# A SURVEY OF MODEL-BASED METHODS FOR GLOBAL OPTIMIZATION

Thomas Bartz-Beielstein
*SPOTSeven Lab, TH Köln, Gummersbach, Germany.*
spotseven.de
thomas.bartz-beielstein@th-koeln.de

**Abstract** This article describes model-based methods for global optimization. After introducing the global optimization framework, modeling approaches for stochastic algorithms are presented. We differentiate between models that use a distribution and models that use an explicit surrogate model. Fundamental aspects of and recent advances in surrogate-model based optimization are discussed. Strategies for selecting and evaluating surrogates are presented. The article concludes with a description of key features of two state-of-the-art surrogate model based algorithms, namely the evolvability learning of surrogates (EvoLS) algorithm and the sequential parameter optimization (SPO).

**Keywords:** Global optimization, Surrogate model.

## 1. Introduction

*Model-based optimization* (MBO) plays a prominent role in todays modeling, simulation, and optimization processes. It can be considered as the most efficient technique for expensive and time-demanding real-world optimization problems. Especially in the engineering domain, MBO is an important practice. Recent advances in computer science, statistics, and engineering in combination with progress in high-performance computing provide tools for handling problems, which were considered unsolvable only a few decades ago. This article presents a survey of MBO for global optimization.

*Global optimization* (GO) can be categorized based on different criteria. For example, the properties of problems to be solved (continuous versus combinatorial, linear versus nonlinear, convex versus multimodal, etc.) can be used. This article presents an algorithmic view on global optimization, i.e., properties of algorithms that search for new solutions are considered.

The term GO will be used in this article for algorithms that are trying to find and explore global optimal solutions with complex, multimodal objective functions [50]. Global optimization problems are difficult to solve, because nearly no structural information (e.g., number of local extrema) is available. Global optimization problems belong to the class of *black-box functions*, i.e., functions for which the analytic form is unknown. Note, the class of black-box functions contains also functions that are easy to solve, e.g., convex functions, which are not discussed in the following. This article focuses on difficult black-box functions.

Consider the *optimization problem* given by

$$\text{Minimize: } f(\mathbf{x}) \quad \text{subject to } \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is referred to as the *objective function* and $\mathbf{x}_l$ and $\mathbf{x}_u$ denote the lower and upper bounds of the search space (region of interest), respectively. This setting arises in many real-world systems when the explicit form of the objective function $f$ is not readily available, e.g., if the user has no access to the source code of a simulator.

This survey covers *stochastic (random) search algorithms*, deterministic GO algorithms are not further discussed. Random and stochastic search will be used synonymously in the remainder of this article.

An iterative search algorithm that uses a stochastic procedure to generate the next iterate is referred to as a *stochastic search algorithm*. The next iterate can be a candidate solution to the GO or a probabilistic model, where solutions can be drawn from. Stochastic search algorithms are considered robust and easy to implement, because they do not depend on any structural information of the objective function such as gradient information or convexity. This feature is one of the main reasons for the popularity of stochastic search in the domain of GO. Stochastic search algorithms can further be categorized as *instance-based* or *model-based* algorithms [71]. Furthermore, there are basically two model-based approaches: (a) *distribution-based* models and (b) *surrogate* models. We consider four important representatives of surrogate model based optimization: (i) *Multi-fidelity metamodeling* uses several models of the same real system and plays an important role in CFD/FEM based simulation and optimization. (ii) *Evolutionary surrogate based optimization* extends the traditional EA framework, and (iii) *Ensemble surrogate based optimization* combines two or more different surrogate models.

So far, we have obtained the GO categorization (or taxonomy) based on algorithms as shown in Fig. 1.

The remainder of this article is structured as follows. After introducing instance-based stochastic search algorithms (category [2.1]), Section 2 describes modeling approaches for stochastic algorithms, i.e., it

```
|
|__ [1] Deterministic
|__ [2] Random Search
     |__ [2.1] Instance based
     |__ [2.2] Model based optimization (MBO)
          |__ [2.2.1] Distribution based
          |__ [2.2.2] Surrogate Model Based Optimization (SBO)
               |__ [2.2.2.1] Single surrogate based
               |__ [2.2.2.2] Multi-fidelity based
               |__ [2.2.2.3] Evolutionary surrogate based
               |__ [2.2.2.4] Ensemble surrogate based
```

*Figure 1:* Taxonomy of model-based approaches in GO

refers to category [2.2.]. This category will be referred to as *model-based optimization* (MBO).

We differentiate between models, which use a distribution ([2.2.1]) and models that use an explicit surrogate model ([2.2.2]). Model-based optimization is the first choice for many optimization problems in industry. Section 3 describes typical applications, illustrating the practical relevance of MBO. Fundamental aspects of and recent advances in surrogate-model based optimization are discussed in Section 4. Strategies for selecting and evaluating surrogates are presented in Section 5. Two MBO algorithms, namely EvoLS and SPO, are presented in Section 6. Finally, a summary and an outlook are given in Section 7.

