Open Issues in Surrogate-Assisted Optimization

Jörg Stork, Martina Friese, Martin Zaefferer, Thomas Bartz-Beielstein, Andreas Fischbach, Beate Breiderhoff, Boris Naujoks and Tea Tušar

Abstract Surrogate-assisted optimization was developed for handling complex and costly problems, which arise from real-world applications. The main idea behind surrogate-assisted optimization is to optimally exhaust the available information to lower the amount of required expensive function evaluations thus saving time, resources and the related costs. This chapter outlines the existing challenges in this field that include benchmarking, constraint handling, constructing ensembles of surrogates and solving discrete and/or multi-objective optimization problems. We discuss shortcomings of existing techniques, propose suggestions for improvements and give an outlook on promising research directions. This is valuable for practitioners and researchers alike, since the increased availability of computational resources on

J. Stork · M. Friese (✉) · M. Zaefferer · T. Bartz-Beielstein · A. Fischbach · B. Breiderhoff · B. Naujoks
Institute of Data Science, Engineering, and Analytics, Faculty of Computer Science and Engineering Science, TH Köln, Steinmüllerallee 1, 51643 Gummersbach, Germany
e-mail: joerg.stork@th-koeln.de

M. Friese
e-mail: martina.friese@th-koeln.de; martina.friese@gmail.com

M. Zaefferer
e-mail: martin.zaefferer@th-koeln.de

T. Bartz-Beielstein
e-mail: thomas.bartz-beielstein@th-koeln.de

A. Fischbach
e-mail: andreas.fischbach@th-koeln.de

B. Breiderhoff
e-mail: beate.breiderhoff@th-koeln.de

B. Naujoks
e-mail: boris.naujoks@th-koeln.de

T. Tušar
Department of Intelligent Systems, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
e-mail: tea.tusar@ijs.si

T. Bartz-Beielstein et al. (eds.), *High-Performance Simulation-Based Optimization*, Studies in Computational Intelligence 833,
https://doi.org/10.1007/978-3-030-18764-4_10

the one hand and the continuous development of new approaches on the other hand raise many intricate new problems in this field.

1 Introduction

The state-of-the-art in handling complex and costly problems arising from real-world applications is the use of surrogate models in optimization. These real-world applications commonly belong to the class of black-box problems, where the available problem information, e.g., mathematical equations or other exploited problem knowledge is very sparse. Often, the only way to extract any information is the evaluation of candidate solutions. These evaluations can consist of difficult and time-consuming simulations or even hazardous physical experiments. The main intention of surrogate-assisted optimization is to reduce the time, resources and the related costs by efficiently using all available information thus lowering the amount of required function evaluations.

The main contribution of this chapter is to outline existing challenges in this field. We show where established techniques are lacking or need to be revised. Furthermore we present an outlook on promising research directions. This is crucial to practitioners as well as researchers, since the increased availability of computational resources as well as the development in the field raise many intricate new problems.

An important open issue, presented in Sect. 1, is the selection of an adequate set of benchmark or test functions, which can be used to evaluate algorithm and modeling performance. There are no tools to determine the applicability of surrogate-based methods in an objective manner. However, on certain data sets, different algorithms can be compared using adequate methods from statistics.

Constraint handling, a great challenge in the field, is discussed in Sect. 2. Different solution strategies can be considered. These strategies have to take care of infeasible offspring, which are generated by feasible ancestors. The most simple strategy, i.e., just omitting these solutions, might not be the best choice. Alternatively, constraints might be modeled by surrogates as well, or integrated into the algorithms. However, the determination of an optimal strategy is a difficult task.

Instead of using only one surrogate, several surrogates can be generated and evaluated in parallel as examined in Sect. 3. All surrogates use the same solutions evaluated by the expensive function. Multiple surrogates can also be used to partition the search space. In the last decade, ensembles of surrogates gained popularity. The surrogates 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 surrogates for each offspring individually [48] or apply sophisticated stacking methods as described in [5]. However, the selection of the best ensemble building methods remains an open issue.

Another important challenge is the handling of combinatorial/discrete search spaces. Many optimization problems in practice cannot be represented by vectors

of real variables. Rather, they have some discrete structures, e.g., categorical variables, strings, sequences, permutations, trees, or graphs in general. Standard surrogate modeling methods will often fail to deal with these methods efficiently. However, current research provides the tools to solve this issue. Most importantly, the usage of similarity measures allows to transfer Gaussian process models (also called Kriging) [28] to the discrete domain. Section 4 introduces problems and promising solution approaches for discrete surrogate-based optimization.

