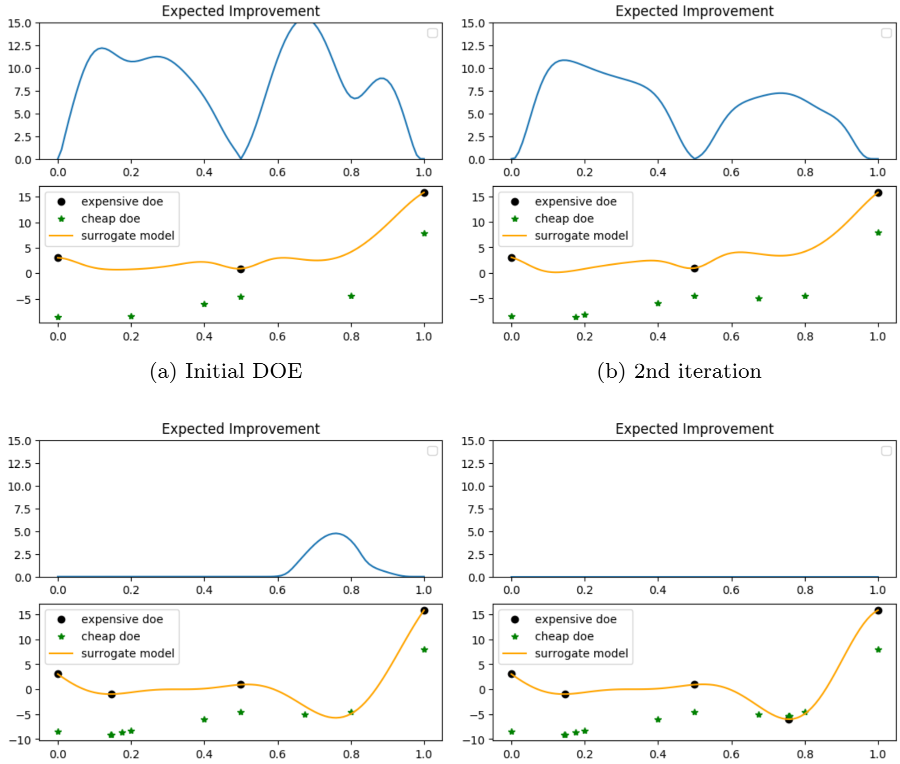
Fig. 1: Multi-fidelity of the 1-D analytic problem [7].

$$X_0 = \{x_1^{LF}, x_2^{LF}, x_3^{LF}, x_4^{LF}, x_5^{LF}, x_6^{LF}\} \text{ and } X_1 = \{x_1^{HF}, x_2^{HF}, x_3^{HF}\}.$$

The corresponding responses are Y_0 and Y_1 where $Y_0 = f_{LF}(X_0)$ and $Y_1 = f_{HF}(X_1)$. MFEGO algorithm is used to sequentially search through the space for the maximum of EI and enrich the multi-fidelity surrogate model. The algorithm has at each step two choices:

- query once the low-fidelity only,
- query once the low-fidelity and once the high-fidelity if the EI cannot be reasonably reduced by a low-fidelity query.

Figure 2 shows the reduction of EI resulting from cheap exploration, high-fidelity exploitation and model enhancement. After adding 4 LF points to explore the space and reduce EI (some low-fidelity points can be indistinguishable due to their proximity in the images above), Fig. 2-(c) shows the local exploitation and enhancement of the model by querying the high-fidelity code once. The next HF sample finds the global optimum of the function.



(a) Initial DOE

(b) 2nd iteration

(c) 4th iteration

(d) 7th iteration

Fig. 2: Evolution of EI throughout MFEGO iterations on the 1-D analytic problem [7].

Table 1: 1-D analytic problem [7] optimization summary.

	HF DOE	LF DOE	HF Opt	LF Opt	Cost
MFEGO	3	6	2	7	5.013
EGO	4	-	11	-	15

To summarize, we find the optimum after 7 iterations: 7 LF samples and 2 HF samples have been added, whereas the initial DOE was 6 LF points and 3 HF points (see Table 1). The classical mono-fidelity EGO approach with an initial DOE of 4 HF points required 11 additional HF points to find the global optimum, with a total cost three times higher than the MFEGO cost (15 compared to 5.013).

