

Inverse surrogate modeling: output performance space sampling

Ivo Couckuyt* and Dirk Gorissen*
Ghent University-IBBT, Ghent, 9050, Belgium

Tom Dhaene† and Filip De Turck†
Ghent University-IBBT Ghent, 9050, Belgium

The use of Surrogate Based Optimization (SBO) is widely spread in engineering design to find optimal performance characteristics of computational expensive simulation codes (i.e., the forward problem). However, often the practitioner knows a priori the desired performance and is interested in finding the associated design parameters (i.e., the inverse problem). A popular method to solve such reverse (inverse) problems is to minimize the error between the simulated performance and the desired goal performance. While this works quite well, it only allows for the identification of one design, namely, the global optimum according to the error function. In this paper, the authors propose a method to sample a whole range of performances using surrogate models. This technique, based on the probability of improvement criterion and kriging models, is successfully demonstrated on the branin function to obtain a quasi-uniform set of samples that lie in a predefined output range.

I. Introduction

This paper is concerned with efficiently solving complex, computational expensive problems using surrogate modeling techniques.¹ Surrogate models, also known as metamodels, are cheap approximation models for computational expensive (black-box) simulations. Surrogate modeling techniques are well-suited to handle, for example, expensive finite element (FE) simulations, computational fluid dynamic (CFD) simulations and, of course, physical experiments.

Depending on the construction and usage of surrogate models several modeling flavors can be distinguished. Surrogate models can be built upfront to approximate the simulation code accurately over the entire input space and, hence, can afterwards be used to replace the expensive code for design, analysis and optimization purposes. On the other hand, the construction of surrogate models can also be integrated in the optimization process. Usually, the latter case, known as Surrogate Based Optimization (SBO), generates surrogate models on the fly that are only accurate in certain regions of the input space, e.g., around optimal regions.

The construction of surrogate models as efficiently as possible is an entire research domain in itself. In order to come to an acceptable model, numerous problems and design choices need to be overcome (*what data collection strategy to use, which variables are relevant, how to integrate domain knowledge, etc.*). Other aspects of surrogate modeling include choosing the right type of approximation model for the problem at hand, a tuning strategy for the surrogate model parameters (=hyperparameters), and a performance measure to assess the accuracy of the surrogate model.²

The general work-flow of surrogate modeling is illustrated in Figure 1. First, an experimental design, e.g., from Design of Experiments (DOE), is specified and evaluated. Subsequently, surrogate models are built to fit this data as good as possible, according to a set of measures (e.g., cross validation). The hyperparameters are estimated using an optimization algorithm. The accuracy of the set of surrogate models is improved until no further improvement can be obtained (or when another stopping criterion, such as a time limit, is met). If the stopping criteria are satisfied the process is halted and the final, best surrogate model is returned. On the other hand, when no stopping criterion is met, a sequential design strategy, also known as active learning or adaptive sampling, will select new samples to be evaluated and the surrogate models are updated with this new data.

Most often, surrogate models are used to solve so-called forward problems. The practitioner is interested in the output or performance characteristics of the simulation system given the input parameters.

*PhD student, Department of Information Technology (INTEC), Ghent University-IBBT, Gaston Crommenlaan 8, Ghent, 9050, Belgium, AIAA student member.

†Professor, Department of Information Technology (INTEC), Ghent University-IBBT, Gaston Crommenlaan 8, Ghent, 9050, Belgium.

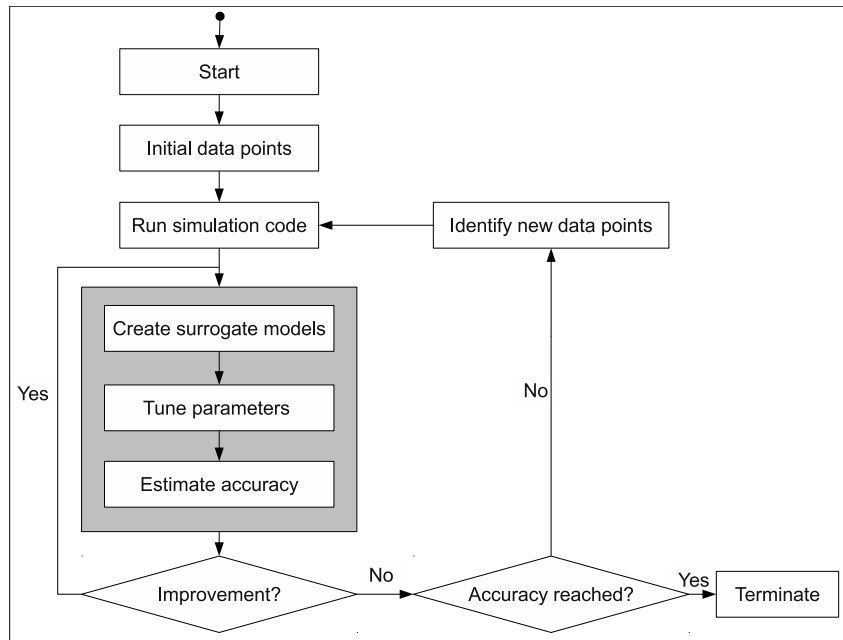


Figure 1: Flow chart of the surrogate modeling process.

