



Multi-fidelity design optimisation strategy under uncertainty with limited computational budget

Péter Zénó Korondi^{1,2} · Mariapia Marchi² · Lucia Parussini¹ · Carlo Poloni^{1,2}

Received: 10 February 2020 / Revised: 9 April 2020 / Accepted: 28 April 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

In this work, a design optimisation strategy is presented for expensive and uncertain single- and multi-objective problems. Computationally expensive design fitness evaluations prohibit the application of standard optimisation techniques and the direct calculation of risk measures. Therefore, a surrogate-assisted optimisation framework is presented. The computational budget limits the number of high-fidelity simulations which makes impossible to accurately approximate the landscape. This motivates the use of cheap low-fidelity simulations to obtain more information about the unexplored locations of the design space. The information stemming from numerical experiments of various fidelities can be fused together with multi-fidelity Gaussian process regression to build an accurate surrogate model despite the low number of high-fidelity simulations. We propose a new strategy for automatically selecting the fidelity level of the surrogate model update. The proposed method is extended to multi-objective applications. Although, Gaussian processes can inherently model uncertain processes, here the deterministic input and uncertain parameters are treated separately and only the design space is modelled with a Gaussian process. The probabilistic space is modelled with a polynomial chaos expansion to allow also uncertainties of non-Gaussian type. The combination of the above techniques allows us to efficiently carry out a (multi-objective) design optimisation under uncertainty which otherwise would be impractical.

Keywords Multi-fidelity learning · Gaussian process regression · Co-Kriging · Design optimisation under uncertainty

✉ Péter Zénó Korondi
PETERZENO.KORONDI@phd.units.it

¹ Department of Engineering and Architecture, University of Trieste, Piazzale Europa, 1, 34127 Trieste, Italy

² ESTECO S.p.A, Building B, 99 Padriciano, Area Science Park, 34149 Trieste, Italy

1 Introduction

Automated design optimisation requires performance evaluation of numerous designs (Bandler 1969). Optimisation strategies can be categorised as gradient-based and gradient-free (Skinner and Zare-Behtash 2018). Gradient-based optimisation uses the gradient information to identify designs with improved performance. When the analytical form of the gradients is not known, numerical gradient calculation increases the computational cost of the performance evaluation (Giering and Kaminski 1998). Moreover, gradient information guides the optimiser to local optima. To avoid this, gradient-based strategies can be launched from different points of the design space (multi-start strategy).

Gradient-free methods uses various heuristics or meta-heuristics to identify new designs with possibly better performance. Many successful gradient-free algorithms are inspired by nature and evaluate the performance of a population of designs at every iteration to reach global optima (Yang 2014).

Both approaches are impracticable when the performance evaluation of a design is computationally expensive and the available computational budget for the optimisation is strongly limited.

Surrogate-assisted optimisation is a commonly-used technique for the design optimisation of computationally expensive problems (Forrester et al. 2008). However, the number of attainable observations can be so sparse that standard surrogate techniques are not able to provide an accurate approximation. In this case, information from cheap low-fidelity (LF) design analyses can be extracted and fused together with the information of high-fidelity (HF) design analyses (Alexandrov et al. 2000). Depending on whether the fidelities can be ranked or not, one can do non-hierarchical information fusion (Matthias et al. 2017) or hierarchical information fusion (Kennedy and O'Hagan 2000; Le Gratiet 2013).

In computer aided engineering design optimisation, hierarchical information fusion gained popularity due to the fact that many numerical solvers inherently provide an approximation of their numerical error (Forrester et al. 2007; Han et al. 2013). This motivated the bi-fidelity hierarchical formulation of the Gaussian process regression (GPR) (Kennedy and O'Hagan 2000; Le Gratiet 2013) which is also called Co-Kriging. Co-Kriging was extended to fuse information from multiple levels of fidelity (Kennedy and O'Hagan 2000) which is called multi-fidelity Gaussian process regression (MF-GPR). The method in Kennedy and O'Hagan (2000) requires to compose and invert a large cross-correlation matrix containing the correlation information of all available fidelity levels. To avoid the construction and inversion of this large matrix, in Le Gratiet (2013) MF-GPR was reformulated in a recursive form. The recursive formulation allows combining any levels of fidelities by a sequential and hierarchical model construction. Instead of building the model by combining the different levels of fidelities in a single step, the multi-fidelity surrogate model is built by sequentially adding the fidelity levels to the model starting from the lowest fidelity to the highest. In this work, we employ the recursive MF-GPR technique as described in Le Gratiet (2013).

In surrogate-assisted design optimisation, the quality of the surrogate have high importance, particularly in the region of interest. Therefore, they are commonly employed in an adaptive optimisation framework. After training the surrogate on an initial design sample the surrogate model is adaptively refined. Following the taxonomy of Bayesian optimisation, this refinement is called acquisition and has two main goals: the surrogate is required to be accurate close to the optimum; and promising regions must be identified. These two goals are also called exploitation and exploration respectively. For single-objective problems this concept is realised in efficient global optimisation (EGO) by updating the GPR at locations where the expected improvement is maximal (Jones et al. 1998). When EGO is employed for multi-objective optimisation, a separate GPR is built for each objective and the objectives are aggregated into a single scalar measure (Knowles 2006; Tinkle et al. 2019). Then the acquisition function can identify the location of the new design candidate. By generating the surrogate on a multi-fidelity data set the acquisition function additionally selects the fidelity of the new design evaluation. In Mostafa et al. (2019) a strategy for selecting the fidelity is presented based on the ratio of the predicted variance reduction and cost of the new evaluation. Their method assumes a nested multi-fidelity data structure.

This paper formulates the fidelity selection criteria without assuming a nested data structure and extends its applicability to the multi-objective case. Additionally, a complete optimisation strategy is proposed for multi-objective design optimisation under uncertainty using a multi-fidelity surrogate. To investigate our optimisation strategy the multi-fidelity test suite proposed by Wang et al. (2017) and Habib et al. (2019) is extended for uncertain problems.

Section 2 introduces GPR and MF-GPR. In Sect. 3 the treatment of uncertainty is discussed. Section 4 describes the optimisation strategy and Sect. 5 the proposed acquisition function formulation. The set of benchmark functions is provided in Sect. 6. The obtained results are discussed in Sect. 7 and the conclusions are drawn in Sect. 8.

2 Multi-fidelity Gaussian process regression

2.1 Gaussian process regression

In design optimisation GPR, also called Kriging, is commonly used to predict the performance of a design. The performance of the design can be calculated by a function $z = f(\mathbf{x})$, with \mathbf{x} the d dimensional design variable, which is often unknown. Therefore, the performance function must be modelled based on some (empirical) observations. GPR models f as a random process \tilde{f} . Therefore, the performance of a design can be given by a random variable $\tilde{f}(\mathbf{x})$:

$$\tilde{f}(\mathbf{x}) = m(\mathbf{x}) + \tilde{\delta}(\mathbf{x}) + \epsilon, \quad (1)$$

where $m(\mathbf{x})$ is the mean trend and represents the global variation of the process. The mean trend is formulated as a least squares regression $m(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta}$, where $\mathbf{h}(\mathbf{x})$ is the vector of regression functions and $\boldsymbol{\beta}$ is the vector of regression coefficients. The $\tilde{\delta}$ is called local departure and represents local variations of the process (Rasmussen

2003). It is modelled as a Gaussian process (GP) with zero mean and a corresponding covariance function ζ , $\tilde{\delta} \sim \mathcal{GP}(0, \zeta)$, so that $\tilde{\delta}(\mathbf{x}) \sim \mathcal{N}(0, \sigma^2)$. The independent noise term $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is included in our model for numerical purposes and not for modelling the uncertainty as it assumes a Gaussian distribution for the performance distribution.

Given n observations, the covariance matrix Σ is an n -by- n symmetric matrix, with entries $\Sigma_{ij} = \zeta(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \Gamma(\mathbf{x}_i, \mathbf{x}_j; \theta)$, where σ^2 is the global process variance, and Γ is the parametric correlation function with hyperparameters θ .

The covariance function describes our prior knowledge on the problem. As our a priori knowledge is rarely complete, the covariance function is described by a parametric function. Its parameters are tuned based on our observations. Rasmussen (2003) provides suggestions on choosing the family of the correlation function. The tuning of its hyperparameters is described in Sect. 2.2.