To finish, we highlight that the *choice of level criterion* can be used with any Exploration/Exploitation algorithm as the same idea can be applied. We use the low fidelity to reduce the uncertainty of the model and thus reduce the Exploration contribution to the *infill sampling criterion*. The highest fidelities can then be used for the Exploitation and effectively minimize the objective.

4 Airfoil shape optimization

To validate MFEGO, compare it to EGO and later gradient-based approaches on a more complex problem, we consider an unconstrained airfoil shape optimization with 15 design variables. Firstly, the test case is described and then the results are presented.

4.1 Test case description

For the airfoil shape optimization test case, we used a parametrization based on a mode decomposition proposed by [17]. This decomposition is a Singular Value Decomposition (SVD) on the camber and thickness of an airfoil database. We use this parametrization to define an airfoil geometry for which we can compute characteristics such as the Lift Coefficient C_l, Drag Coefficient C_d, and Pitching Moment C_m. The goal is to find the global optimal geometry of the airfoil as computed by the high-fidelity code ADflow. ADflow has a Reynolds Averaged Navier-Stokes (RANS) multi-block flow solver developed at the MDOlab (University of Michigan), [18,19]. In order to help the MFEGO algorithm to find the optimum at the lowest cost possible, we use a low-fidelity code called Xfoil [20], which takes a fraction (1/200) of the HF code time to give an approximation of the result using hypotheses simplifying the problem. We use the correlation between the low and high fidelities (Xfoil and ADflow) to improve the accuracy of the model and perform faster optimizations. It is worth mentioning that, thanks to the works of [21–23], ADflow is able to compute derivatives through a combination of automatic differentiation and adjoint method. This allowed us to have a gradient-based optimization reference to compare the Bayesian optimization approaches with, especially considering that 2D airfoil optimizations are unimodal [24]. We should also point out that to realize this optimization, work has been done on the noise estimation and regression/re-interpolation of the multi-fidelity surrogate model based on the work of [7].

4.2 15-D unconstrained optimization

We consider the optimization problem described in Table 2. To compare MFEGO and EGO, let us define the series '*Gain*' of a Bayesian optimization at iteration i as:

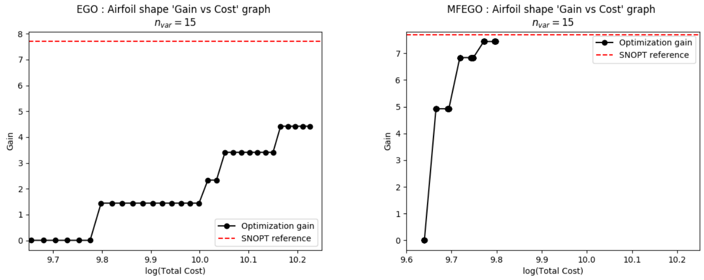
$$\text{Gain}_i = |sol_i - sol_{DOE}|$$

where sol_i is the best known solution after the i^{th} iteration and sol_{doe} is the best known solution resulting from the random sampling of the initial Design Of Experiment. We use the *Gain* to compare the efficiency of the EGO and MFEGO algorithms.

For this problem we had access to the gradient-based optimum using SNOPT [13] represented by the red dashed line in Figure 3. Figure 3 shows the *Gain* as a function of the cost compared with the SNOPT reference. We see that EGO

Table 2: Definition of the 15-D unconstrained optimization problem.

	Function/variable	Description	Quantity	Range
maximize	L/D	Lift-to-Drag ratio	1	
with respect to	α	Angle of attack	1	$[0.0, 8.0]$ (°)
	θ	Thickness modes	7	$[0, 1]$
	δ	Camber modes	7	$[0, 1]$
		Total variables	**15**	

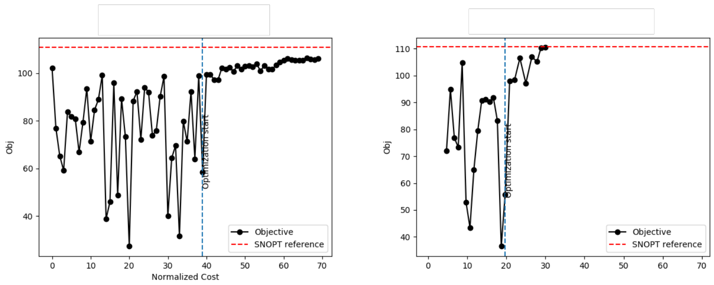


(a) EGO (b) MFEGO

Fig. 3: Comparison of '*Gain*' as a function of the cost for EGO and MFEGO for unconstrained L/D maximization (15 Design Variables). The red dashed line is the optimum found by SNOPT.