The surrogate models define the mapping between the input space (design space) and the output space (performance space).

In contrast, the focus of the reverse (inverse) problem is on exploring the input space. Ideally, a surrogate model could be created that maps the output parameters to the input parameters (as opposite to forward modeling) of the complex system over the entire input space. However, many inverse problems are typically ill-posed. Considering Hadamard's definition of ill-posedness,³ the two outstanding problems hampering the creation of a full inverse surrogate model are non-uniqueness and instability. A good overview of the associated intricacies is presented by Barton in.⁴ For all the above reasons, the inverse problem is often reduced to the task of finding one (or more) input parameter combination for a certain output characteristic. Still, it is possible that,

1. no such input parameter combination exists
2. more than one input parameter combination satisfies the given output characteristic.

A typical inverse problem is the estimation of some (physical) material or design parameter, e.g., the permittivity of a substrate⁵ or the elasticity of rubber,⁶ given the desired output or system behavior. A popular solution is to convert the reverse problem to a (forward) optimization problem. Namely, a simulation model is constructed, parametrized by the properties or design parameters of interest. By minimizing the error between the parametrized simulation model and the measured data the input parameters (material properties) of the simulation model are obtained that correspond with the measurements or desired output, see Figure 2.

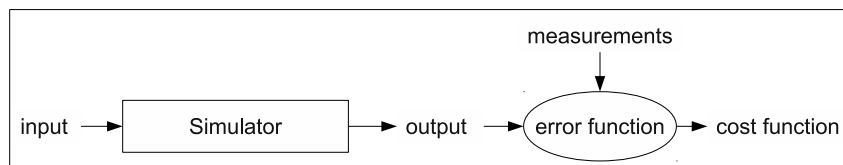


Figure 2: The inverse problem is often solved by minimizing the error function between the simulation output and the measured data.

The focus of this paper is to efficiently solve inverse problems directly, i.e., identifying the regions in the input space that correspond to the desired output value(s). The contribution of this paper is a novel method, i.e., a sequential design step in surrogate modeling, that is able to efficiently sample the (disconnected) input regions, in a quasi-uniform way. The surrogate model of choice is the Gaussian Process (GP) based kriging. Kriging is a popular surrogate model for the approximation of deterministic

computer code.⁷ GP enable the use of statistical infill criteria in the sequential design strategy. The presented method consists of the extension of such a statistical infill criterion, namely, the Probability of Improvement (PoI),⁸ and a new search strategy to exploit the infill criterion.

This approach has been implemented in a flexible research platform for surrogate modeling, the **SU**rrogate **MO**deling (SUMO) Toolbox^{a,2} and is illustrated on the branin function.

Section II describes the use of infill criteria and in particular the generalization of the PoI criterion. A new search strategy to exploit infill criteria is described in section III. This approach is demonstrated on the branin function. Details of the setup are found in section IV. Results and conclusion form the last two sections of this paper, i.e., sections V and VI.

II. Infill criteria

In engineering, infill criteria are (sampling) functions, also known as figures of merit or metrics, that measure how *interesting* a data point is in the input space. Starting from an initial approximation of the simulation system, new sample points (infill or update points) are selected based on an infill criterion. The variety of infill criteria ranges from the accuracy of the prediction (e.g., for creating globally accurate surrogate models) to the prediction itself (e.g., to facilitate optimization). In global SBO it is crucial that the infill criterion is a balance between exploration^b and exploitation^c.