# 2. Stochastic Search Algorithms

## 2.1 Instance-Based Algorithms

Instance-based algorithms ([2.1]) maintain a *single solution*, $\mathbf{x}$, or *population*, $P(t)$, of candidate solutions. The iteration or time step is denoted as $t$. The construction of new candidate solutions depends explicitly on the previously generated solutions. Simulated annealing [36], evolutionary algorithms (EAs) [4], and tabu search [19] are prominent representatives of this category. The key elements of instance-based algorithms are shown in Algorithm 1.

## 2.2 MBO: Model-Based Algorithms

Model-based optimization algorithms ([2.2]) generate a population of new candidate solutions $P'(t)$ by sampling from a model (or a distri-

**Algorithm 1** Instance-based Algorithm

```
t = 0. SetInitialPopulation(P)
Evaluate(P).
while not TerminationCriterion() do
    Generate a set of new candidate solutions P'(t ) according to a
    specified random mechanism.
    Update the current population P(t+1) based on population P(t)
    and candidate solutions in P'(t).
    Evaluate(P(t + 1)).
    t = t + 1.
end while
```

bution). The model (distribution) reflects structural properties of the underlying true function $f$. They are based on the idea that by adapting the model (or the distribution), the search is directed into regions with improved solutions.

One of the key ideas in MBO is the replacement of expensive, high fidelity, fine grained function evaluations, $f(\mathbf{x})$, with evaluations, $\hat{f}(\mathbf{x})$, of an adequate cheap, low fidelity, coarse grained model, $M$. After presenting typical examples in Section 3, two different approaches for generating cheap models will be presented in Section 4.

## 3. Applications of MBO

Simulation-based design of complex engineering problems, e.g., structural design of vehicles, use *computational fluid dynamics* (CFD) and *finite element modeling* (FEM) methods. The solvers require a large number of computer simulations to guarantee an exact solution. Hence, this is one of the most popular and successful application areas for MBO. There are two variants of MBO in this field of application: (i) metamodel (category [2.2.2.1]) and (ii) multi-fidelity approximation (category [2.2.2.2]) approaches. The former approach uses *one or several different* metamodels, whereas the latter uses several instances with different parameterizations of the *same* metamodel.

### 3.1 Metamodels

There are several publications that describe metamodeling approaches in aerospace design. The development of effective numerical methods for managing the use of approximation concepts in optimization for a 31-variable helicopter rotor design, which was part of a collaboration between Boeing, IBM, and Rice University, is described by Booker et al. [7, 8]. Giannakoglou [18] discusses an aerodynamic shape design problem. Queipo et al. [51] present a multi-objective optimal design of a liquid rocket injector and discuss fundamental problems that arise in MBO. A surrogate-assisted evolutionary optimization framework, which is applied to an airfoil shape optimization problem using computational fluid dynamic is presented in [70]. Forrester and Keane [17] describe recent advances of MBO in aerospace design.

The design of ship propellers in the field of ship propulsion technology is described by Emmerich and Hundemer [14]. The authors model the features of a propeller design as a function of its resulting efficiency, torque coefficients, thrust coefficients, and cavitation. An implementation of a first-order potential-based panel method is used to calculate the hydrodynamic performance of a given propeller.

Li et al. [44] describe the optimization of feature detectors in ultrasound images. They present a study of *radial basis function networks* (RBFN) for metamodeling in heterogeneous, i.e., mixed-integer, parameter spaces.

Although the application of metamodeling techniques has progressed remarkably in the past last decades, the question remains "How far have we really come?" This issue is addressed in [59].

## 3.2 Multi-Fidelity Approximation

In addition to metamodels, *multi-fidelity metamodeling* methods have been developed. Multi-fidelity metamodeling uses several models of the same real system, where each model has its own degree of detail representing the real process. A typical example is the use of several simulation models with different grid sizes in FEM [26].

Sun et al. [60] describe a multi-fidelity optimization approach for sheet metal forming process. Further examples of multi-fidelity metamodeling are presented in [63]. The authors analyze the performance of Kriging [33] when multi-fidelity gradient data is introduced along with multi-fidelity function data to approximate black-box simulations.

Koziel et al. [39] present a methodology for fast multi-objective antenna optimization with *co-Kriging*. Co-Kriging is an extension of Kriging, which uses the correlations between the models of various fidelities, so that cheap- and expensive simulation data can be combined into one metamodel [15, 35]. Co-Kriging-based sequential design strategies are presented by Le Gratiet and Cannamela [43]. The authors simulate a spherical tank under internal pressure. Further applications from the water industry are published by Razavi et al [53]. Tuo et al. [62] proposed a finite-element analysis with its mesh density as the tuning parameter. A problem in casting simulation is used to illustrate this approach.