has long plateaus where the objective does not improve, whereas MFEGO improves at (almost) every HF call. This is due to the fact that EGO is an Exploitation/Exploration compromise. By using the lower-fidelity, MFEGO can explore the design space more cheaply, leaving HF calls for effective improvement of the objective. This observation is consistent across multiple runs. This is particularly important when increasing the dimension of the problem. Using multiple fidelities allows MFEGO to scale better than EGO w.r.t the number of design variables. One way to make further use of the multi-fidelity kriging is to reduce the cost of the initial DOE. For example, instead of using 40 HF (for a total cost of 40), we use 16 HF and (approximately) 800 LF (for a total equivalent cost of (approximately) 20). We see in Figure 4 that by doing so, the space is better mapped, and that this improves greatly the speed of convergence at a much lower cost. The dashed vertical blue line in Figure 4 separates the initial DOE building phase and the optimization driven by the Bayesian algorithm phase that comes after that.

Figure 4 shows that MFEGO gives better results than EGO even with a less expensive initial DOE. The low-fidelity, much cheaper, contributes some information to the surrogate model. MFEGO searched through the design space using 437 LF points and probed 8 HF points to attain the SNOPT solution (110.7 for SNOPT against 110.5 for MFEGO). We summarize the results in Table 3. The columns **HF DOE** and **LF DOE** in Table 3 denote the number of samples of each fidelity used to build the initial DOE. The columns **HF Opt** and **LF Opt** are respectively the number of high-fidelity (HF) and low-fidelity (LF) calls that the optimization routine used after that. The **Cost** is a cost normalized by the HF cost, so that HF calls contribute 1 to the Total Cost, and LF calls contribute 1/200. It can be



(a) EGO

(b) MFEGO

Fig. 4: Comparison of evolution of objective as function of iterations for EGO and MFEGO for unconstrained L/D maximization with 15 design variables. The red dashed line is the optimum found by SNOPT and the blue one marks the optimization starting point.

Table 3: L/D maximization: Comparison of cost and objective for EGO and MFEGO for the unconstrained optimization with 15 design variables.

	HF DOE	LF DOE	HF Opt	LF Opt	Cost	Obj
EGO	40	-	30	-	70	104.9
MFEGO	16	744	8	437	29.89	110.5
SNOPT	-	-	21	-	21	110.7

directly interpreted as the CPU time needed to reach the solution. We should note that EGO and MFEGO were stopped before 'convergence' (Bayesian optimization doesn't have a characterization of optimum, a budget of iterations is generally imposed as stopping criterion) and that at 'convergence', EGO and MFEGO should give the same optimum. In Table 3, the cost of SNOPT optimizations combines the number of direct problems solved (13), and the number of adjoint problems solved to compute derivatives ($13 \times 1 = 13$). The overall cost is then calculated as:

$$\mathrm{Cost}_{SNOPT} = \frac{time_{Direct} + time_{Adjoint}}{time_{Direct}} \times N_{iter}$$

where $time_{Direct}$ is the time needed for solving the N_{iter} direct problems, and $time_{Adjoint}$ is the time needed to solve $N_{iter} \times N_{funcs}$ adjoint problems. N_{funcs} is the number of functions evaluated, here 1: L/D for the objective. If constraints exist, then the number of evaluated functions goes up. N_{iter} was 13 in this case.

The calculation of the SNOPT cost allows us to compare it to the MFEGO and EGO costs. We see that with less than half the cost, MFEGO was much more efficient than EGO. However, SNOPT is approximately 40% faster than MFEGO, this is due to multiple factors that would not be present in a general case:

- the present problem, as said earlier, is unimodal. In a general case, the SNOPT cost would be multiplied by a chosen number of multistarts to counteract the multimodality of the problem,
- the present problem being not properly constrained (unconstrained optimization), the optimum is found on the edge of the design space, where the surrogate model is the least accurate. The surrogate model hence spends much more time learning the function in the zone of interest,

– the addition of constraints is expected to affect the cost of MFEGO and SNOPT in two ways that are favorable to MFEGO. Firstly, if the constraints and bounds are chosen properly, the optimum will move to the middle of the design space where the surrogate is more accurate. Secondly, with the addition of constraints, SNOPT will need, in addition to the direct computations, the adjoint solving for the constraints, whereas MFEGO only needs the direct computations.