A well-known infill criterion that is able to effectively solve this exploitation/exploration trade-off is Expected Improvement (EI), which has been popularized by Jones et al.^{9,10} as the Efficient Global Optimization (EGO) algorithm. Jones wrote an excellent discussion regarding the infill criteria approach in.⁸ Subsequently, Sasena compared different infill criteria for optimization and investigated extensions of those infill criteria for constrained optimization problems in.¹¹

A. Probability of Improvement (PoI)

Among several statistical infill criteria investigated by Jones the PoI is the focus of this work. The PoI equation (1), defined below, can be interpreted graphically (see Figure 3). At $\mathbf{x} = 0.5$, a Gaussian probability density function is drawn and expresses the uncertainty about the predicted function value of a sampled and unknown function $y = f(\mathbf{x})$. Thus, the uncertainty at any point \mathbf{x} is treated as the realization of a random variable $Y(\mathbf{x})$ with mean $\hat{y} = \hat{f}(\mathbf{x})$ (= prediction) and variance $\hat{s}^2 = \hat{\sigma}^2(\mathbf{x})$ (= prediction variance). Assuming the random variable $Y(\mathbf{x})$ is normally distributed, then the shaded area under the Gaussian probability density function is the *PoI* of any newly calculated function value $f(\mathbf{x})$ over the intermediate minimum function value f_{min} (the dotted line). PoI is denoted as $P(Y(\mathbf{x}) \leq f_{min})$, i.e.,

$$\begin{aligned} PoI = P(Y(\mathbf{x}) \leq f_{min}) &= \int_{-\infty}^{f_{min}} Y(\mathbf{x}) dY \\ &= \Phi\left(\frac{f_{min} - \hat{y}}{\hat{s}}\right), \end{aligned} \quad (1)$$

where $\Phi(t)$ is the standard normal cumulative distribution function $\Phi(t) = \frac{1}{2} \left[1 + erf\left(\frac{t}{\sqrt{2}}\right) \right]$ and $erf(\cdot)$ is the error function.

PoI or any other statistical infill criteria (e.g., EI) are optimized over \mathbf{x} to find the subsequent data point to evaluate. Note, however, that besides the prediction $\hat{y} = \hat{f}(\mathbf{x})$ of the surrogate model, a point-wise variance estimation $\hat{s}^2 = \hat{\sigma}^2(\mathbf{x})$ of the surrogate is also required. Both predictions (\hat{y} and \hat{s}^2) are provided by the kriging surrogate model.

B. Generalized Probability of Improvement (gPoI)

While PoI is a very useful infill criterion for optimization, it only focuses on the global optimum, not on a range of output values. The authors extend the idea of the PoI criterion to allow identification of an arbitrarily band in the output space. Let $[T_1, T_2]$ be the range of interest in the output space. Then the probability that the function value $f(\mathbf{x})$ of a point \mathbf{x} lies in this range is defined as:

^aThe SUMO Toolbox can be downloaded from: <http://sumo.intec.ugent.be>. An AGPL open source license is available for research purposes.

^bexploration = enhancing the overall accuracy of the surrogate model

^cexploitation = enhancing the accuracy of the surrogate model solely in the region of the (current) optimum

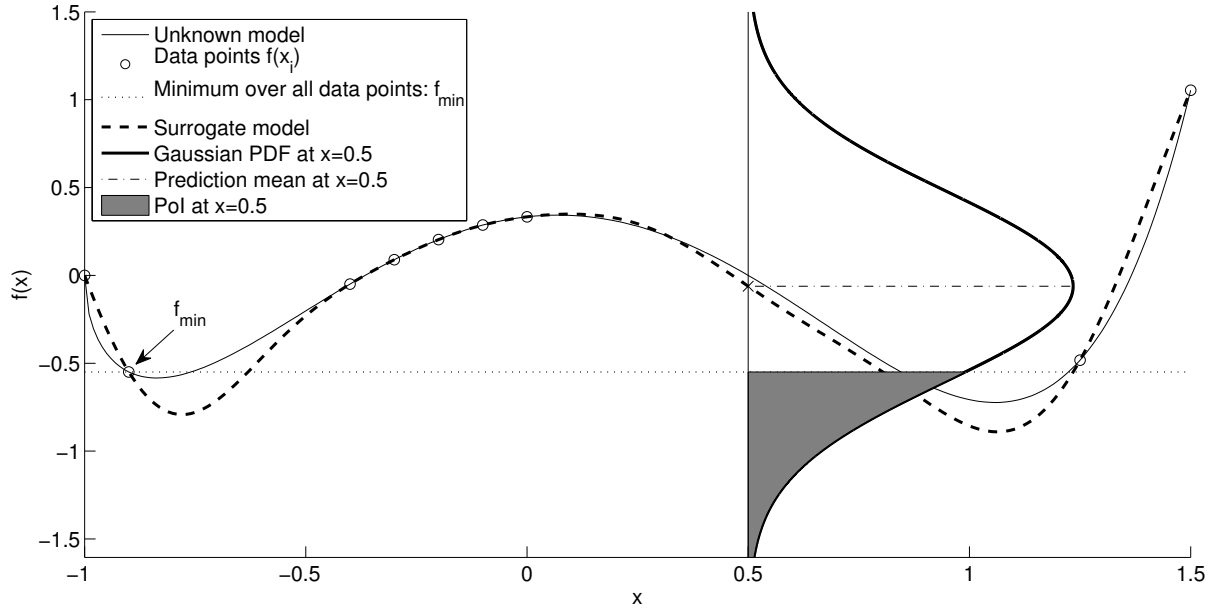


Figure 3: Graphical illustration of a Gaussian Process and Probability of Improvement (PoI). A surrogate model (dashed line) is constructed based on some data points (circles). For each point the surrogate model predicts a Gaussian probability density function (PDF). E.g., at $x = 0.5$ an example of such a PDF is drawn. The volume of the shaded area is the PoI over the minimum function value f_{\min} .