Kleijnen [37] presents an overview of the most recent approaches in simulation practice. The book covers multi-fidelity metamodeling as well.

## 4. Key Elements of MBO

This section describes two different MBO approaches: (i) distribution based ([2.2.1]) and (ii) surrogate-model based optimization ([2.2.2.]).

### 4.1 Distribution-Based Approaches

If the metamodel is a distribution, the most basic form of an MBO can be implemented as shown in Algorithm 2:

**Algorithm 2** Distribution-based Algorithm

$t = 0$. Let $p(t)$ be a probability distribution.
**while** not TerminationCriterion() **do**
  Randomly generate a population of candidate solutions $P(t)$ from $p(t)$.
  Evaluate($P(t)$).
  Update the distribution using population (samples) $P(t)$ to generate a new distribution $p(t+1)$.
  $t = t + 1$.
**end while**

Distribution-based algorithms generate a sequence of iterates (probability distributions) $\{p(t)\}$ with the hope that

$$p(t) \to p^* \text{ as } t \to \infty,$$

where $p^*$ is a limiting distribution, which assigns most of its probability mass to the set of optimal solutions. So it is the probability distribution (as opposed to candidate solutions as in instance-based algorithms) that is propagated from one iteration to the next.

*Estimation of distribution algorithms* (EDA) are popular distribution-based algorithms, which became popular in the field of evolutionary algorithms [41]. Variation operators such as mutation and recombination, which modify candidate solutions (so-called individuals in EA), were replaced by a distribution based procedure: the new population of candidate solutions is generated according to the probability distribution induced or estimated from the promising candidate solution from the current population. Larraaga and Lozano [41] review different ways of using probabilistic models as EDA instantiations.

Although distribution-based approaches play an important role in GO, they will not be discussed further in this article. The reader is referred to [24]. The authors discuss advantages and outline many of the different types of EDAs. In addition, Hu et al. [25] present recent approaches and a unified view on distribution-based approaches. We will concentrate on surrogate model-based approaches, which have their origin in statistical design and analysis of experiments, especially in response surface methodology.

## 4.2 Surrogate Model-Based Approaches

In general, surrogates are used, when the outcome of a process cannot be directly measured. Surrogates imitate the behavior of the real model as closely as possible, while being computationally cheaper to evaluate. The surrogate model is also known as a response surface, metamodel, approximation, coarse grained, or simply the *cheap* model. Simple surrogate models are constructed using a data-driven approach. They can be refined by integrating additional points or domain knowledge, e.g., constraints, into the surrogate.

A minimalistic *surrogate model-based optimization* (SBO) algorithm is shown in Algorithm 3. A wide range of surrogates was applied in the last

**Algorithm 3** Surrogate Model Based Optimization (SBO) Algorithm

$t = 0$. SetInitialPopulation($P(t)$)
Evaluate($P(t)$)
**while** not TerminationCriterion() **do**
  Use $P(t)$ to build a cheap model $M(t)$
  $P'(t+1) =$ GlobalSearch($M(t)$)
  Evaluate($P'(t+1)$)
  $P(t+1) \subseteq P(t) + P'(t+1)$
  $t = t + 1$
**end while**

decades. Classical regression models such as *polynomial regression* or response surface methodology [9], *support vector machines* (SVM) [65], artificial neural networks [72], radial basis functions [49], or *Gaussian process* (GP) models, which are sometimes referred to as *design and analysis of computer experiments* or Kriging [2, 11, 38, 56, 57] are the most prominent approaches. Forrester et al. [16] present a comprehensive introduction to SBO with several examples. Table 1 in [66] lists popular metamodeling techniques and the related components such as experimental design, sampling methods, metamodels, and model fitting techniques.

### 4.3 Surrogate-Assisted Evolutionary Algorithms

*Surrogate-assisted evolutionary algorithms* (category [2.2.2.3]) are evolutionary algorithms that decouple the evolutionary search and the direct evaluation of the objective function. A cheap surrogate model, $M$, replaces evaluations of an expensive objective function, $f$.

A combination of a genetic algorithm and neural networks for aerodynamic design optimization is suggested in [22]. Ratle [52] creates an approximate model of the fitness landscape using Kriging interpolation to accelerate the convergence of EAs. Jin and et al. [31] investigate the convergence property of an *evolution strategy* (ES) with neural network based fitness evaluations. Emmerich et al. [13] present several MBO approaches for ES. Jin [30] presents a survey of surrogate-assisted evolutionary algorithms approaches. Jin and Sendhoff [32] use clustering techniques and neural networks ensembles to reduce the number of function evaluations. Branke and Schmidt [10] propose not evaluate every candidate solution (individual), but to just estimate the objective function value of some of the individuals. The reduction in the number of function evaluations is obtained by estimating an individual's function value on the basis of previously observed objective function values of neighboring individuals. Zhou et al. [70] present a surrogate-assisted EA framework, which uses computationally cheap hierarchical surrogate models constructed through online learning to replace the exact computationally expensive objective functions during evolutionary search.