In multi-objective optimization [23], incorporating surrogates for complex, demanding optimization problems became popular in recent years as well. Several attempts have been proposed to effectively and efficiently combine approaches from both areas, surrogate-assisted and multi-objective optimization, for example [47]. However, the field still misses further specifically tailored ideas as well as basics to allow for an efficient start and a rigorous analysis. To this end, Sect. 5 summarizes already available achievements and focuses on missing features that need to be added in the future.

Section 6 sheds some light on additional topics related to dynamic problems, problem dimensionality and noisy problems. The handling of dynamic optimization problems plays an important role, because many real-world problems occur in time-varying environments. Another unsolved problem arises from high-dimensional and large-scale data. Existing surrogates might not be applicable, because of their computational complexity. New surrogates might need to be considered as well as new integration schemes. This is especially important since many applications profit from parallelization of simulation systems, e.g., in computational fluid dynamics simulation. Thus, larger and more complex problems can be considered. Noise handling is also an important research topic. Especially, how to handle noise during the surrogate building process.

Finally, Sect. 7.3 ends the chapter with some concluding remarks.

2 Benchmarking

Benchmarking is a core issue when developing algorithms. Optimization algorithm developers need test functions to (i) investigate and understand algorithm behavior and (ii) compare competing algorithms.

In the past, several test suites for optimization algorithms were presented and considered as state of the art. However, there are several drawbacks of these classical test suites, namely:

- Problem instances are mostly artificial and have no direct link to real-world settings.
- Since there is a fixed number of test instances, algorithms can be fitted or tuned to this specific and very limited set of test functions. As a consequence, studies (benchmarks) provide insight how these algorithms perform on this specific set of test instances, but no insight on how they perform in general is gained.

- Statistical tools for the comparisons of several algorithms on several test problem instances are relatively complex and not easy to analyse.

To retrieve a meaningful benchmark, practitioners and researchers have to select a number of test instances from a huge field of test classes and interpret results of algorithms applied on them. This selection of problems and methods with their parameters is crucial for benchmarking results.

A widely used practice is to select a fixed number of different test instances of potentially different difficulty. These instances are fulfilling some criteria, e.g., difficulty (non-separability, resistance to hill-climbing) and diversity [77]. The results of newly proposed or otherwise favored methods should be compared with those of competing algorithms. Here, an important issue is the extensive research and experimentation already done with plenty of methods on lots of well-known test functions (see, for example, the COCO platform with data from more than 170 optimization algorithms [35]). Optimization methods stemming from applications are mostly only able to compete with established algorithms on the chosen instances if they are extensively tuned on them.

The no free lunch theorem for optimization [78] states that all optimization algorithms have the same performance, if averaged over all possible problems. This implies that a general-purpose, universal optimization strategy that outperforms competing strategies does not exist. As a consequence, the only way for one method to outperform another is its adaptation to the structure of the specific problem [38]. Clearly, experimenters and researchers should bear this idea in mind. The core issue thus lies in the a priori knowledge of the exact problem instance(s) to be solved and the lacking capability to generalize the results beyond the solved instances. Keeping that in mind, experiments should be based on test instances (randomly) drawn from specific problem classes, which is often difficult, because known test suites provide only a limited number of test functions. Additionally, many commonly used test suites may not provide any relation to real-world problems.

To overcome these two major issues, Bartz-Beielstein [4] introduced a methodology which is applicable to most real-world regression or optimization problems. The basic idea is to generate problem classes rather than completely different single instances and the a priori unknown (random) selection of a number of instances to retrieve generalizable results. The problem classes can be generated by taking data from real-world problems. Features of the data are extracted and learned, e.g., by modeling the data. Model parameters can be stochastically varied to retrieve diverse instances. This methodology, which generates problem classes rather than using one fixed instance, is constructed in a sophisticated way to extract relevant features from real-world problems. These features are parametrized, which allows for generation of infinite number of test problem instances. Mixed statistical models can be used for the analysis [51], because the instances have to be treated as random variables.

Fischbach et al. [26] applied this approach on real-world data taken from an industrial experiment. Here, a Kriging model is trained with the data and the model parameters are changed in a controlled manner. For instance, the nugget parameter, which enables noise handling, or parameters of the selected correlation function are

varied. This leads to different predictors that can be used as test instances. To avoid instances that have no relation to the real world data or instances that are too similar to each other, statistical measures of (dis-)similarity are applied. Instances that are outside of certain bounds with respect to some measure of similarity are removed from the test suite.