Given n observations at design locations $\mathbf{x}^{(n)}$, the prediction $\tilde{f}(\mathbf{x}^*)$ at a new location \mathbf{x}^* is given by a probability distribution conditioned on the n evaluated design performance values $\mathbf{z}^{(n)}$ and the hyperparameters $\beta, \sigma^2, \sigma_\epsilon^2, \theta$. This is also called posterior distribution. Due to our Gaussian process assumption, the posterior distribution is a Gaussian distribution:

$$p(\tilde{f}(\mathbf{x}^*) | \mathbf{z}^{(n)}, \beta, \sigma^2, \sigma_\epsilon^2, \theta) \sim \mathcal{N}(\hat{m}(\mathbf{x}^*), \hat{s}^2(\mathbf{x}^*)), \quad (2)$$

where the mean $\hat{m}_Z(\mathbf{x}^*)$ and variance $\hat{s}_Z^2(\mathbf{x}^*)$ are:

$$\hat{m}_Z(\mathbf{x}^*) = \mathbf{h}(\mathbf{x}^*)\beta + \zeta(\mathbf{x}^*, \mathbf{x}^{(n)})^T (\Sigma + \sigma_\epsilon^2 \mathbf{I})^{-1} (\mathbf{z}^{(n)} - \mathbf{H}\beta), \quad (3)$$

$$\hat{s}_Z^2(\mathbf{x}^*) = \zeta(\mathbf{x}^*, \mathbf{x}^*) - \zeta(\mathbf{x}^*, \mathbf{x}^{(n)})^T (\Sigma + \sigma_\epsilon^2 \mathbf{I})^{-1} \zeta(\mathbf{x}^*, \mathbf{x}^{(n)}), \quad (4)$$

where $\zeta(\mathbf{x}^*, \mathbf{x}^{(n)})$ is the covariance vector of the new design and the evaluated designs and $\zeta(\mathbf{x}^*, \mathbf{x}^*)$ is the value of the variance of the new design. $\mathbf{H} = \mathbf{h}(\mathbf{x}^{(n)})$ is the matrix of regression functions evaluated at sample locations.

In surrogate-assisted design optimisation, the GPR mean is calculated in every point of a finite grid of the design space to obtain a prediction of the performance landscape. The prediction error of this landscape can be given by the GPR variance which provides useful information about how much confidence we can have in our prediction.

2.2 Parameter estimation of Gaussian process regression

The accuracy of the GPR model highly depends on the chosen covariance function (Rasmussen 2003). We assume stationarity, meaning that the covariance of two designs depends only on their distance measure and not on their positions in the design space. A popular covariance function family is the Matérn covariance functions (Rasmussen 2003). Its hyperparameters can be tuned by maximising their likelihood conditioned the observation. Assuming that \tilde{f} is a multivariate normal

distribution the GPR parameters $(\boldsymbol{\beta}, \sigma^2, \sigma_\epsilon^2, \boldsymbol{\theta})$ can be determined by the maximum likelihood estimation (MLE), where the likelihood function \mathcal{L} has the form:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\beta}, \sigma^2, \sigma_\epsilon^2, \boldsymbol{\theta} | \mathbf{z}^{(n)}) \\ = \frac{1}{(2\pi)^{n/2} \sqrt{|\boldsymbol{\Sigma} + \sigma_\epsilon^2 \mathbf{I}|}} \exp\left(-\frac{1}{2}(\mathbf{z}^{(n)} - \mathbf{H}\boldsymbol{\beta})^T (\boldsymbol{\Sigma} + \sigma_\epsilon^2 \mathbf{I})^{-1} (\mathbf{z}^{(n)} - \mathbf{H}\boldsymbol{\beta})\right). \end{aligned} \quad (5)$$

The generalised least squares estimate of $\boldsymbol{\beta}$ is given by:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{H}^T (\boldsymbol{\Sigma} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{H}\right)^{-1} \mathbf{H}^T (\boldsymbol{\Sigma} + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{z}^{(n)}, \quad (6)$$

The other parameters: σ^2 , σ_ϵ^2 and hyperparameters $\boldsymbol{\theta}$ are obtained by minimising the negative log-likelihood function:

$$\sigma^2, \sigma_\epsilon^2, \boldsymbol{\theta} = \arg \min_{\sigma^2, \sigma_\epsilon^2, \boldsymbol{\theta}} -\log(\mathcal{L}), \quad (7)$$

where the log-likelihood function $\log(\mathcal{L})$ is:

$$\begin{aligned} \log(\mathcal{L})(\sigma^2, \sigma_\epsilon^2, \boldsymbol{\theta}) = & -\frac{1}{2}n \log(2\pi) - \frac{1}{2} \log\left(|(\boldsymbol{\Sigma} + \sigma_\epsilon^2 \mathbf{I})|\right) \\ & - \frac{1}{2}(\mathbf{z}^{(n)} - \mathbf{H}\hat{\boldsymbol{\beta}})^T (\boldsymbol{\Sigma} + \sigma_\epsilon^2 \mathbf{I})^{-1} (\mathbf{z}^{(n)} - \mathbf{H}\hat{\boldsymbol{\beta}}), \end{aligned} \quad (8)$$

2.3 Multi-fidelity Gaussian process regression

MF-GPR can be framed in a recursive manner (Le Gratiet 2013). Information is ordered according to its fidelity level, where the rank of the lowest fidelity is $l = 1$ and the highest is $l = L$. The lowest fidelity information is modelled by a GPR and denoted by \tilde{f}_1 . Starting from the 2nd level, an auto-regressive model is then built at each fidelity level:

$$\begin{cases} \tilde{f}_l(\mathbf{x}) = \rho_{l-1}(\mathbf{x})\tilde{f}_{l-1}(\mathbf{x}) + \mathbf{h}_l^T(\mathbf{x})\boldsymbol{\beta}_l + \tilde{\delta}_l(\mathbf{x}) + \epsilon_l, \\ \tilde{f}_{l-1}(\mathbf{x}) \perp \tilde{\delta}_l(\mathbf{x}), \\ \rho_{l-1}(\mathbf{x}) = \mathbf{g}_{l-1}^T(\mathbf{x})\boldsymbol{\beta}_{\rho_{l-1}}, \end{cases} \quad (9)$$

where $\tilde{\delta}_l(\mathbf{x})$ is a Gaussian process with zero mean and covariance function $\sigma_l^2 \Gamma_l(\mathbf{x}, \mathbf{x}')$, and $\rho_{l-1}(\mathbf{s})$ represents a scale factor between $\tilde{f}_l(\mathbf{x})$ and $\tilde{f}_{l-1}(\mathbf{x})$. $\mathbf{g}_{l-1}(\mathbf{x})$ and $\mathbf{h}_l(\mathbf{x})$ are vectors of regression functions and $\boldsymbol{\beta}_{\rho_{l-1}}$ and $\boldsymbol{\beta}_l$ are the vectors of coefficients. The ϵ_l is an independent Gaussian noise term used for numerical regularisation. The symbol \perp means that $\tilde{f}_{l-1}(\mathbf{x})$ and $\tilde{\delta}_l(\mathbf{x})$ are assumed to be independent. This assumption is required to derive the Bayesian parameter estimation for the recursive MF-GPR (Le Gratiet 2013).

The recursive formulation of the Gaussian process $\tilde{f}_l(\mathbf{x})$ models the response at level l as a function of the previous level Gaussian process model \tilde{f}_{l-1} . The prediction at the l^{th} -level is given by a conditional probability:

$$p(\tilde{f}_l(\mathbf{x}^*)|\tilde{f}_{l-1}, \mathbf{z}_l^{(n_l)}, \boldsymbol{\beta}_l, \rho_{l-1}, \sigma_l^2, \sigma_{\epsilon_l}^2, \boldsymbol{\theta}_l) \sim \mathcal{N}(\hat{m}_l(\mathbf{x}^*), \hat{s}_l^2(\mathbf{x}^*)), \quad (10)$$

where $\mathbf{z}_l^{(n_l)}$ is the observation vector of n_l observations and the MF-GPR mean $\hat{m}_l(\mathbf{x}^*)$ is:

$$\hat{m}_l(\mathbf{x}^*) = \rho_{l-1}(\mathbf{x}^*)\hat{m}_{l-1}(\mathbf{x}^*) + \mathbf{h}_l^T(\mathbf{x}^*)\boldsymbol{\beta}_l + \boldsymbol{\zeta}_l^T(\boldsymbol{\Sigma}_l + \sigma_{\epsilon_l}^2\mathbf{I})^{-1}\mathbf{K}_l, \quad (11)$$

$$\mathbf{K}_l = (\mathbf{z}_l^{(n_l)} - \rho_{l-1}(\mathbf{x}_l^{(n_l)}) \odot \tilde{f}_{l-1}(\mathbf{x}_l^{(n_l)}) - \mathbf{H}_l\boldsymbol{\beta}_l), \quad (12)$$

and the MF-GPR variance is:

$$\begin{aligned} \hat{s}_l^2(\mathbf{x}^*) &= \rho_{l-1}^2(\mathbf{x}^*)\hat{s}_{l-1}^2(\mathbf{x}^*) \\ &+ \boldsymbol{\zeta}_l(\mathbf{x}^*, \mathbf{x}^*) - \boldsymbol{\zeta}(\mathbf{x}^*, \mathbf{x}_l^{(n_l)})^T(\boldsymbol{\Sigma} + \sigma_{\epsilon}^2\mathbf{I})^{-1}\boldsymbol{\zeta}(\mathbf{x}^*, \mathbf{x}_l^{(n_l)}). \end{aligned} \quad (13)$$

By comparing Eqs. (3)–(4) and Eqs. (11)–(13), we can see that the auto-regressive model solves a simple GPR problem at each level. The random process is the sum of the scaled model of the previous level and a difference term. Accordingly, the mean and the variance of the model are also the sums of the scaled mean and variance of the previous level model and the mean and variance of the difference term.