## 5. Quality Criteria: How to Select Surrogates

The model building and selection process is crucial for the effectivity and efficiency of SBO. Fundamental for the improvement of a selected surrogate model as well as for the selection of an alternative surrogate model type is the evaluation of the expensive (true) objective function, which requires the determination of sample points. In the selection of adequate sample points, two conflicting goals have to be satisfied. The sample points can be selected with respect to

- exploration, i.e., improving the model quality (related to the model $M$) or
- exploitation, i.e., improving the optimization and determining the optimum (related to the objective function $f$).

Furthermore, regarding the model choice, the user can decide whether to use a

- single model, i.e., one unique global model is used during the optimization or
- multiple models, i.e., an ensemble of different, possibly local, models.

The static SBO uses a single, global surrogate model, which is usually refined by *adaptive sampling*. The same model type, e.g., Kriging interpolation, is used during the optimization. This is category [2.2.2.1] in Fig. 1.

## 5.1 Model Refinement

*Adaptive sampling,* a well-known selection strategy, proceeds as follows: An initial model, which uses a limited amount of sample points from the expensive objective function, is refined during the optimization. Adaptive sampling identifies new points, so-called *infill points*. Adaptive sampling tries to find a balance between exploration, i.e., improving the overall, global quality of the surrogate model, and exploitation, i.e., improving the local quality (in the region of the actual optimum), of the surrogate model. A popular adaptive sampling method is *expected improvement* (EI) [34, 45], which is discussed in [33]. The EI approach handles the initialization and refinement of a surrogate model, but not the selection of the model itself. The popular *efficient global optimization* (EGO) algorithm uses a Kriging model, because Kriging inherently determines the prediction variance, which is necessary for the EI criterion.

But there is no proof that Kriging is the best choice. Alternative surrogate models, e.g., regression trees, support vector machine, or lasso and ridge regression may be better suited. An *a priory* selection of the best suited surrogate model is conceptually impossible in the framework treated in this article, because of the black-box setting described in Section 1.

## 5.2 Multiple Models

Instead of using one surrogate model only, several models $M_i$, $i = 1, 2, \dots, p$, can be generated and evaluated in parallel. Each model uses the same candidate solutions (from the population $P$) and results from expensive function evaluations.

Multiple models can also be used to partition the search space. The *tree-based Gaussian process* (TGP) approach uses regression trees to partition the search space into separate regions and to fit local GP surrogates in each region [21]. Nelson et al. [47] propose an algorithm, that creates a tree-based partitioning of an aerodynamic design space and employs independent Kriging surfaces in each partition. Couckuyt et al. [12] propose to combine an *evolutionary model selection* (EMS) algorithm with the EI criterion in order to dynamically select the best performing surrogate model type at each iteration of the EI algorithm. A new expensive sample point, $\mathbf{x}'$, is chosen based on the EI criterion at each iteration step $t$. The point $\mathbf{x}'$ itself is based on the best surrogate model found by the EMS algorithm.

In the last decade, ensembles of surrogate models gained popularity (category [2.2.2.4]) in Fig. 1. Zerpa et al. [69] use multiple surrogate models and build an adaptive weighted average model of the individual surrogates. Goel at al. [20] explore the possibility of using the best surrogate model or a weighted average surrogate model instead of one single model. Model quality, i.e., the errors in surrogates, is used to determine the weights assigned to each model. Sanchez et al. [55] present a weighted-sum approach for the selection of model ensembles. The models for the ensemble are chosen based on their performance and the weights are adaptive and inversely proportional to the local modeling errors.

Recent approaches such as the evolvability learning of surrogates approach implement local models for each offspring individually [42]. This results in an adaptive semi-partition [40] of the search space.

## 5.3 Criteria for Selecting a Surrogate

Note, this paragraph does not consider the selection of a new sample point as done in EI. Here, we consider criteria for the selection of one (or several) surrogate models, e.g., Kriging models or SVMs [65].