5 Conclusion

In this paper we formulate a natural multi-fidelity extension for EGO. Optimization problems have been treated within the SEGOMOE framework. We prove, on a first promising high-dimensional test case (15 design variables), the benefits of multi-fidelity information sources, especially when it comes to improving the exploration of the design space. Future work will focus on extending and running the algorithm on constrained test cases possibly with multimodality (e.g aerodynamic shape optimization of a wing). We also think it will be beneficial to integrate the gradient information in the surrogate model and possibly in the devising of new criteria to allow for faster optimization times.

Acknowledgments

The author would like to thank the ISAE-Supaero Foundation for its financial support and its role in making this project possible. The author is also grateful to Dr. Bartoli, Dr. Lefèbvre from ONERA as well as Prof. J.R.R.A Martins from the University of Michigan and Prof. Morlier from ISAE-Supaero for their guidance. The author would also thank Rémi Lafage for his support with the SMT implementation and Rémy Priem for his support with the SEGOMOE framework.

References

1. Robert L. Winkler. Combining probability distributions from dependent information sources. *Manage. Sci.*, 27(4):479–488, April 1981.
2. Rémi Lam, Douglas L Allaire, and Karen E Willcox. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0143, 2015.
3. Matthias Poloczek, Jialei Wang, and Peter I. Frazier. Multi-information source optimization. In *NIPS*, 2017.
4. R. M. Lewis and S. G. Nash. A multigrid approach to the optimization of systems governed by differential equations. 2000.
5. D. Huang, T. T. Allen, W. I. Notz, and R. A. Miller. Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5):369–382, Nov 2006.
6. Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.
7. Alexander Forrester, Andras Sobester, and Andy Keane. Multi-fidelity optimization via surrogate modelling. 463:3251–3269, 12 2007.

8. Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, 1998.
9. Loic Le Gratiet. *Multi-fidelity Gaussian process regression for computer experiments.* Thesis, Université Paris-Diderot - Paris VII, October 2013.
10. Michael James Sasena. Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations, 2002.
11. N Bartoli, I Kurek, R Lafage, T Lefebvre, R Priem, MA Bouhlel, J Morlier, V Stilz, and R Regis. Improvement of efficient global optimization with mixture of experts: methodology developments and preliminary results in aircraft wing design. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, At Washington DC*, 2016.
12. Nathalie Bartoli, Thierry Lefebvre, Sylvain Dubreuil, Romain Olivanti, Nicolas Bons, Joaquim Martins, Mohamed-Amine Bouhlel, and Joseph Morlier. An adaptive optimization strategy based on mixture of experts for wing aerodynamic design optimization. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 4433, 2017.
13. Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
14. Alexander Forrester, András Sóbester, and Andy Keane. *Engineering design via surrogate modelling : a practical guide.* 2008.
15. Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.
16. J. Močkus. *On Bayesian Methods for Seeking the Extremum*, pages 400–404. Springer Berlin Heidelberg, Berlin, Heidelberg, 1975.
17. Jichao Li, Mohamed Amine Bouhlel, and Joaquim Martins. A data-based approach for fast airfoil analysis and optimization. 01 2018.
18. E. van der Weide, G. Kalitzin, J. Schluter, and J. J. Alonso. Unsteady turbomachinery computations using massively parallel platforms. In *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2006. AIAA 2006-0421.
19. Charles A. Mader and Joaquim R. R. A. Martins. Stability-constrained aerodynamic shape optimization of flying wings. *Journal of Aircraft*, 50(5):1431–1449, September 2013.
20. Mark Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In *Low Reynolds number aerodynamics*, pages 1–12. Springer, 1989.
21. L. Hascoët. Tapenade: A tool for automatic differentiation of programs. In *Proceedings of 4^{th} European Congress on Computational Methods, ECCOMAS'2004, Jyvaskyla, Finland*, 2004.
22. Charles A. Mader, Joaquim R. R. A. Martins, Juan J. Alonso, and Edwin van der Weide. ADjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA Journal*, 46(4):863–873, April 2008.
23. Zhoujie Lyu, Gaetan K. Kenway, Cody Paige, and Joaquim R. R. A. Martins. Automatic differentiation adjoint of the Reynolds-averaged Navier–Stokes equations with a turbulence model. In *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, Jul. 2013.
24. Oleg Chernukhin and David Zingg. Multimodality and global optimization in aerodynamic design. 51:1342–1354, 06 2013.