2.4 Parameter estimation of multi-fidelity Gaussian process regression

Similarly to the single level GPR, the parameters can be determined by the MLE. The likelihood function has the form:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\beta}_l, \rho_{l-1}, \sigma_l^2, \sigma_{\epsilon_l}^2, \boldsymbol{\theta}_l | \mathbf{z}_l^{(n_l)},) \\ = \frac{1}{(2\pi)^{n_l/2} \sqrt{|\boldsymbol{\Sigma}_l + \sigma_{\epsilon_l}^2\mathbf{I}|}} \exp\left(-\frac{1}{2}\mathbf{K}_l^T(\boldsymbol{\Sigma}_l + \sigma_{\epsilon_l}^2\mathbf{I})^{-1}\mathbf{K}_l\right). \end{aligned} \quad (14)$$

The generalised least squares estimate of $\boldsymbol{\beta}$ and ρ_{l-1} is given by:

$$\begin{aligned} \begin{bmatrix} \hat{\rho}_{l-1} \\ \hat{\boldsymbol{\beta}}_l \end{bmatrix} &= \left(\begin{bmatrix} \tilde{f}_{l-1}(\mathbf{x}_l^{(n_l)}) & \mathbf{H}_l \end{bmatrix}^T (\boldsymbol{\Sigma}_l + \sigma_{\epsilon_l}^2\mathbf{I})^{-1} \begin{bmatrix} \tilde{f}_{l-1}(\mathbf{x}_l^{(n_l)}) & \mathbf{H}_l \end{bmatrix} \right)^{-1} \\ &\quad \left(\begin{bmatrix} \tilde{f}_{l-1}(\mathbf{x}_l^{(n_l)}) & \mathbf{H}_l \end{bmatrix}^T (\boldsymbol{\Sigma}_l + \sigma_{\epsilon_l}^2\mathbf{I})^{-1} \begin{bmatrix} \tilde{f}_{l-1}(\mathbf{x}_l^{(n_l)}) \end{bmatrix} \right). \end{aligned} \quad (15)$$

The other parameters: σ_l^2 , $\sigma_{\epsilon_l}^2$ and hyperparameters $\boldsymbol{\theta}_l$ are obtained by minimising the negative log-likelihood function:

$$\sigma_l^2, \sigma_{\epsilon_l}^2, \theta_l = \arg \min_{\sigma_l^2, \sigma_{\epsilon_l}^2, \theta_l} -\log(\mathcal{L}), \quad (16)$$

where the log-likelihood function $\log(\mathcal{L})$ is:

$$\begin{aligned} \log(\mathcal{L}) & \left(\sigma_l^2, \sigma_{\epsilon_l}^2, \theta_l \right) \\ &= -\frac{n_l}{2} \log(2\pi) - \frac{1}{2} \log \left(\left| \left(\Sigma_l + \sigma_{\epsilon_l}^2 \mathbf{I} \right) \right| \right) - \frac{1}{2} \mathbf{K}_l^T \left(\Sigma_l + \sigma_{\epsilon_l}^2 \mathbf{I} \right)^{-1} \mathbf{K}_l. \end{aligned} \quad (17)$$

3 Uncertainty handling and risk measures

In engineering the designer can have control over only a limited number of variables. Therefore, the variations of the performance function, which can stem from unmodelled phenomena and uncertain environmental parameters, are taken into account to ensure the robust and reliable performance of real-world systems. In Beyer and Sendhoff (2007), uncertainty modelling techniques are grouped in three categories: deterministic, probabilistic and possibilistic. In our work, the uncertainty is treated probabilistically. We assume that all uncertainty can be determined by a random variable with a corresponding probability distribution function (PDF). Furthermore, we assume that the probability distributions of the environmental parameters are known. With the probabilistic assumption also the performance uncertainty is fully determined by its probability distribution. The performance PDF is usually unknown a priori or too costly to determine. Uncertainty quantification methods are handy to overcome this problem. However, decision making is difficult based on probability distributions due to their multitude of possible realisations (Rockafellar and Royset 2015). Therefore, various properties of the distribution, or risk measures \mathcal{R} , are used to quantify the uncertainty.

The analytical computation of any risk measure is typically intractable apart from some benign problems. Therefore, the calculation of most risk measures requires a statistically significant sample, which is impractical if the performance evaluation is computationally expensive. For this reason, surrogate-assisted uncertainty quantification can be used to replace the expensive performance evaluations with cheap evaluations of the surrogate model of the probabilistic space. In this way, a statistically significant sample can be achieved at low cost.

Ideally, the design and uncertainty space should be modelled together. However, the construction of a surrogate directly on the coupled space of design variables and uncertainty variables is computationally challenging due to the high dimensionality of the problem. Therefore, we separate the design and probability spaces. At each design sample location, we sample the uncoupled probabilistic space by changing only the uncertain variables and train a surrogate model of the probabilistic space. These local probabilistic models can be used to evaluate the high number of samples required for the computation of the uncertainty measure \mathcal{R} . With the above

described procedure, one can calculate the risk measure \mathcal{R} for each design at relatively low computational cost.

The domain separation is a critical step as it forces us to train separate surrogates for the probability space at each design sample location and an additional surrogate over the risk measure \mathcal{R} . This introduces additional numerical error due to the nested approximations but allows us to optimise a computationally heavy design problem under uncertainty.

The type of the surrogate models can be chosen deliberately depending on the problem. In our case a multi-fidelity surrogate is chosen to model the landscape of the risk measures \mathcal{R} as described in Sect. 2. The probability space is modelled by a Polynomial Chaos Expansion (PCE).¹ PCE is a very efficient technique and, depending on the polynomial family of the regression functions, can model the probability space of various probability distributions (Choi et al. 2004).

As risk measure, a coherent and regular measure should be chosen to avoid the dependency on scaling (Rockafellar and Royset 2015; Quagliarella et al. 2014). Therefore, we adopt the superquantile \bar{q}_ζ in this work:

$$\mathcal{R} = \bar{q}_\zeta = \mathbb{E}[\tilde{y} | \tilde{y} \geq q_\zeta(\tilde{y})] = \frac{1}{1-\zeta} \int_\zeta^1 q_\tau(\tilde{y}) d\tau, \quad (18)$$

where q_ζ is the quantile and ζ is the chosen superquantile level. The superquantile calculates the expected value of the random variable if its value is higher than a certain quantile. Consequently, $\zeta = 0$ corresponds to the mean value and $\zeta = 1$ corresponds to the worst case scenario. In engineering practice, the superquantile is chosen over the quantile when the shape of the probability distribution tail is relevant as in case of reliability-based optimisation (Rockafellar and Uryasev 2002).

3.1 Surrogate-assisted uncertainty propagation

In this section the surrogate-assisted uncertainty quantification is described by employing it on an academic example. Let us denote the input uncertainty with \tilde{u} and the probabilistic output with \tilde{y} . The performance function is denoted by g . Then the performance of a design x can be given by:

$$\tilde{y} = g(x, \tilde{u}), \quad (19)$$

Usually, the performance function is not known. However, in our academic example we can neglect the dependency on the deterministic design variables and imagine that:

$$\tilde{y} = \sinh(\tilde{u}). \quad (20)$$

¹ Polynomial Chaos Expansion can be considered as a particular subclass of Gaussian process regression with particular regression functions and Kronecker delta covariance function (Schobi et al. 2015).

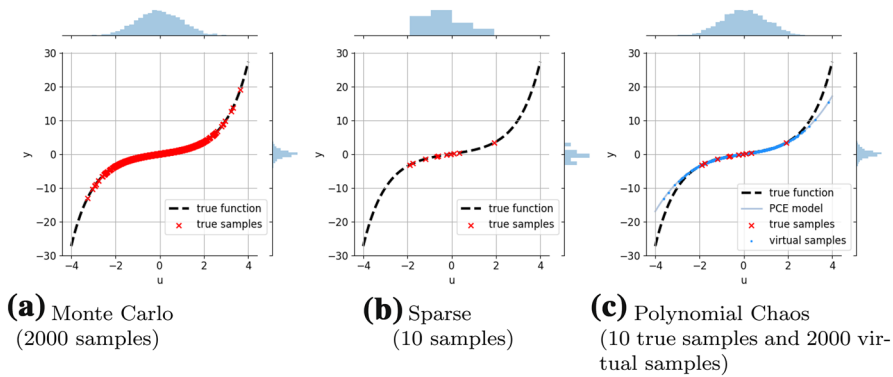


Fig. 1 Comparison of Monte Carlo sampling and virtual sampling with PCE. **a** Monte Carlo (2000 samples). **b** Sparse (10 samples). **c** Polynomial Chaos (10 true samples and 2000 virtual samples)

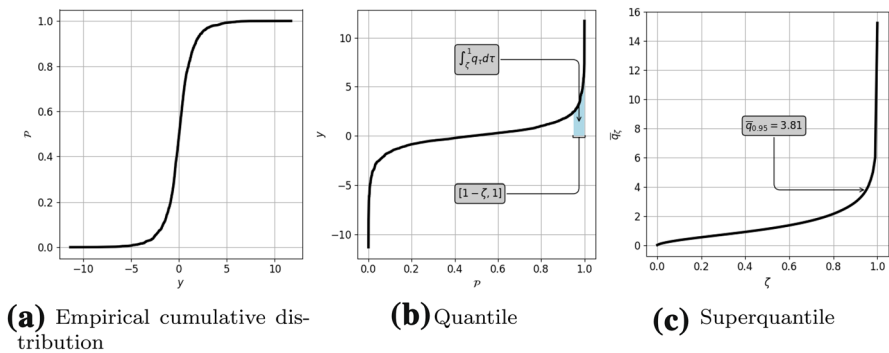


Fig. 2 Superquantile calculation. **a** Empirical cumulative distribution. **b** Quantile. **c** Superquantile

The distribution of the input is considered to be standard normal Gaussian $\mathcal{N}(0, 1)$. In order to accurately approximate the distribution of the output \tilde{y} we have to employ an uncertainty quantification method. The most straightforward solution would be to use Monte Carlo (MC) sampling (Lemieux 2009). However, the MC technique cannot be applied directly when the evaluation of g is expensive. Therefore, a PCE is built on a sparse sample with standard least angles regression techniques and the MC samples are obtained virtually by using the PCE as a surrogate model for g . The comparison between the direct sampling technique and surrogate-assisted uncertainty quantification is displayed in Fig. 1. We can see that PCE introduces some approximation error but it allows us to draw a high number of samples to approximate the distribution of \tilde{y} .