$$\begin{aligned}
 gPoI(\mathbf{x}) &= P(T_1 \leq Y(\mathbf{x}) \leq T_2) = \int_{T_1}^{T_2} Y(\mathbf{x}) dY \\
 &= P(T_2 \leq Y(\mathbf{x})) - P(T_1 \leq Y(\mathbf{x})) \\
 &= \int_{-\infty}^{T_2} Y(\mathbf{x}) dY - \int_{-\infty}^{T_1} Y(\mathbf{x}) dY \\
 &= \Phi\left(\frac{T_2 - \hat{y}}{\hat{s}}\right) - \Phi\left(\frac{T_1 - \hat{y}}{\hat{s}}\right), \tag{2}
 \end{aligned}$$

where $\Phi(t)$ is the standard normal cumulative distribution function $\Phi(t) = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{t}{\sqrt{2}}\right) \right]$ and $\operatorname{erf}(\cdot)$ is the error function.

This criterion represents the probability that a new point \mathbf{x} is in the desired output interval $[T_1, T_2]$. Hence, the abbreviation ‘‘PoI’’ is not completely correct anymore as the focus is now on sampling an interval instead of improving the optimum. Unlike PoI, the gPoI cannot be simply optimized to identify new samples. Existing samples that are already lying in the output range $[T_1, T_2]$ have probability one (see Figure 6a), and, hence, straightforward optimization of the criterion will result in duplicate samples. Other (space-filling) strategies have to be devised that makes fully use of the information provided by gPoI. In fact, note that this is actually a constrained sampling problem. If uncertainty was not taken into account a simple constraint would suffice, i.e., $c(\mathbf{x}) = T_1 \leq f(\mathbf{x}) \leq T_2$. By taking uncertainty into account, (new) regions of interest may be identified faster and more accurately.

III. Search strategies

The techniques explained in the previous sections are all utility functions. Such utility functions are used to identify interesting new points in a sequential design strategy. Various search strategies exist to exploit the utility functions.

For instance, the original EGO algorithm⁹ simply optimizes the EI. In particular, the deterministic branch and bound methodology was used to find the global optimum. To that end, a convex upper bound had to be calculated for the EI. However, if one wants to use other utility functions (or other types of