Conventionally, surrogate models are assessed and chosen according to their estimated true error [29, 58]. The *mean absolute error* (MAE) and the *root mean square error* (RMSE) are commonly used as performance metrics. Error measures are discussed in [28]. Willmott and Matsuura [67] presents a comparison of MAE and RMSE. Generally, attaining a surrogate model that has minimal error is the desired feature. Methods from statistics, statistical learning [23], and machine learning [46], such as the simple holdout approach, cross-validation, and the bootstrap are used to choose adequate surrogate models. Tenne and Armfield [61] propose a surrogate-assisted memetic algorithm which generates accurate surrogate-models using multiple cross-validation tests. However, the definition of the corresponding training sets (sampling) represents a critical issue for the accuracy and efficiency of the meta-models.

The model error is not the only criterion for selecting surrogate models. In contrast to the surrogate model selection approaches so far, the evolvability learning of surrogates approach [42], which will be presented in Section 6.1, uses fitness improvement for determining the quality of surrogate models in enhancing search improvement.

# 6. Examples

## 6.1 Evolvability Learning of Surrogates

The *evolvability learning of surrogates* (EvoLS) algorithm, which is introduced by Le et al. [42], belongs to the category of *surrogate-assisted evolutionary algorithms* ([2.2.2.3]).

The authors of EvoLS recommend selecting surrogate models that enhance search improvement in the context of optimization. EvoLS processes information about the (i) different fitness landscapes, (ii) state of the search, and (iii) characteristics of the search algorithm to statistically determine the so-called *evolvability* of each surrogate model. The evolvability of a surrogate model estimates the expected improvement of the objective function value that the new candidate solution has gained after a local search has been performed on the related surrogate model. Three basic steps are necessary for calculating the evolvability (a detailed calculation is presented in [42]):

- Variation. Let $\mathbf{x}$ denote the parent and $\mathbf{y}$ be the offspring generated from $\mathbf{x}$ by evolutionary variation operators, e.g., mutation and/or recombination. Le at al. [42] make a simplified assumption of uniformity in the offspring distribution. Let $V(R)$ denote the volume of an $n$-dimensional cuboid

  $$R = \left[\min_{j=1..N}\{x_j^{(i)}\}, \max_{j=1..N}\{x_j^{(i)}\}\right]_{i=1,..,n}.$$

  The density distribution is modeled as

  $$P(\mathbf{y}\,|P(t),\mathbf{x}) = \mathcal{U}(R) = \begin{cases} 1/(R) & \text{if } y \in R \\ 0 & \text{otherwise.} \end{cases}$$

  The evolutionary variation operators recombination and uniform mutation force the offspring to be located in the $n$-dimensional region $R$. To determine the probability at time step $t$ of moving from parent $\mathbf{x}$ via stochastic variation, the followoing weights can be used:

$$w_i(\mathbf{x}) = \frac{P(\mathbf{y}_i \,|P(t), \mathbf{x})}{\sum_{j=1}^{K} P(\mathbf{y}_j \,|P(t), \mathbf{x})}.$$

The weight measures the influence of the samples $(\mathbf{y}_i, \varphi_M(y_i))$ on the evolvability.

- Local search. After recombination and mutation, a local search is performed. It uses a local metamodel, $M$, for each offspring. The local optimizer, $\varphi_M$, uses an offspring $\mathbf{y}$ as an input and returns $\mathbf{y}^*$ as the refined offspring. The local optimizer on the surrogate model guarantees (theoretically) convergence to the stationary point of the exact objective function [1, 48].

- Evolvability. Finally, the evolvability measure can be estimated as follows:

$$Ev_M(\mathbf{x}) = f(\mathbf{x}) - \sum_{i=1}^{K} f(\mathbf{y}_i^*) \times w_i(\mathbf{x}).$$

## 6.2 Sequential Parameter Optimization

Early versions of the *sequential parameter optimization* (SPO) combined methods from *design of experiments* (DOE), *response surface methodology* (RSM), *design and analysis of computer experiments* (DACE), and regression trees for the analysis of algorithms [3, 5, 6]. The SPO was developed as a tool for the analysis and for an understanding of the working principles of EAs. The SPO tools might as well be integrated into the evolutionary loop and therefore improve performance of an EA. This consideration lays the cornerstone for the development of the SPO as an optimizer.

Subsequent versions of the SPO use a sequential, model based approach to optimization. Nowadays, the SPO is an established parameter tuner and an optimization algorithm, which has been extended in several ways. For example, Hutter et al. [27] benchmark an SPO derivative, the so-called *sequential model-based algorithm configuration* (SMAC) procedure, on the BBOB set of blackbox functions. They demonstrate that with a small budget of $10 \times d$ evaluations of $d$-dimensional functions, SMAC in most cases outperforms the state-of-the-art blackbox optimizer CMA-ES.