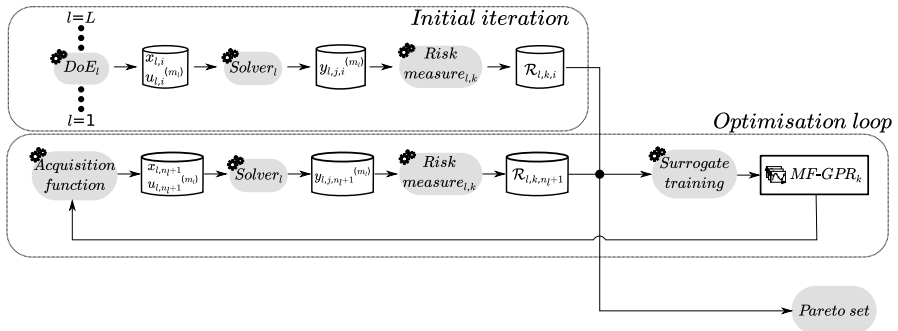


Fig. 3 Workflow of surrogate construction

After obtaining the distribution of \tilde{y} an empirical cumulative distribution function \mathcal{P} can be calculated, see Fig. 2a. By inverting \mathcal{P} we can obtain the quantile values of \tilde{y} . Then the superquantile \bar{q}_ζ can be obtained by Eq. (18). Imagine that we are interested in the $\bar{q}_{0.95}$ value. This requires the integration of the quantile curve between 0.95 and 1, as depicted in Fig. 2b. The superquantile values corresponding to each ζ level is given in Fig. 2c and the superquantile corresponding to level $\bar{q}_{0.95}$ is 3.81.

3.2 Implications for design optimisation of real-world systems

In the context of design optimisation, we are interested in one or more system responses or Quantities of Interest (QoIs). The uncertainties of input variables and environmental parameters can result in multiple uncertain system responses of the performance functions \tilde{y}_j (with $j = 1, \dots, n_q$), i.e. in multiple uncertain QoIs. These uncertainties have to be considered in the design optimisation process otherwise the obtained optimum will be biased. Thus, optimisation under uncertainty does not usually consider, as the deterministic, QoIs to be optimised, but rather risk measures \mathcal{R}_k (with $k = 1, \dots, n_r$) of the QoIs. Note that the number of QoIs n_q is not the same as the number of uncertainty measures n_r in the sense that various risk measures can be calculated from the distribution of a QoI. For example, one might be interested in optimising both the mean \bar{q}_0 and the $\bar{q}_{0.95}$ of a random valued QoI.

This results in a multi dimensional objective space. However, for the surrogate construction the objectives are treated separately and for each objective \mathcal{R}_k a surrogate model is constructed.²

² It is also possible to construct a multi-output GP. However, the training of a multi-output GP is challenging as a kernel should be found which can capture the correlation of both the input and output space (Press William et al. 2007).

4 Optimisation strategy

Surrogate-assisted design optimisation is employed when the computational burden of performance evaluations prohibits the optimisation to reach a global optimum. The construction of a surrogate model involves two steps: design sampling and model training. In our surrogate-assisted optimisation strategy these two steps are repeated until the computational budget is exhausted. The strategy is depicted in Fig. 3 and given by Algorithm 1.

The initial design sample is generated by employing a Design of Experiment (DoE) technique (Kuehl Robert and Kuehl 2000). In our assumption multiple solvers of different fidelities are available; hence, a design sample is generated for each fidelity level $\mathbf{x}_l^{(n_l)}$, where n_l is the number of samples generated at fidelity l . Moreover, the uncertain space is sampled as well. For each design sample $\mathbf{x}_{l,i}$ an additional sample is generated in the probabilistic space $\mathbf{u}_{l,i}^{(m_l)}$.

At each fidelity level, every design is evaluated m_l times resulting in a performance matrix with entries of $\mathbf{y}_{l,j,i}^{(m_l)}$, where i is the corresponding design index and j is the index of the QoI. To obtain the k^{th} objective \mathcal{R}_k the corresponding risk measure is calculated on the performance values $\mathbf{y}_{l,j,i}^{(m_l)}$. Repeating this for each design samples at each fidelity results in $\mathcal{R}_{l,k,i}$ multi-fidelity risk measure samples.

At this point the performance information coming from different fidelities is fused together resulting in the MF-GPR model. For each risk measure a MF-GPR model (MFGPR _{k}) is built. These models are cheap-to-evaluate approximations of our underlying problem and help us to identify design candidates. Due to the expensive performance evaluation the computational expense of the model training can be multiple magnitude orders smaller than the performance evaluation of a single design. Therefore, after the initial iteration the design sample is augmented by a single design with a corresponding probabilistic sample $\mathbf{x}_{l,n_l+1}, \mathbf{u}_{l,n_l+1}^{(m_l)}$. For the new design the corresponding objectives can be calculated as previously described resulting in \mathcal{R}_{l,k,n_l+1} . Then the surrogate models of the objectives are retrained on the augmented risk measure samples. We iterate this process until the computational budget is exhausted and then return the Pareto set of evaluated designs.

Algorithm 1 Multi-fidelity Optimisation Strategy

Input: d, L, n_l, n_q, n_r, m_l ▷ Problem and model properties
Output: $\mathbf{x}_{L,pareto}, \mathcal{R}_{L,1:n_r,pareto}, \text{MFGPR}_{1:n_r}$

```

1: for  $l=1:L$  do
2:    $\mathbf{x}_l \leftarrow \text{generateDesignSample}()$ 
3:   for  $i=1:n_l$  do
4:      $\mathbf{u}_{l,i} \leftarrow \text{generateProbabilisticSample}(\mathbf{x}_{l,i})$  ▷ The size of  $\mathbf{u}_{l,i}$  is  $[1 \times m_l]$ 
5:      $\mathbf{y}_{l,1:n_q,i} \leftarrow \text{evaluateDesignPerformance}(\mathbf{x}_{l,i}, \mathbf{u}_{l,i})$ 
6:      $\mathcal{R}_{l,1:n_r,i} \leftarrow \text{evaluateRiskMeasure}(\mathbf{y}_{l,1:n_q,i})$ 
7:   end for
8: end for
9: for  $k=1:n_r$  do
10:   $\text{MFGPR}_k \leftarrow \text{trainMFGPR}(\mathcal{R}_{1:L,k,1:n_l})$  ▷ MFGPR model for each risk measure
11: end for
12:  $flag \leftarrow True$ 
13: while  $flag$  do
14:   $\mathbf{x}_{l,n_l+1} \leftarrow \text{AcquisitionFunction}(\text{MFGPR}_{1:n_r})$ 
15:   $\mathbf{u}_{l,n_l+1} \leftarrow \text{generateProbabilisticSample}(\mathbf{x}_{l,n_l+1})$ 
16:   $\mathbf{y}_{l,1:n_q,n_l+1} \leftarrow \text{evaluateDesignPerformance}(\mathbf{x}_{l,n_l+1})$ 
17:   $\mathcal{R}_{l,1:n_r,n_l+1} \leftarrow \text{evaluateRiskMeasure}(\mathbf{y}_{l,n_l+1})$ 
18:  for  $k=1:n_r$  do
19:     $\text{MFGPR}_k \leftarrow \text{trainMFGPR}([\mathcal{R}_{1:L,k,1:n_l}, \mathcal{R}_{l,k,n_l+1}])$ 
20:  end for
21:  if  $\text{budgetExhausted}$  then
22:     $flag \leftarrow False$ 
23:  else
24:     $\mathbf{x}_{L,pareto}, \mathcal{R}_{L,1:n_r,pareto} \leftarrow \text{findParetoOptimalDesigns}(\mathbf{x}_{L,:}, \mathcal{R}_{L,:,:})$ 
25:  end if
26: end while
27: return  $\mathbf{x}_{L,pareto}, \mathcal{R}_{L,1:n_r,pareto}, \text{MFGPR}_{1:n_r}$ 

```

5 Acquisition function and fidelity level selection

An acquisition function defines the information gain which can be obtained by a new observation at a location \mathbf{x}_{n+1} . Therefore, we can use this function to guide our optimisation algorithm where to sample next.

As discussed in Sect. 2.1, a GPR surrogate is initially trained on $\mathbf{x}_{1:n}$ observations. GPR comes with the advantage that the model error can be predicted by the GP variance. Provided this, our acquisition function aims at two goals: (a) to improve on the current best objective value and (b) to reduce the model variance at locations where the optimum can be possibly located.