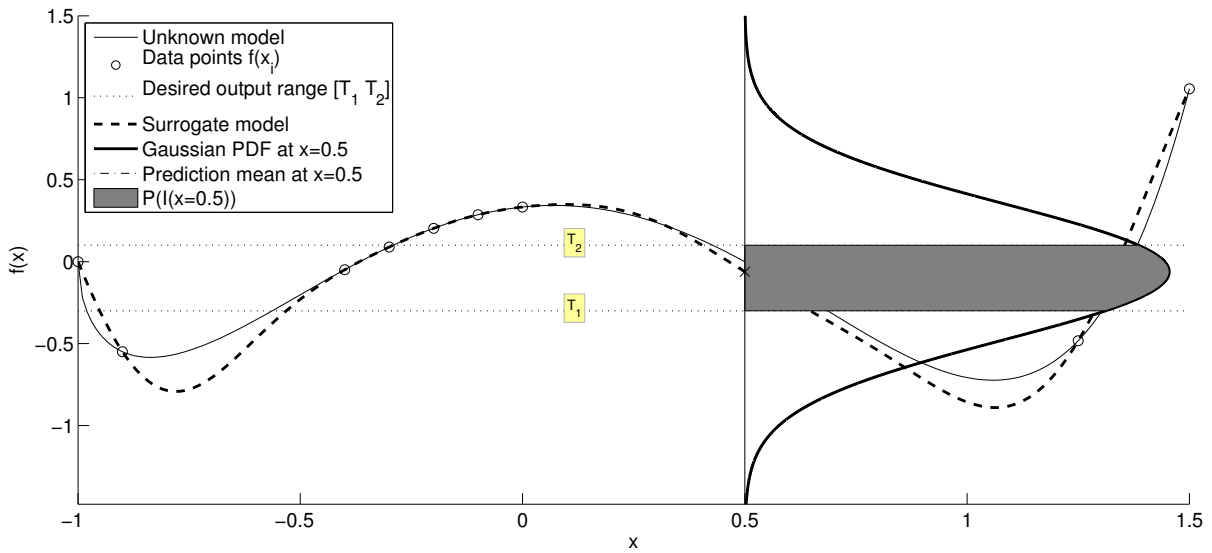


Figure 4: Graphical illustration of a Gaussian Process and the generalized Probability of Improvement (gPoI). A surrogate model (dashed line) is constructed based on some data points (circles). For each point the surrogate model predicts a Gaussian probability density function (PDF). E.g., at $x = 0.5$ an example of such a PDF is drawn. The volume of the shaded area is the gPoI based on the desired output range $[T_1, T_2]$.

surrogate models) this upper bound must be redefined. In later work more black-box optimization methods were used with similar results, e.g., the DIviding REctangles (DIRECT)¹² or an extensive pattern search. Moreover, multi-modal optimization methods are suggested in literature (e.g., in^{13,14}) to select multiple samples in one iteration, taking full advantage of parallel computing.

Global optimization methods are not suited to directly exploit the new gPoI criterion because there might exist multiple (or even an infinite number of) solutions. Therefore the authors adapt a generic sampling framework for sequential design. The algorithmic flow is depicted in Figure 5. The search strategy is configured as followed: First, n candidate samples are drawn from the uniform distribution. Subsequently, these candidates are ranked according to two ($k = 2$) criteria: the gPoI criterion (see Figure 6a) and a maximin distance (MD) criterion that calculates the Euclidean distance to the closest existing sample (scaled into $[0, 1]$ using an estimate on the upper bound). The latter criterion takes care of the space-filling properties in the input space, in effect introducing negative probability around existing samples (see Figure 6b). The combined sequential sampling criterion is now defined as the weighted average of the two criteria, see Figure 6c. To maximize space-fillingness only one ($m = 1$) sample is chosen during each iteration, namely, the candidate sample with the highest combined score.

This approach is a nice balance between exploration (space-filling) and exploitation (uniformly sampling in the input range). If the gPoI criterion is low across the whole input space the MD criterion will dominate and, hence, the input space will be further explored, enhancing the accuracy of the surrogate model. On the other hand, as the number of samples increases the MD score will decrease, enabling the exploitation of the gPoI criterion.

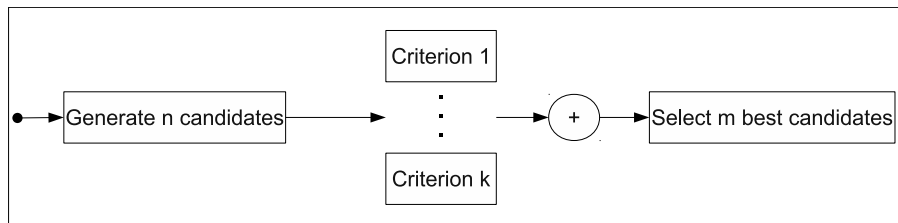
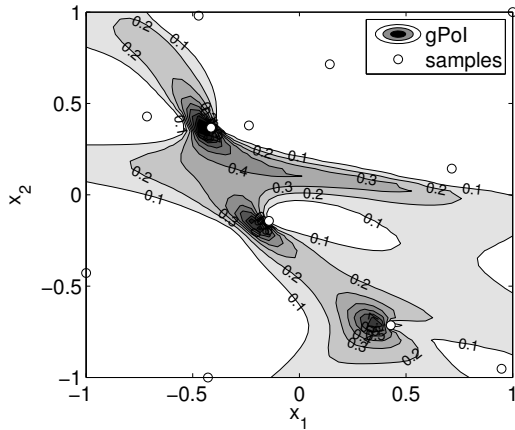
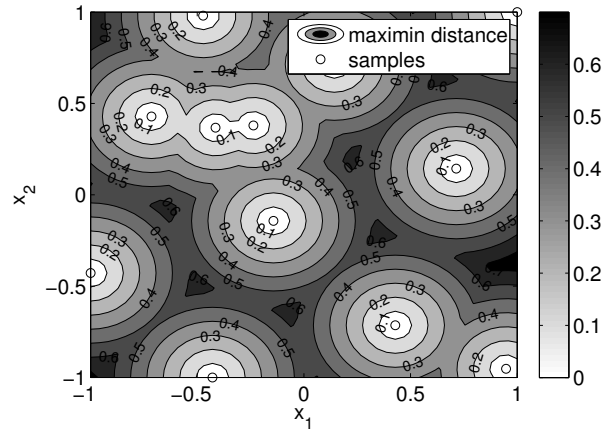