Zaefferer et al. [80] applied the above described procedure on a combinatorial real-world problem. They test different variants of efficient global optimization for combinatorial problems (CEGO) [81]. CEGO extends the application of surrogate models for optimization on combinatorial search spaces by employing a distance measure, e.g., Hamming distance or swap distance. The underlying real-world data set stems from the field of DNA sequences. The experimentally evaluated sequences are each assigned a fitness value based on the affinity to some target protein. This data set allows for the comparison of derived test instances utilizing a Kriging model predictor with the reference data and reveals a major disadvantage of the prediction, because the (sometimes extensive) smoothness of the prediction reduces the difficulty of the test instances by eliminating a large portion of the local optima. This is an inherent problem of test functions based on regression methods—the real-world data is nearly always subject to smoothing. Hence, Zaefferer et al. proposed a simulation approach to generate test instances avoiding the pitfall of smoothing. Gaussian process simulation can reflect the behavior or moments of the underlying real-world problem rather than just the data itself. Besides the potential to avoid smoothing, Gaussian process simulation may also provide an intuitive way to create varied test instances from the model, without the need for variating model parameters [81].

However, the simulation approach comes along with some practical issues. Firstly, a number of samples has to be selected as the basis of the simulation. To generate a test function, the simulated samples are the basis for an interpolation which may again introduce undesirable smoothing. An estimation of how many simulation samples are (at least) necessary to produce a simulation with similar characteristics as the real world problem is currently not available. Even if available, this number may grow large, especially for rather rugged fitness landscapes. Hence, the Kriging simulation may require the computation of large correlation matrices, according to the dimensionality of the problem instance and the number of points to simulate. This requires an efficient way to handle large correlation matrices within the Kriging model.

Summarizing the current state of research, the following questions remain to be answered:

- What similarity measures and bounds should be used to identify useful test instances?
- Can landscape features be used to categorize test instances into meaningful classes, such that results become more easy to interpret?
- If model-based test instances are generated, how to select a model that is able to reflect the behavior of the underlying problem?

- If a certain model is chosen, comparing search procedures that employ the same modeling method may have an advantage over competitors. How can this potential bias be addressed?
- In the case of simulation-based models, how can the appropriate number of simulated samples be computed?
- How can Gaussian processes be best simulated for large numbers of samples?
- To receive statistically robust benchmarking results, how many experiments are required? How many instances should be generated, and how many repeated algorithm runs performed?
- Benchmark data may not meet the assumption of standard parametric test methods. What performance measures and test methods should be used to analyze such data?

3 Constraints

Optimization is an essential part of any engineering, economic and social system. Such real world systems usually have a lot of constraints. In order to solve such problems, optimization algorithms need to be combined with constraint handling techniques. Within the recent past, several techniques have been developed. They can be classified as follows [17]:

- *Penalty functions* are terms, which are added to the objective function. They consist of a penalty parameter multiplied by a measurement of violation of the constraints (e.g distance to the feasible region or the amount of violated constraints):
  - *Death penalty* rejects infeasible solutions and generates a new solution as long as a feasible solution can be found. It is easy to use and no further calculations are necessary. The main problem is stagnation within small feasible regions. Furthermore, the user gets no information from the infeasible solutions.
  - *Static penalty* has a fixed penalty factor during the whole optimization process.
  - *Dynamic Penalty* increases the penalty factor during optimization.
  - *Adaptive penalty* techniques set the values of all parameters involved automatically using feedback from the search process without user intervention.

  The main advantage of the use of penalty functions is their simplicity; however, their main disadvantage are the penalty factors, which determine the severity of the punishment. They must be chosen by the user and their values are problem-dependent.

- *Repair algorithms* attempt to find a feasible solution by modifying the infeasible solutions using one of the following strategies:
  - The repair is made for evaluation purposes only.
  - The repaired solution replaces the original one. This choice can be made probabilistically.