For single objective minimisation³ problems, the Expected Improvement (EI) and for multi-objective problems the HyperVolume Improvement of Lower Confidence Bound (HVI_{LCB}) are discussed in the following.

³ For maximisation the appropriate signs must be changed.

5.1 Expected improvement

The Expected Improvement (*EI*) calculates the expected value of the improvement on the current best evaluation by adding a new observation (Keane 2006). The magnitude of the improvement is:

$$\tilde{I}(\mathbf{x}_{n+1}) = \max(0, z_{\min} - \tilde{z}(\mathbf{x}_{n+1})), \quad (21)$$

where $z_{\min} = \min_{x_i=1, \dots, n} z(\mathbf{x}_i)$ is the current best evaluation. The $\tilde{z}(\mathbf{x}_{n+1})$ is not known a priori. Therefore it is replaced by a Gaussian random variable with a mean $\hat{m}(\mathbf{x}_{n+1})$ and standard deviation $\hat{s}^2(\mathbf{x}_{n+1})$ as defined by Eqs. (11) and (13) respectively. Consequently, the improvement \tilde{I} is also random. The probability of the improvement can be calculated (Emmerich et al. 2006):

$$P[\tilde{I}(\mathbf{x}_{n+1})] = P[\tilde{z}(\mathbf{x}_{n+1}) \leq z_{\min}] \quad (22)$$

$$= \Phi\left(\frac{z_{\min} - \hat{m}(\mathbf{x})}{\hat{s}^2(\mathbf{x})}\right), \quad (23)$$

where Φ is the cumulative density function of the standard normal distribution. The argument of the cumulative density function is called germ and denoted by ξ :

$$\xi(\mathbf{x}) = \frac{z_{\min} - \hat{m}(\mathbf{x})}{\hat{s}^2(\mathbf{x})}, \quad (24)$$

The expectation of the improvement $\mathbb{E}[\tilde{I}(\mathbf{x}_{n+1})]$ can be marginalised out and given in closed form (from now on denoted by *EI*):

$$EI(\mathbf{x}_{n+1}) = \hat{s}^2(\mathbf{x}_{n+1}) \left(\xi(\mathbf{x}_{n+1}) \Phi(\xi(\mathbf{x}_{n+1})) + \phi(\xi(\mathbf{x}_{n+1})) \right), \quad (25)$$

where ϕ is the probability density function of the standard normal distribution. The location of the new observation can be determined by maximising the *EI*(\mathbf{x}).

$$\mathbf{x}_{n+1} = \max_{\mathbf{x}} EI(\mathbf{x}). \quad (26)$$

The maximum *EI* acquisition function has proven its efficiency in a wide range of applications.

5.2 HyperVolume improvement of lower confidence bound

In multi-objective optimisation the one-step lookahead strategy requires to aggregate the objectives into a single measure (Lam et al. 2018). Similarly to the single-objective case, an acquisition function can be used to determine the location of the new observation. Here, the HyperVolume Improvement (*HVI*) is adopted to scalarise the multi-objective problem. *HVI* is a popular technique due to the fact that the HyperVolume (*HV*) measure is a reliable performance indicator which does not

require any assumption on the true Pareto set (Emmerich et al. 2006). *HVI* is defined as:

$$HVI = HV(\mathcal{P} \cup \hat{y}_{n+1}) - HV(\mathcal{P}), \quad (27)$$

where the hypervolume *HV* is a Lebesgue-measure in the objective space and \mathcal{P} denotes the current Pareto set. The next observation is made where the predicted mean value of the GPR maximises the *HVI* measure. However, this strategy does not consider the variance of the prediction but only exploits the input space. To introduce some exploratory behaviour into the strategy, the *HVI* of the Lower Confidence Bound (LCB) of the prediction can be maximised instead as suggested by Emmerich et al. (2006):

$$\mathbf{x}_{n+1} = \max_{\mathbf{x}} HVI_{LCB}(\mathbf{x}), \quad (28)$$

$$y_{LCB}(\mathbf{x}) = \mu(\mathbf{x}) - \alpha\sigma(\mathbf{x}), \quad (29)$$

where α is a positive parameter that represents how optimistic our prediction is considering uncertainty. $\alpha = 2$ as suggested in Emmerich et al. (2006).

5.3 Fidelity level selection

In our optimisation workflow the surrogate construction is made on a multi-fidelity data set. This requires an acquisition strategy which determines also the fidelity of the next observation and not only its location. We propose a strategy that, first, determines the location of the next observation with one of the above-mentioned functions and then determines at which level of fidelity the new observation should be carried out. To determine the fidelity of the new observation, we can define the amount of information which can be gained for each fidelity level by performing the observation at that level. As a gain measure, we use the predicted variance reduction. This gain can be scaled with the observation cost c_l of the fidelity level l . By comparing the scaled costs of the different fidelities, we can choose the fidelity level which provides the highest *Scaled Expected Variance Reduction* (SEVR):

$$l = \max_{1:L} SEVR_l, \quad (30)$$

where the variance reduction $\tilde{\sigma}_l$ can be recursively defined as:

$$\tilde{\sigma}_1 = \frac{\rho_1 \hat{s}_{Z_1}^2(\mathbf{x}_{new})}{c_1}, \quad \text{for } l = 1, \quad (31)$$

$$\tilde{\sigma}_l = \frac{\rho_{l-1} \hat{s}_{Z_{l-1}}^2(\mathbf{x}_{new}) + \hat{s}_{\delta_l}^2(\mathbf{x}_{new})}{c_l}, \quad \text{for } l > 2. \quad (32)$$

The SEVR is defined as:

$$\text{SEVR}_l = \frac{\tilde{\sigma}_l}{c_l}. \quad (33)$$

The pseudo code of the acquisition function for single-objective problems with fidelity level selection is given in Algorithm 2.

Algorithm 2 Acquisition function for single-objective optimisation

Input: MFGPR_L

Output: x_{l,n_l+1}

```

1:  $\hat{m}_L \leftarrow \text{predictTrend}(\text{MFGPR}_L)$ 
2:  $\hat{s}_L^2 \leftarrow \text{predictVariance}(\text{MFGPR}_L)$ 
3:  $EI(\mathbf{x}) \leftarrow \text{calculateExpectedImprovement}(\mathcal{R}_{min}, \hat{m}_L(\mathbf{x}), \hat{s}_L^2(\mathbf{x}))$ 
4:  $(\mathbf{x}_{?,n_l+1}) \leftarrow \max EI(\mathbf{x})$ 
5: for  $l=1:L$  do
6:    $\tilde{\sigma}_l \leftarrow \text{predictVarianceReduction}(\text{MFGPR}_l, (\mathbf{x}_{?,n_l+1}))$ 
7:    $c_l \leftarrow \text{getCost}(l)$ 
8:    $\text{SEVR}_l \leftarrow \text{calculateSEVR}(\tilde{\sigma}_l, c_l)$ 
9: end for
10:  $l \leftarrow \text{argmax}(\text{SEVR}_l)$ 
11: return  $x_{l,n_l+1}$ 

```

The pseudo code of the acquisition function for multi-objective problems with fidelity level selection is given in Algorithm 3.

Algorithm 3 Acquisition function for multi-objective optimisation

Input: MFGPR_L

Output: x_{l,n_l+1}

```

1: for  $k=1:n_r$  do
2:    $\hat{m}_{L,k} \leftarrow \text{predictTrend}(\text{MFGPR}_{L,k})$ 
3:    $\hat{s}_{L,k}^2 \leftarrow \text{predictVariance}(\text{MFGPR}_{L,k})$ 
4: end for
5:  $HVI_{LCB}(\mathbf{x}) \leftarrow \text{calculateHyperVolumeImprovement}(\mathcal{R}_{pareto}, \hat{m}_{L,1:k}(\mathbf{x}), \hat{s}_{L,1:k}^2(\mathbf{x}))$ 
6:  $(\mathbf{x}_{?,n_l+1}) \leftarrow \max HVI_{LCB}(\mathbf{x})$ 
7: for  $l=1:L$  do
8:    $\tilde{\sigma}_l \leftarrow \text{predictVarianceReduction}(\text{MFGPR}_l, (\mathbf{x}_{?,n_l+1}))$ 
9:    $c_l \leftarrow \text{getCost}(l)$ 
10:   $\text{SEVR}_l \leftarrow \text{calculateSEVR}(\tilde{\sigma}_l, c_l)$ 
11: end for
12:  $l \leftarrow \text{argmax}(\text{SEVR}_l)$ 
13: return  $x_{l,n_l+1}$ 

```

6 Multi-fidelity/multi-objective and uncertain benchmarks

The mathematical formulation of the MF-GPR allows the combination of any level of fidelities. Here we limit ourselves to investigate only two-level models. A test suite for MF single objective optimisation was proposed by Wang et al. (2017). The test suite was extended by Habib et al. (2019) for multi-objective problems. We

adopt their work with minor modifications and also extend the benchmark tests with a low-fidelity model which combines residual and stochastic errors at the same time. We also extend the test functions by introducing a probabilistic term on both the high- and low-fidelity level to emulate uncertain problems in Sect. 6.1.