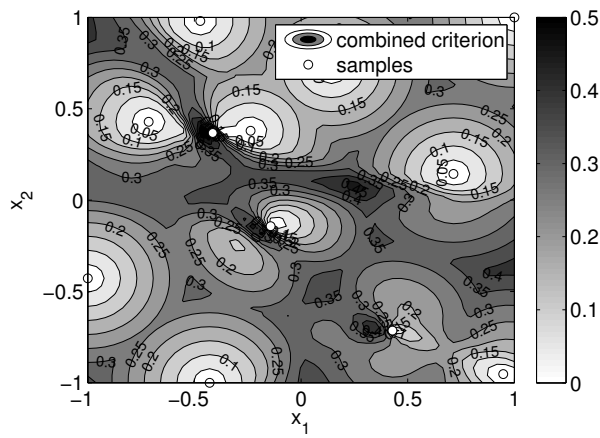
Figure 5: General flow of a sequential design strategy.



(a) Contour plot of the gPoI criterion.



(b) Contour plot of the MD criterion.



(c) Contour plot of the combined criteria.

Figure 6: Two sequential sampling criteria (gPoI and MD) and the combined weighted average. (for the 2D branin function (3))

IV. Experimental setup

Version 7.0 of the SUMO toolbox² is utilized to adaptively update a kriging model by sampling in the output space. The method is illustrated on the branin function, acting as the expensive simulation code, and is defined by,

$$f(x_1, x_2) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10. \quad (3)$$

Note, that the branin function is a typical benchmark function for optimization, while here it is used for constraint sampling in the output space. Specifically, points are sampled in the performance range $[T_1, T_2] = [20, 30]$.

The configuration of the toolbox is as follows. An initial set of eight samples is generated by an optimal maximin Latin Hypercube Design (LHD;¹⁵). Subsequently, infill points are selected using the search strategy explained in section III. The process is halted when the number of samples exceeds 100. The surrogate model of choice is an efficient custom implementation of kriging. The hyperparameters are determined using Maximum Likelihood Estimation (MLE). The actual optimization of the likelihood function is accomplished by a Sequential Quadratic Programming method (SQPLab^{d16}), taking into account derivative information. The optimization of the hyperparameters θ occur in log space with lower bounds $\theta_{lower} = (-2, -2)$, upper bounds $\theta_{upper} = (2, 2)$, starting point $\theta_0 = (0.5, 0.5)$, and the standard Gaussian correlation function.