The most recent version, SPO2, is currently under development. It will integrate state-of-the-art ensemble learners. The SPO2 *ensemble engine* can be briefly outlined as follows: The portfolio of surrogate models includes a pleiotropy of metamodels such as regression trees and random forest, *least angle regression* (LARS), and Kriging. The SPO2 ensemble engine uses cross validation to select an improved model from the portfolio of candidate models [64]. It implements methods for creating a weighted combination of several surrogate models to build the improved model and methods, which use stacked generalization to combine several level-0 models of different types with one level-1 model into an ensemble [68]. The level-1 training algorithm is typically a relatively simple linear model.

Preliminary results indicate that the SPO2 ensemble engine can lead to significant performance improvements of the SPO algorithms, which is illustrated by the following example: Rebolledo et al. [54] present a comparison of different data driven modeling methods. The first instance of a data driven linear Bayesian model is compared with several linear regression models, a Kriging model and a genetic programming model. The models are built on industrial data for the development of a robust gas sensor. The data contain limited amount of samples and a high variance. The mean square error of the models implemented in a test dataset is used as the comparison strategy. Two sensors were tested in this comparison. The mean squared errors are as follows. Linear model (0.76), OLS (0.79), lasso (0.56), Kriging (0.57), Bayes (0.79), and genetic programming (0.58). SPO2 obtained an MSE of 0.38, which outperforms the best model. Results from the second sensor read as follow: Linear model (0.67), OLS (0.80), lasso (0.49), Kriging (0.49), Bayes (0.79), and genetic programming (0.27). Here, SPO2 obtained an MSE of 0.29.

This first real-world application example demonstrates the potential of SBO with ensembles (category [2.2.2.4]).

## 7. Summary

Especially in the engineering domain, model-based approaches are probably the most efficient methods for expensive and time-demanding real-world optimization problems. This article proposed a taxonomy of model based algorithms for global optimization problems. The taxonomy was developed from an algorithm-centered perspective. The categorization scheme, which started with a bird's eye view on GO, was refined as summarized in Fig. 1. Finally, working principles of two state-of-the-art MBO algorithms were shown. EvoLS, which constructs a metamodel for every new candidate solution, and SPO2, which uses an ensemble engine to combine a broad variety of surrogate models. The survey presented in the first sections of this article as well as the examples in Section 6 emphasize the trend to ensemble based metamodels.

**Acknowledgement:** This work has been supported by the *Bundesministeriums fur Wirtschaft und Energie* under the grants KF3145101WM3 und KF3145103WM4. This work is part of a project that has received funding from the *European Unions Horizon 2020 research and innovation program* under grant agreement No 692286.

## References

[1] N. M. Alexandrov, J. E. Dennis Jr, R. M. Lewis, and V. Torczon. A trust-region framework for managing the use of approximation models in optimization. *Structural Optimization*, 15(1):16–23, 1998.

[2] A. B. Antognini and M. Zagoraiou. Exact optimal designs for computer experiments via Kriging metamodelling. *Journal of Statistical Planning and Inference*, 140(9):2607–2617, 2010.

[3] T. Bartz-Beielstein. Experimental Analysis of Evolution Strategies—Overview and Comprehensive Introduction. Technical report, Nov. 2003.

[4] T. Bartz-Beielstein, J. Branke, J. Mehnen, and O. Mersmann. Evolutionary Algorithms. *WIREs Data Mining and Knowledge Discovery*, 4:178–195, 2014.

[5] T. Bartz-Beielstein, C. Lasarczyk, and M. Preuss. Sequential Parameter Optimization. *Proceedings of the Congress on Evolutionary Computation (CEC)*, pages 773–780, 2005.

[6] T. Bartz-Beielstein, K. E. Parsopoulos, and M. N. Vrahatis. Design and analysis of optimization algorithms using computational statistics. *Applied Numerical Analysis and Computational Mathematics*, 1(2):413–433, 2004.

[7] A. J. Booker, J. E. Dennis Jr, P. D. Frank, D. B. Serafini, and V. Torczon. Optimization Using Surrogate Objectives on a Helicopter Test Example. In *Computational Methods for Optimal Design and Control*, pages 49–58. Birkhäuser Boston, Boston, MA, 1998.

[8] A. J. Booker, J. E. Dennis Jr, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1):1–13, 1999.

[9] G. E. P. Box and K. B. Wilson. On the Experimental Attainment of Optimum Conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45, 1951.

[10] J. Branke and C. Schmidt. Faster convergence by means of fitness estimation. *Soft Computing*, 9(1):13–20, 2005.

[11] D. Büche, N. N. Schraudolph, and P. Koumoutsakos. Accelerating Evolutionary Algorithms With Gaussian Process Fitness Function Models. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 35(2):183–194, 2005.

[12] I. Couckuyt, F. De Turck, T. Dhaene, and D. Gorissen. Automatic surrogate model type selection during the optimization of expensive black-box problems. *Proceedings of the Winter Simulation Conference (WSC)*, pages 4269–4279, 2011.