The exact deterministic function of the underlying problem is denoted by $f(x)$. Any approximation of $f(x)$ can be generally described as the following:

$$\bar{f}(x, \phi) = f(x) + e(x, \phi). \quad (34)$$

where the approximation $\bar{f}(x, \phi)$ is constructed as sum of the exact function $f(x)$ plus some error term $e(x, \phi)$. The error term depends on the accuracy level ϕ of the approximation. The error term is the biggest when $\phi = 0$ and when $\phi = 10$ the error term becomes zero and the approximation provides exact results.

We use the following modified Rastrigin function for testing the single-objective optimisation:

$$f(x) = \left(d + \sum_{j=1}^d [(x_j - 0.5)^2 - \cos(20\pi(x_j - 0.5))] \right), \quad (35)$$

where the number of design variables d can be chosen arbitrarily.

For multi-objective optimisation we use the DTLZ3 (Deb et al. 2005) test function. The number of objective functions can be chosen arbitrarily with the restriction that $n_{obj} \geq 2$ and $d \geq n_{obj}$.

$$f(x) = DTLZ3. \quad (36)$$

For the error term $e(x, \phi)$, three types are considered: *residual*, *stochastic*, *instability error*. The error terms represent the most common error characteristics of real-life approximation models (Wang et al. 2017).

The residual error term takes the following form:

$$e_{res}(x, \phi) = \sum_{j=1}^d [\alpha(\phi) \cos(\omega(\phi)x_j + \beta(\phi) + \pi)], \quad (37)$$

$$\alpha(\phi) = (1 - 0.1\phi), \quad (38)$$

$$\omega(\phi) = 10\pi\alpha(\phi), \quad (39)$$

$$\beta(\phi) = 0.5\pi\alpha(\phi), \quad (40)$$

where d is the number of deterministic design variables.

The stochastic error term takes the following form:

$$e_{stoc}(x, \phi) = \mathcal{N}(\mu_\epsilon, \sigma_\epsilon), \quad (41)$$

$$\mu_\epsilon(x, \phi) = \frac{1}{10d} v(\phi) \gamma(x), \quad (42)$$

$$\sigma_\epsilon(\phi) = 0.1 v(\phi), \quad (43)$$

$$v(\phi) = 1 - 0.1 \phi, \quad (44)$$

$$\gamma(x) = \sum_{j=1}^d (1 - |x_j|), \quad (45)$$

The instability error term takes the following form:

$$e_{ins}(\phi) = \begin{cases} 10d, & \text{if } r \leq \sigma_\epsilon(\phi), \\ 0, & \text{if } r > \sigma_\epsilon(\phi), \end{cases} \quad (46)$$

where r is a random number between 0 and 1.

6.1 Multi-fidelity test problems under uncertainty

To emulate uncertain problems, the single-objective test function is extended by introducing a probabilistic term $g(x, u, \phi_u)$ on both fidelity levels:

$$f(x, u, \phi_u) = f(x) + g(x, u, \phi_u), \quad (47)$$

$$\bar{f}(x, u, \phi, \phi_u) = f(x) + g(x, u, \phi_u) + e(x, \phi), \quad (48)$$

where the exact test $f(x, u)$ depends on both controllable design x and uncontrollable uncertain variables u . The level of stochasticity is determined by ϕ_u . When $\phi_u = 0$ the problem reduces to the previously described deterministic case.

The probabilistic term $g(x, u, \phi_u)$ depends on both design and uncertain variables:

$$g(x, u, \phi_u) = \sum_{i=1}^{d_u} \left[\alpha(\phi_u) \cosh \left(\frac{1}{10} (2 - \gamma(x)) \pi u_i \right) \right], \quad (49)$$

$$\alpha(\phi_u) = 0.1 \phi_u, \quad (50)$$

$$\gamma(x) = \sum_{j=1}^d (1 - |x_j - x_j^*|), \quad (51)$$

where d_u is the number of uncertain input parameters and $x_j^* = 0.4$ is the value of the j^{th} design vector where the uncertainty is smallest. The deterministic function $f(x)$ is the same as described previously by Eq. (35).

To extend the multi-objective *DTLZ3* function with uncertainty, we add the $g(u, x, \phi_u)$ term to the approach function $h(x)$ resulting in the following equation:

$$f_1(x, u, \phi_u) = \cos\left(x_1 \frac{\pi}{2}\right)(1 + \rho(x_M)), \quad (52)$$

$$f_2(x, u, \phi_u) = \sin\left(x_1 \frac{\pi}{2}\right)(1 + \rho(x_M)), \quad (53)$$

$$\bar{f}_k(x, u, \phi, \phi_u) = f_k(x) + g(x, u, \phi_u) + e(x, \phi), \quad (54)$$

$$\rho(x_M) = \left(d_M + \sum_{j=1}^{d_M} [(x_j - 0.5)^2 - \cos(20\pi(x_j - 0.5))] \right), \quad (55)$$

where 2 objectives are considered.

7 Results

7.1 Comparison of single- and multi-fidelity surrogates

In this section the surrogate model quality of single- and multi-fidelity GPR is compared for the deterministic case. When to use MF surrogates is still an open question (Fernández-Godino et al. 2019). Multi-fidelity models are more complex than their single-fidelity counterparts. Hence, the comparison between the two techniques is not straightforward. The MF-GPR has more parameters than standard GPR and therefore the relative performance of the two techniques depends on the extra parameters of the MF-GPR. In Wang et al. (2017) the shape and correlation properties of MF benchmark test functions were investigated. In the following, we discuss the performance of MF-GPR on MF benchmark test functions.

The parameter settings of the used test cases are described in Table 1. Table 2 compares the accuracy of GPR and MF-GPR considering the different error types of the low-fidelity level defined in Sect. 6. The accuracy measure $\text{Error}_{\text{avg}}$ is the mean absolute error calculated on 2000 points generated with a Latin Hypercube sampling. In these simulations the computational budget was fixed to 300 units, which corresponds exactly to 30 high-fidelity samples. The cost of a high-fidelity calculation is 100 and the cost of a low-fidelity calculation is 50. Moreover, for the MF-GPR model the computational budget was divided in two equal parts, i.e. 150 and 150, to obtain high- and low-fidelity samples.

Figure 4a shows that the landscape cannot be accurately modelled by a standard GPR because of the limited number of samples. By obtaining information from low-fidelity models a better approximation can be achieved as shown in Fig. 4b–e. The MF-GPR model uses half HF samples w.r.t. classical GPR, but the high number of LF samples provides more valuable information about the

Table 1 List of test cases for deterministic single-objective problems

Test case	High-fidelity		Low-fidelity		
	$f(x)$	$cost_{HF}$	$e(x, \phi)$	ϕ	$cost_{LF}$
TD0	Eq. (35) + Eq. (49)	10	–	–	–
TD1	Eq. (35) + Eq. (49)	10	Eq. (37)	5	5
TD2	Eq. (35) + Eq. (49)	10	Eq. (41)	5	5
TD3	Eq. (35) + Eq. (49)	10	Eq. (46)	5	5
TD4	Eq. (35) + Eq. (49)	10	Eq. (37) + Eq. (41)	5	5
Input					
x	$0 \leq x \leq 1, \quad d = 1$				

Table 2 Comparison of single- and multi-fidelity GPR for deterministic problems

Test case	Budget	LF error type	Surrogate	Samples	$Error_{avg}$
TD0	300	–	GPR	30(HF)	1.522e-1
TD1	300	Residual	MF-GPR	15(HF) 30(LF)	5.480e-2
TD2	300	Stochastic	MF-GPR	15(HF) 30(LF)	4.419e-2
TD3	300	Instability	MF-GPR	15(HF) 30(LF)	1.545e-1
TD4	300	Residual + stochastic	MF-GPR	15(HF) 30(LF)	5.930e-2

landscape. We can see that the error type of the low-fidelity model can have a significant effect on the surrogate quality. The MF-GPR performs the best when the LF model has only stochastic errors and performs the worst when the LF model has instabilities. GPR is able to handle instabilities only up to the magnitude of the regularising term σ_e in Eq. (3), which is also called nugget, whereas higher instabilities can cause severe approximation errors. In a MF-GPR the low-fidelity level approximation error is propagated to the higher fidelity level.