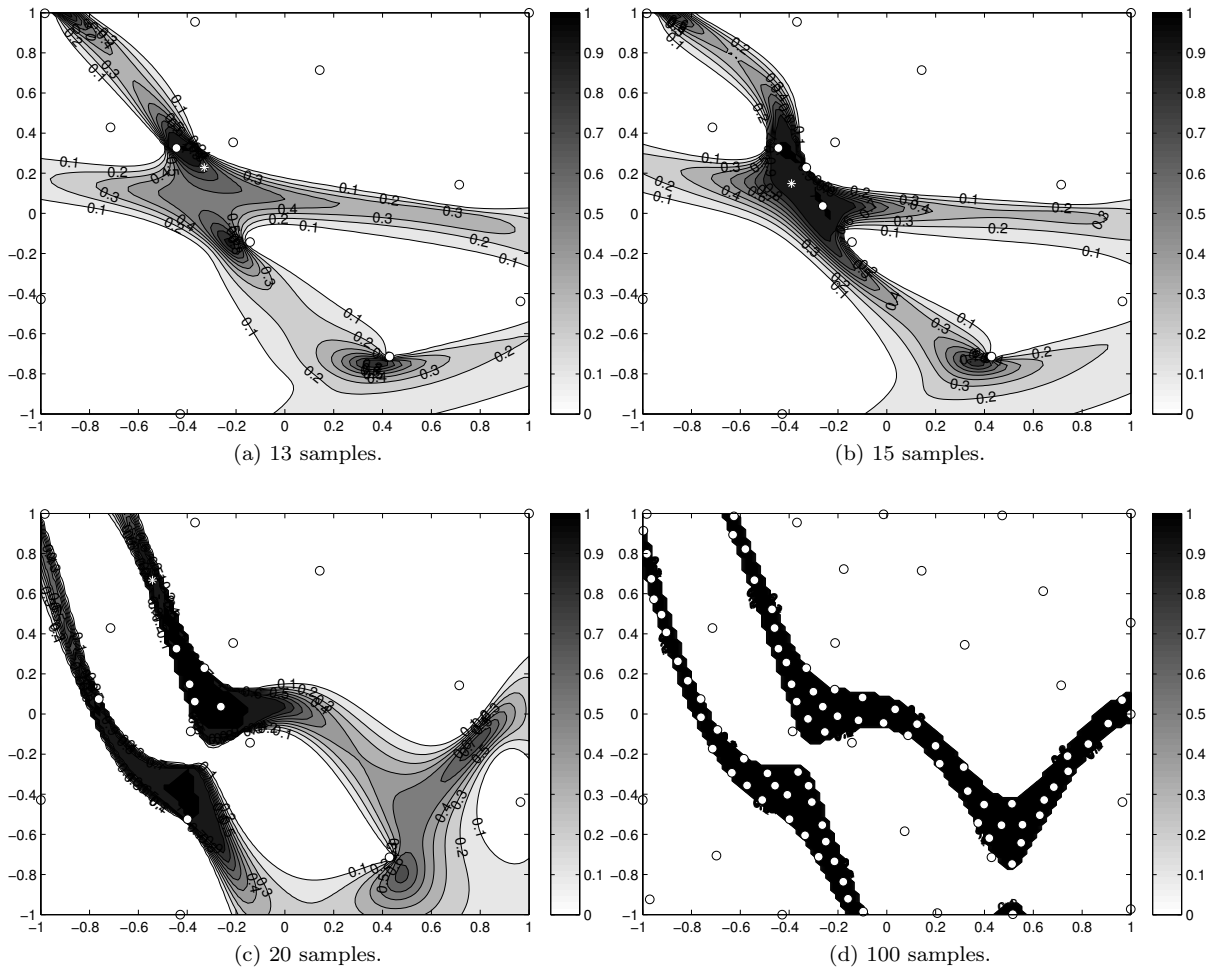
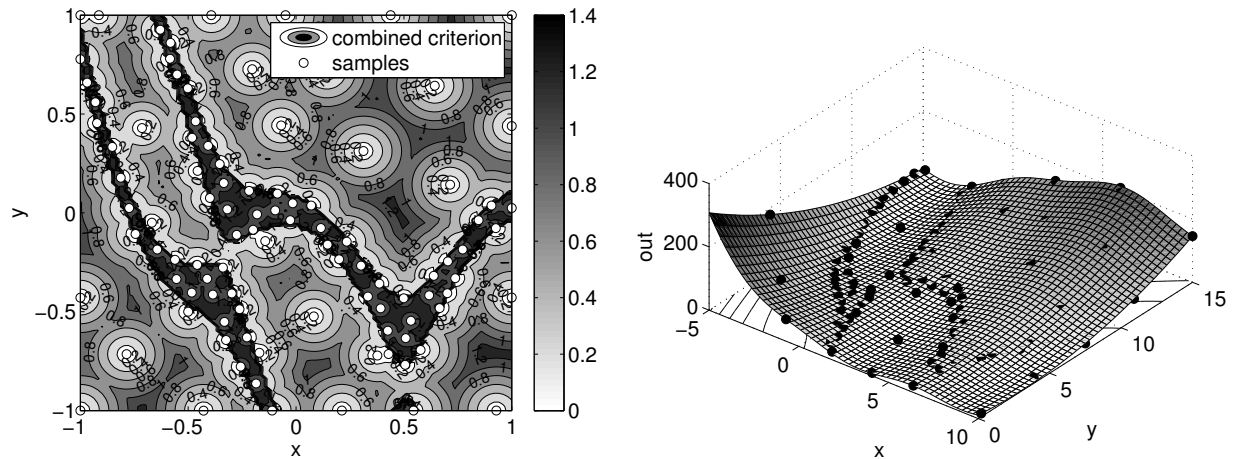


Figure 7: Exploring and exploiting the $gPoI$ criterion. The existing samples are denoted by the circles and the subsequent sample to evaluate is drawn as a star.

^dSQPLab is found at <http://www-rocq.inria.fr/~gilbert/modulopt/optimization-routines/sqplab/sqplab.html>

V. Results

Figure 7 shows snapshots of the gPoI criterion during the sampling process. At the start (see Figure 7a) the kriging model is not yet accurate, hence, the most promising regions are located around or close to existing samples. Subsequently, these regions are explored at the edges (see Figure 7c and 7b). As the kriging model becomes more accurate the input regions corresponding to the output range of interest are located nicely by the gPoI criterion, see Figure 7c. After this exploration phase, the located regions are exploited by uniformly sampling (space-filling) until 100 samples have been evaluated. In the last plot (Figure 7d) it is seen that the the band of interest is densely sampled in a quasi-uniform way.