[13] M. Emmerich, A. Giotis, M. özdemir, T. Bäck, and K. Giannakoglou. Metamodel-assisted evolution strategies. *Lecture Notes in Computer Science*, 2439:361–370, 2002.

[14] M. Emmerich, J. Hundemer, M.-C. Varcol, B. Naujoks, and M. Abdel-Maksoud. Design Optimization of Ship Propellers by Means of Advanced Metamodel-Assisted Evolution Strategies. *Proceedings of the International Conference on Design Optimization (ERCOFTAC)*, 2006 .

[15] A. Forrester, A. Sóbester, and A. Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 463(2088):3251–3269, 2007.

[16] A. Forrester, A. Sóbester, and A. Keane. *Engineering Design via Surrogate Modelling*. Wiley, 2008.

[17] A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, 2009.

[18] K. C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, 38(1):43–76, 2002.

[19] F. Glover and M. Laguna. Tabu Search. In C. Reeves (Ed.) *Modern Heuristic Techniques for Combinatorial Problems*, Oxford, U.K., Blackwell Scientific Publishing, 1993.

[20] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo. Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 33(3):199–216, 2006.

[21] R. B. Gramacy. tgp: An R Package for Bayesian Nonstationary, Semiparametric Nonlinear Regression and Design by Treed Gaussian Process Models. *Journal of Statistical Software*, 19(9):1–46, 2007.

[22] P. Hajela and E. Lee. Topological optimization of rotorcraft subfloor structures for crashworthiness considerations. *Computers & Structures*, 64(1-4):65–76, 1997.

[23] T. Hastie. *The elements of statistical learning : data mining, inference, and prediction*. Springer, New York, 2nd ed., 2009.

[24] M. Hauschild and M. Pelikan. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3):111–128, 2011.

[25] J. Hu, Y. Wang, E. Zhou, M. C. Fu, and S. I. Marcus. A Survey of Some Model-Based Methods for Global Optimization. In D. Hernández-Hernández and J. A. Minjárez-Sosa (Eds.) *Optimization, Control, and Applications of Stochastic Systems*, pages 157–179. Birkhäuser Boston, Boston, 2012.

[26] E. Huang, J. Xu, S. Zhang, and C. H. Chen. Multi-fidelity Model Integration for Engineering Design. *Procedia Computer Science*, 44:336–344, 2015.

[27] F. Hutter, H. Hoos, and K. Leyton-Brown. An Evaluation of Sequential Model-based Optimization for Expensive Blackbox Functions. *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 1209–1216, 2013.

[28] R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[29] R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23(1):1–13, 2001.

[30] Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2003.

[31] Y. Jin, M. Olhofer, and B. Sendhoff. On Evolutionary Optimization with Approximate Fitness Functions. *Proceedings of the Annual Conference on Genetic and Evolutionary Computation (GECCO)*, 2000.

[32] Y. Jin and B. Sendhoff. Reducing Fitness Evaluations Using Clustering Techniques and Neural Network Ensembles. *Lecture Notes in Computer Science*, 3102:688–699, 2014.

[33] D. R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21:345–383, 2001.

[34] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13:455–492, 1998.

[35] M. Kennedy. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[36] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, 1983.

[37] J. P. C. Kleijnen. *Design and Analysis of Simulation Experiments*. International Series in Operations Research and Management Science. Springer International Publishing, 2015.

[38] J. P. C. Kleijnen. Kriging metamodeling in simulation: A review. *European Journal of Operational Research*, 192(3):707–716, 2009.

[39] S. Koziel, A. Bekasiewicz, I. Couckuyt, and T. Dhaene. Efficient Multi-Objective Simulation-Driven Antenna Design Using Co-Kriging. *IEEE Transactions on Antennas and Propagation*, 62(11):5900–5905, 2014.

[40] M. Kryszkiewicz, J. F. Peters, H. Rybinski, and A. Skowron (Eds.) *Rough Sets and Intelligent Systems Paradigms*, volume 4585 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 2007.

[41] P. Larraaga and J. A. Lozano. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer, Boston MA, 2002.

[42] M. N. Le, M. N. Le, Y. S. Ong, Y. S. Ong, S. Menzel, S. Menzel, Y. Jin, Y. Jin, B. Sendhoff, and B. Sendhoff. Evolution by adapting surrogates. *Evolutionary Computation*, 21(2):313–340, 2013.

[43] L. Le Gratiet and C. Cannamela. Kriging-based sequential design strategies using fast cross-validation techniques with extensions to multi-fidelity computer codes. *arXiv.org*, Oct. 2012.

[44] R. Li, M. T. M. Emmerich, J. Eggermont, E. G. P. Bovenkamp, T. Bäck, J. Dijkstra, and J. H. C. Reiber. Metamodel-assisted mixed integer evolution strategies and their application to intravascular ultrasound image analysis. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 2764–2771, 2008.