In the previous simulations the total computational budget was fixed to 300. The size of the computational budget has a significant effect on the surrogate quality. Figure 5 shows that the relative performance of GPR and MF-GPR changes depending on the size of the total budget. When the budget is too small, the available samples are not even enough to approximate well the low-fidelity model. This causes an even higher approximation error at the top-fidelity level. In this case, the samples from the low-fidelity model introduce a larger error than the amount of information they provide. Similarly, when the budget is big enough to have enough HF samples to accurately approximate the landscape, the low-fidelity deteriorate the multi-fidelity approximation. Consequently, in these

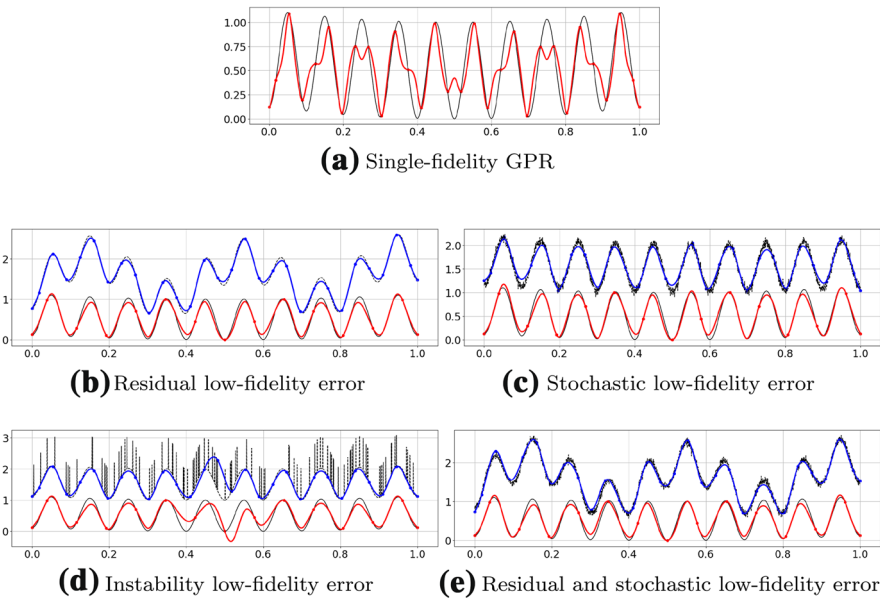


Fig. 4 Landscapes of the single- and multi-fidelity surrogate models. (In the plots for the sake of better visibility the low-fidelity curves are shifted up by 1.). **a** Single-fidelity GPR, **b** residual low-fidelity error, **c** stochastic low-fidelity error, **d** instability low-fidelity error, **e** residual and stochastic low-fidelity error

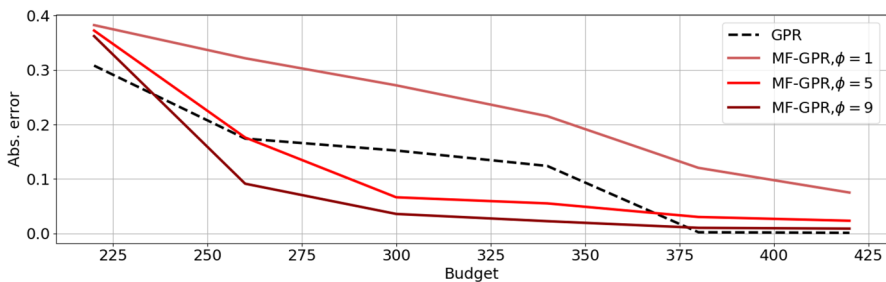


Fig. 5 Comparison of single- and multi-fidelity GPR for single-objective deterministic case. The lines represent the absolute errors of the surrogate models calculated at 2000 locations sampled by latin hyper-cube sampling. The black dashed line corresponds to the GPR model; the three differently shaded red lines correspond to the three error levels of the low-fidelity model. (Color figure online)

domains MF-GPR is not useful. MF-GPR provides significantly better results when the available budget is not able to provide enough HF samples. In this case, instead of a handful of HF samples, many LF samples can be generated which can provide valuable information for the landscape approximation. Consequently, we recommend to use MF-GPR for such a problem setting.

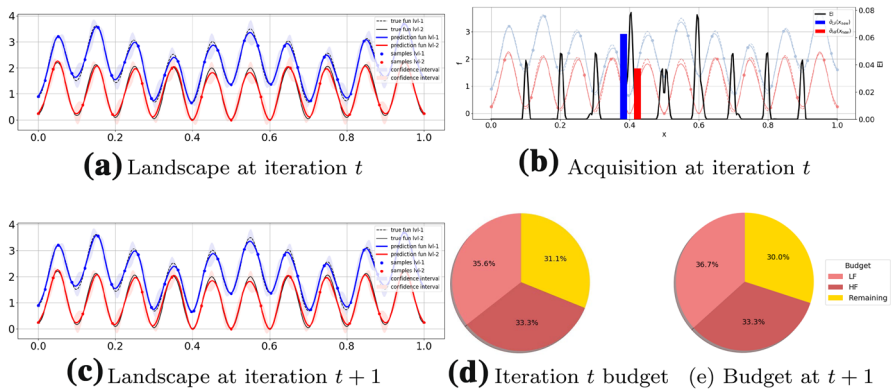
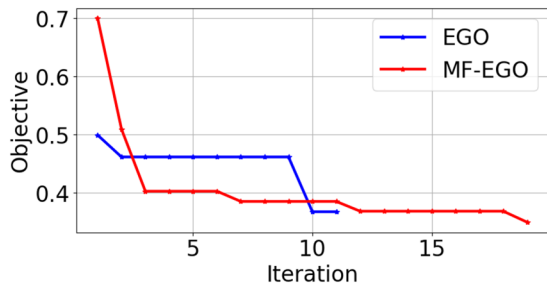


Fig. 6 Iteration step of multi-fidelity EGO. **a** Landscape at iteration t . **b** Acquisition at iteration t . **c** Landscape at iteration $t+1$. **d** Iteration t budget. **e** Budget at $t+1$

Fig. 7 Comparison of single- and multi-fidelity surrogate-assisted optimisation convergence. (The convergence histories were obtained by a trial run of TU2 with a budget of 4000.)



7.2 Iteration of a multi-fidelity surrogate-based optimisation

This section illustrates the internal processes and performance of our strategy through its application to the single-objective deterministic MF test-case of Eq. (35). To illustrate Algorithm 1 and Algorithm 3, the progression from iteration t to $t+1$ is shown in Fig. 6a–e. In Fig. 6a the predicted low- and high-fidelity landscapes are plotted. At iteration t the MF-GPR was trained on 15 HF and 31 LF samples. The EI of this prediction was calculated resulting in the black curve plotted in Fig. 6b. The maximum of the EI was located at $x = 0.405$. At this location, the SEVR were calculated and compared. The low cost of the LF evaluation resulted in a higher SEVR. Therefore a LF fidelity sample was generated and the MF-GPR surrogate was retrained. The new landscape prediction at $t+1$ is shown in Fig. 6c. Figure 6d, e shows the proportions of the total budget spent on low- and high-fidelity samples at iteration t and $t+1$ respectively.

7.3 Multi-fidelity surrogate-based optimisation under uncertainty

Table 3 summarises the uncertain single-objective test problems and their parameter settings.

Table 3 List of test cases for uncertain single-objective problems

	High-fidelity			Low-fidelity		
Test case	$f(x)$	ϕ_u	$cost_{HF}$	$e(x, \phi)$	ϕ	$cost_{LF}$
TU0	Eq. (35) + Eq. (49)	–	10	–	–	–
TU1	Eq. (35) + Eq. (49)	5	10	Eq. (37) + Eq. (41)	1	5
TU2	Eq. (35) + Eq. (49)	5	10	Eq. (37) + Eq. (41)	5	5
TU3	Eq. (35) + Eq. (49)	5	10	Eq. (37) + Eq. (41)	9	5
Inputs						
x	$0 \leq x_j \leq 1, \quad j = 1 \dots d, \quad d = 1$					
u	$u_j \sim \mathcal{N}(0, 1), \quad j = 1 \dots d_u, \quad d_u = 1$					
Risk measure						
$\bar{q}_{0.95}$	Superquantile Eq. (18)					

Table 4 Comparison of single- and multi-fidelity surrogate-assisted optimisation results

Test case	Budget	MF-GPR			GPR (TU0)	
		ϕ	Samples	Objective	Samples	Objective
TU1	2000	1	126(HF) 711(LF)	0.360	200 (HF)	0.425
TU2	2000	5	138(HF) 124(LF)	0.38290	200 (HF)	0.425
TU3	2000	9	60(HF) 150(LF)	0.38291	200 (HF)	0.425
TU1	4000	1	229(HF) 1686(LF)	0.363	400 (HF)	0.352
TU2	4000	5	184(HF) 432(LF)	0.364	400 (HF)	0.352
TU3	4000	9	220(HF) 190(LF)	0.38247	400 (HF)	0.352

The values in the table are averaged over 10 independent runs. The better mean objectives are highlighted in bold

The evolution of the objective value obtained with the classic GPR-based EGO algorithm and our MF-EGO approach is compared on Fig. 7. We can see that the higher number of HF samples of classical GPR provides a better starting point for the optimisation. However, the limited budget causes a premature stop of the optimisation algorithm. When MF-GPR is employed in the EGO algorithm, many low-fidelity samples can be generated to explore the design landscape and then HF samples are mostly generated to exploit the identified region of interest. The obtained objectives with different budgets confirm our findings made with the deterministic simulations of the previous subsection. In Table 4, we can see that when the budget is low (2000) then the GPR is not able to provide an accurate approximation. For this reason, the strategy employing MF-GPR outperforms the one with GPR. We

Table 5 List of test cases for uncertain multi-objective problems

	High-fidelity			Low-fidelity		
Test case	$f(x)$	ϕ_u	$cost_{HF}$	$e(x, \phi)$	ϕ	$cost_{LF}$
TMU0	Eq. (52) – (53)	–	20	–	–	–
TMU1	Eq. (52) – (53)	5	20	Eq. (37) + Eq. (41)	5	2.5
TMU2	Eq. (52) – (53)	5	20	Eq. (46)	5	2.5
TMU3	Eq. (52) – (53)	5	20	Eq. (37) + Eq. (41)	9	5
Inputs						
x	$0 \leq x_j \leq 1, \quad j = 1 \dots d, \quad d = 2$					
u	$u_j \sim \mathcal{N}(0, 1), \quad j = 1 \dots d_u, \quad d_u = 2$					
Risk measure						
$\bar{q}_{0.95}$	Superquantile Eq. 18					