(a) Contour plot of the combined sequential sampling criterion (gPoI and MD) (snapshot at 101 samples). (b) Final kriging model of the branin function (101 samples).

Figure 8: Final kriging model of the branin function, accurate at the desired output range.

Exploration at the edges of the high probability areas of the gPoI criterion is due to the MD criterion. Figure 8a depicts a contour plot of the combined sequential sampling criterion, i.e., the average of the gPoI criterion and the maximin distance criterion, that determines the sample that is selected. In addition, a landscape plot of the final surrogate model (most accurate in the desired performance range) is seen in Figure 8b.

VI. Conclusion and future work

This paper introduced a simple but powerful method for constrained output (=performance) sampling problems, i.e., given a certain performance range identify the corresponding parameters in the design space. This approach, based on a generalized PoI criterion, is implemented in a Matlab surrogate modeling toolbox and tested on the branin function with interesting results.

For this two dimensional problem the regions of interest are densely sampled. While the proposed search strategy is relatively dimension-free, more work is needed to investigate the impact of the number of input variables (the dimension) on the number of samples needed to sufficiently cover the input regions that correspond to the desired output performances.

Acknowledgments

Ivo Couckuyt is funded by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

- ¹Wang, G. and Shan, S., "Review of Metamodeling Techniques in Support of Engineering Design Optimization," *Journal of Mechanical Design*, Vol. 129, No. 4, 2007, pp. 370–380.
- ²Gorissen, D., Crombecq, K., Couckuyt, I., Demeester, P., and Dhaene, T., "A Surrogate Modeling and Adaptive Sampling Toolbox for Computer Based Design," *Journal of Machine Learning Research*, Vol. 11, 2010, pp. 2051–2055.
- ³Hadamard, J., "Sur les problèmes aux dérivées partielles et leur signification physique," *Tech. Rep.* 49–52, Princeton University Bulletin, 1902.
- ⁴Barton, R., "Issues in development of simultaneous forward-inverse metamodels," *Proceedings of the Winter Simu-*

lation Conference, 2005, pp. 209–217.

⁵Couckuyt, I., Declercq, F., Dhaene, T., and Rogier, H., “Surrogate-Based Infill Optimization Applied to Electromagnetic Problems,” *Advances in design optimization of microwave/rf circuits and systems (special issue)*, Vol. 20, No. 5, 2010, pp. 492.

⁶Aernouts, J., Couckuyt, I., Crombecq, K., and Dirckx, J., “Elastic characterization of membranes with a complex shape using point indentation measurements and inverse modelling,” *International Journal of Engineering Science*, Vol. 48, 2010, pp. 599–611.

⁷Sacks, J., Welch, W. J., Mitchell, T., and Wynn, H. P., “Design and analysis of computer experiments,” *Statistical science*, Vol. 4, No. 4, 1989, pp. 409–435.

⁸Jones, D. R., “A Taxonomy of Global Optimization Methods Based on Response Surfaces,” *Global Optimization*, Vol. 21, 2001, pp. 345–383.

⁹Jones, D. R., Schonlau, M., and Welch, W. J., “Efficient Global Optimization of Expensive Black-Box Functions,” *Journal of Global Optimization*, Vol. 13, No. 4, Nov. 1998, pp. 455–492.

¹⁰Schonlau, M., *Computer Experiments and Global Optimization*, Ph.D. thesis, University of Waterloo, 1997.

¹¹Sasena, M., *Flexibility and Efficiency Enhancements For Constrained Global Design Optimization with Kriging Approximations*, Ph.D. thesis, University of Michigan, 2002.

¹²Jones, D., Perttunen, C., and Stuckman, B., “Lipschitzian optimization without the Lipschitz constant,” *Optimization Theory and Applications*, Vol. 79, No. 1, 1993, pp. 157–181.

¹³Ponweiser, W., Wagner, T., and Vincze, M., “Clustered Multiple Generalized Expected Improvement: A Novel Infill Sampling Criterion for Surrogate Models,” *Congress on Evolutionary Computation*, 2008.

¹⁴Sobester, A., Leary, S. J., and Keane, A. J., “A parallel updating scheme for approximating and optimizing high fidelity computer simulations,” *Structural and Multidisciplinary Optimization*, Vol. 27, 2004, pp. 371–383.

¹⁵Dam, E., van Husslage, B., den Hertog, D., and Melissen, J., “Maximin Latin hypercube designs in two dimensions,” *Operations Research*, Vol. 55, No. 1, 2007, pp. 158–169.

¹⁶Bonnans, J., Gilbert, J., Lemaréchal, C., and Sagastizábal, C., *Numerical Optimization: Theoretical and Practical Aspects*, Springer, 2006.