[45] J. Mockus, V. Tiesis, and A. Zilinskas. Bayesian Methods for Seeking the Extremum. In L. C. W. Dixon and G. P. Szegö (Eds.) *Towards Global Optimization*, pages 117–129. Amsterdam, 1978.

[46] K. P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

[47] A. Nelson, J. Alonso, and T. Pulliam. Multi-Fidelity Aerodynamic Optimization Using Treed Meta-Models. *Proceedings of the Fluid Dynamics and Co-located Conferences*, 2007.

[48] Y. S. Ong, P. B. Nair, and A. J. Keane. Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling. *AIAA Journal*, 41(4):687–696, 2003.

[49] M. Powell. Radial Basis Functions. *Algorithms for Approximation*, 1987.

[50] M. Preuss. *Multimodal Optimization by Means of Evolutionary Algorithms*. Natural Computing Series. Springer International Publishing, Cham, 2015.

[51] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Kevin Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, 2005.

[52] A. Ratle. Accelerating the Convergence of Evolutionary Algorithms by Fitness Landscape Approximation. *Lecture Notes in Computer Science*, 1498:87–96, 1998.

[53] S. Razavi, B. A. Tolson, and D. H. Burn. Review of surrogate modeling in water resources. *Water Resources Research*, 48(7):n/a–n/a, 2012.

[54] M. A. Rebolledo Coy, S. Krey, T. Bartz-Beielstein, O. Flasch, A. Fischbach, and J. Stork. Modeling and Optimization of a Robust Gas Sensor. Technical Report 03/2016, Cologne Open Science, Cologne, 2016.

[55] E. Sanchez, S. Pintos, and N. V. Queipo. Toward an Optimal Ensemble of Kernel-based Approximations with Engineering Applications. *Proceedings of the IEEE International Joint Conference on Neural Network Proceedings*, pages 2152–2158, 2006.

[56] T. J. Santner, B. J. Williams, and W. I. Notz. *The Design and Analysis of Computer Experiments*. Springer, Berlin, Heidelberg, New York, 2003.

[57] M. Schonlau. Computer Experiments and Global Optimization. PhD thesis, University of Waterloo, Ontario, Canada, 1997.

[58] L. Shi and K. Rasheed. A Survey of Fitness Approximation Methods Applied in Evolutionary Algorithms. In *Computational Intelligence in Expensive Optimization Problems*, pages 3–28. Springer, Berlin, Heidelberg, 2010.

[59] T. Simpson, V. Toropov, V. Balabanov, and F. Viana. Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization: A Review of How Far We Have Come - Or Not. *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 1–22, 2012.

[60] G. Sun, G. Li, S. Zhou, W. Xu, X. Yang, and Q. Li. Multi-fidelity optimization for sheet metal forming process. *Structural and Multidisciplinary Optimization*, 44(1):111–124, 2011.

[61] Y. Tenne and S. W. Armfield. A Versatile Surrogate-Assisted Memetic Algorithm for Optimization of Computationally Expensive Functions and its Engineering Applications. In *Success in Evolutionary Computation*, pages 43–72. Springer, Berlin, Heidelberg, 2008.

[62] R. Tuo, C. F. J. Wu, and D. Yu. Surrogate Modeling of Computer Experiments With Different Mesh Densities. *Technometrics*, 56(3):372–380, 2014.

[63] S. Ulaganathan, I. Couckuyt, F. Ferranti, E. Laermans, and T. Dhaene. Performance study of multi-fidelity gradient enhanced kriging. *Structural and Multidisciplinary Optimization*, 51(5):1017–1033, 2014.

[64] M. J. van der Laan and S. Dudoit. Unified Cross-Validation Methodology For Selection Among Estimators and a General Cross-Validated Adaptive Epsilon-Net Estimator: Finite Sample Oracle Inequalities and Examples. 2003.

[65] V. N. Vapnik. *Statistical learning theory*. Wiley, 1998.

[66] G. G. Wang and S. Shan. Review of Metamodeling Techniques in Support of Engineering Design Optimization. *Journal of Mechanical Design*, 129(4):370–380, 2007.

[67] C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(7982):1–4, 2005.

[68] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[69] L. E. Zerpa, N. V. Queipo, S. Pintos, and J.-L. Salager. An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates. *Journal of Petroleum Science and Engineering*, 47(3-4):197–208, 2005.

[70] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum. Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 37(1):66–76, 2007.

[71] M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo. Model-Based Search for Combinatorial Optimization: A Critical Survey. *Annals of Operations Research*, 131(1-4):373–395, 2004.

[72] J. M. Zurada. Analog implementation of neural networks. *IEEE Circuits and Devices Magazine*, 8(5):36–41, 1992.