Table 6 Comparison of single- and multi-fidelity surrogate-assisted multi-objective optimisation under uncertainty

Test case	Budget	MF-GPR			GPR (TMU0)	
		ϕ	Samples	IGD	Samples	IGD
TMU1	6000	5	231(HF) 516(LF)	0.172	300 (HF)	0.303
TMU2	6000	5	230(HF) 540(LF)	0.233	300 (HF)	0.303
TMU3	6000	9	215(HF) 409(LF)	0.209	300 (HF)	0.303
TMU1	8000	5	273(HF) 995(LF)	0.164	400 (HF)	0.203
TMU2	8000	5	293(HF) 826(LF)	0.182	400 (HF)	0.203
TMU3	8000	9	264(HF) 658(LF)	0.163	400 (HF)	0.203
TMU1	10,000	5	319(HF) 1421(LF)	0.175	500 (HF)	0.160

The IGD values of the obtained Pareto fronts are calculated with a reference set of 100 points. The values in the table are averaged over 10 independent runs. The better mean IGD values are highlighted in bold

can also notice that by increasing the fidelity of the low level function $\phi = 5$ and $\phi = 9$, the obtained objective becomes worse. The reason for this is that the budget is fixed and by increasing the fidelity of the low-level function also its cost increases, hence the number of LF samples becomes also sparse. Therefore the advantage of employing MF-GPR is diminished, however the total number of samples are still higher than in the GPR case resulting in a slightly better objective value.

When the budget is increased to a level that the GPR is able to provide accurate approximation, there is no advantage to use MF-GPR. In the case of fidelity

level $\phi = 1$ and $\phi = 5$ the obtained objective values are comparable with both GPR and MF-GPR. However, in case of $\phi = 9$ the low-fidelity level with sparse samples clearly provides a bad approximation which affects the quality on the top level as well and results in a significantly worse objective value.

7.4 Multi-objective optimisation under uncertainty with multi-fidelity surrogate

The results of the multi-objective problem are in agreement with the results obtained for the single objective case. The parameter settings of the investigated test functions are summarised in Table 5. As Table 6 shows that the advantage of employing a MF-GPR is significant when the approximation on the low-fidelity is sufficiently good. We can observe when the low-fidelity level is prone to instability error the obtained IGD values are worse compared to when the low-fidelity has only residual and stochastic error. Similarly to the single-objective tests increasing the accuracy of the low-fidelity level might decrease to overall performance of the algorithm. At any fidelity level the accuracy can be increased typically by increasing the computational cost too which can result in a sparse sample on the low-fidelity level. As expected the IGD values decrease with increasing budget. Similarly to the single objective case, when the number of samples are high enough the GPR model approximation becomes satisfactorily good. From this point employing MF-GPR is unnecessary.

8 Conclusions

In this paper we presented a complete optimisation strategy for multi-objective design optimisation under uncertainty when the computational budget is highly limited. Our optimisation strategy employs multi-fidelity surrogate models to tackle the problem of expensive design evaluations. Furthermore, to obtain statistically significant samples for the risk measures, our strategy employs a surrogate-assisted uncertainty quantification technique.

For multi-fidelity surrogate-assisted optimisation we proposed an acquisition function concept which is independent of the number of fidelity levels and the number of samples of the fidelity levels. Moreover, the acquisition function is derived for both single-objective and multi-objective optimisation.

To evaluate the performance of our optimisation strategy, we extended the existing multi-fidelity and multi-objective test suite for testing uncertain problems.

Based on our obtained results we draw the conclusion that multi-fidelity surrogates are advantageous when the experimental budget is highly limited and standard surrogate techniques cannot provide accurate predictions. In this case if cheap low-fidelity simulations are available, multi-fidelity surrogates are able to make better predictions. The advantage of employing a multi-fidelity surrogate is diminished or even becomes a disadvantage when the number of high-fidelity observations are enough for standard surrogate techniques.

Beside using the multi-fidelity surrogate to combine various levels of fidelity of numerical simulations, the technique can be employed to combine physical

measurements with numerical experiments. In industrial environment it is also common to have an already available surrogate model of the problem to be optimised. In this case the multi-fidelity surrogate technique can be used to update the already existing surrogate with new observations at a fraction of the cost of retraining the entire model.

Acknowledgements This work was partially supported by the H2020-MSCA-ITN-2016 UTOPIAE, Grant Agreement 722734.

References

- Alexandrov N et al (2000) Optimization with variable-fidelity models applied to wing design. In: 38th Aerospace sciences meeting and exhibit
- Bandler JW (1969) Optimization methods for computer-aided design. *IEEE Trans Microw Theory Tech* 17(8):533–552
- Beyer H-G, Sendhoff B (2007) Robust optimization-a comprehensive survey. *Comput Methods Appl Mech Eng* 196(33–34):3190–3218
- Choi S-K et al (2004) Polynomial chaos expansion with latin hypercube sampling for estimating response variability. *AIAA J* 42(6):1191–1198
- Deb K et al (2005) Scalable test problems for evolutionary multiobjective optimization. *Evolutionary multiobjective optimization*. Springer, Berlin
- Emmerich MTM, Giannakoglou KC, Naujoks B (2006) Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Trans Evolut Comput* 10(4):421–439
- Fernández-Godino GM et al (2019) Issues in deciding whether to use multifidelity surrogates. *AIAA J* 57(5):2039–2054
- Forrester AJ, Sóbester A, Keane AJ (2007) Multi-fidelity optimization via surrogate modelling. In: *Proceedings of the royal society a: mathematical, physical and engineering sciences*
- Forrester A, Sóbester A, Keane A (2008) *Engineering design via surrogate modelling: a practical guide*. Wiley, New York
- Giering R, Kaminski T (1998) Recipes for adjoint code construction. *ACM Trans Math Softw (TOMS)* 24(4):437–474
- Habib A, Singh HK, Ray T (2019) A multiple surrogate assisted multi/many-objective multi-fidelity evolutionary algorithm. *Inf Sci* 502:537–557
- Han ZH, Görtz S, Zimmermann R (2013) Improving variable-fidelity surrogate modeling via gradient-enhanced Kriging and a generalized hybrid bridge function. *Aerosp Sci Technol* 25:177–189
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13:455–492
- Keane AJ (2006) Statistical improvement criteria for use in multiobjective design optimization. *AIAA J* 44(4):879–891
- Kennedy MC, O’Hagan A (2000) Predicting the output from a complex computer code when fast approximations are available. *Biometrika* 87(1):1–13
- Knowles J (2006) ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multi-objective optimization problems. *IEEE Trans Evolut Comput* 10(1):50–66
- Kuehl RO (2000) *Design of experiments: statistical principles of research design and analysis*, 2nd edn. Duxbury Press, Belmont, CA
- Lam R et al. (2018) *Advances in Bayesian optimization with applications in aerospace engineering*. In: 2018 AIAA non-deterministic approaches conference
- Le Gratiet L (2013) *Multi-fidelity Gaussian process regression for computer experiments*. Diss. Université Paris-Diderot-Paris VII
- Lemieux C (2009) *Monte carlo and quasi-monte carlo sampling*. Springer, Berlin
- Meliani M et al (2019) Multi-fidelity efficient global optimization: methodology and application to airfoil shape design. In: *AIAA aviation 2019 forum*

- Poloczek M, Jialei W, Peter F (2017) Multi-information source optimization. In: *Advances in neural information processing systems*
- Press WH et al (2007) *Numerical recipes: the art of scientific computing*, 3rd edn. Cambridge University Press, Cambridge
- Quagliarella D, Petrone G, Iaccarino G (2014) Optimization under uncertainty using the generalized inverse distribution function. In: Fitzgibbon W, Kuznetsov YA, Neittaanmäki P, Pironneau O (eds) *Modeling, simulation and optimization for science and technology*. Springer, Dordrecht, pp 171–190
- Rasmussen CE (2004) Gaussian processes in machine learning. In: Bousquet O, von Luxburg U, Rätsch G (eds) *Advanced lectures on machine learning. ML 2003. Lecture Notes in Computer Science*, vol 3176. Springer, Berlin, Heidelberg
- Rockafellar T, Royset J (2015) Engineering decisions under risk averseness. *ASCE-ASME J Risk Uncertain Eng Syst A Civ Eng* 1(2):04015003
- Rockafellar T, Uryasev S (2002) Conditional value-at-risk for general loss distributions. *J Bank Finance* 26:1443–1471
- Schobi R, Sudret B, Wiart J (2015) Polynomial-chaos-based kriging. *Int J Uncertain Quantification* 5(2):171–193
- Skinner SN, Zare-Behtash H (2018) State-of-the-art in aerodynamic shape optimisation methods. *Appl Soft Comput* 62:933–962
- Tinkle C et al (2019) A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. *Soft Comput* 23(9):3137–3166
- Wang H, Jin Y, Doherty J (2017) A generic test suite for evolutionary multifidelity optimization. *IEEE Trans Evolut Comput* 22(6):836–50
- Yang X-S (2014) *Nature-inspired optimization algorithms*. Elsevier, Amsterdam

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.