- *Stochastic ranking* introduces a probability value P_f to compare infeasible solutions based on their objective function value. That is, given any pair of two adjacent solutions, they are compared according to their objective function values with probability 1 in case both are feasible; otherwise, this probability is P_f.
- *Multi-objective optimization*, where the objective and constraint functions are both minimized. Mezura-Montes and Coello Coello [50] propose a classification of these methods, based on the way they transform the nonlinear programming problem into a multi-objective optimization problem:
  - Approaches that transform the constrained problem into an unconstrained bi-objective optimization problem, where the first objective is the original objective function and the second objective is the sum of constraint violation.
  - Techniques that transform the constrained problem into an unconstrained multi-objective optimization problem, where the original objective function and each constraint of the original problem are treated as separate objectives.

  Solving multi-objective optimization problems can sometimes be more expensive and complicated than solving the conventional constrained problems. Therefore, it is often not efficient to use multi-objective optimization, especially when the number of constraint functions is relatively high.

The simplest strategy to integrate constraint handling into surrogate-assisted optimization is to reject infeasible solutions. This might not be the best choice. Alternatively, constraints might be replaced by models as well (especially if constraints are expensive to evaluate), or integrated into the algorithms. However, the decision for an optimal strategy is a difficult task. Conventional constraint-based solvers need a high number of function evaluations. If only the constraint is considered to be expensive, only this should be considered for surrogate-modeling. Handling constraints directly within the algorithm is a relatively new research area. Although several surrogate-assisted approaches have been proposed in the last few years in order to reduce the expensive function evaluations, there is no significant progress in this field. Literature on surrogate modeling for constrained optimization problems is rare. Employment of fast surrogate models to approximate the objective as well as all constraint functions is a known approach, but expensive to compute. Forrester et al. [27] discuss the benefits of different infill sampling criteria used in surrogate-model-based constrained global optimization. Here surrogate models are used to approximate both the objective and constraint functions with the assumption that these are computationally expensive to compute. Singh et al. [63] developed a multi-objective constrained optimization algorithm, which makes use of Kriging models in conjunction with multi-objective probability of improvement and probability of feasibility criteria to drive the sample selection process economically. Hussein et al. [40] present an approach that proposes a constraint handling method within a surrogate modeling approach. It can solve difficult constrained test problems from the multi-objective optimization literature.

4 Ensembles of Surrogates

In computer-aided sequential parameter optimization the use of surrogates is crucial and its result strongly depends on the choice of the right surrogate. For choosing an adequate surrogate, features of the objective function must be considered as well as the properties of the different available types of surrogates. Hence, we need a system that takes care of this task. Several different approaches have been proposed to facilitate or even accomplish this.

The *expected improvement* (EI) approach handles the initialization and refinement of a model, but not the selection of the model type. 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 [45]. But there is no proof that Kriging is the best choice.

Alternative surrogates, e.g., regression trees [13], support vector machines [11, 65, 73], or lasso [71] and ridge regression [36] may be better suited, depending on the problem. For example, Müller and Shoemaker [52] reported that Kriging models and combinations including Kriging performed in general worse than when radial basis function models were used. An *a priori* selection of the best suited model is often impossible in the framework treated in this chapter, because of the black-box nature of the underlying optimization problems. Regarding the surrogate choice, the user can decide whether to use (i) a single, global surrogate during the optimization, (ii) multi-fidelity surrogates, or (iii) multiple surrogates, $M_i, i = 1, 2, \ldots, p$. Ensembles provide a simple solution to the problem of selecting the most adequate model, because they comprehend and aggregate several surrogates. In contrast to multi-fidelity surrogates, ensembles do not necessarily combine surrogates of different fidelity. Classical ensemble methods rather combine data driven surrogates of similar or identical fidelity. One example are random forests, which represent an ensemble of simple tree-based surrogates.

Several model selection strategies can be implemented. Well-known strategies are:

- *Round robin* and *randomized choosing* are the most simplistic implementations of ensemble-based strategies. In the former approach, the surrogates are chosen in a circular order independent of their previously achieved success. In the latter approach, the surrogate is selected randomly from the list of available surrogates. The previous success of the surrogate is not a decision factor.
- *Greedy strategies* choose the surrogate that provided the best function value so far, while the SoftMax strategy uses a probability vector, where each element represents the probability for a corresponding surrogate to be chosen [66]. The probability vector is updated depending on the reward received for the chosen surrogates.
- *Bagging* [36] combines results from randomly generated training sets and can also be used in function approximation.
- *Boosting* [36] sequentially combines several weak learners to a strong one in a stochastic setting.

- *Weighted averaging* approaches do not choose a specific surrogate's result but rather combine it by averaging. Since bad models should not deteriorate the overall result, a weighting scheme is introduced. Every model's result for a single design point is weighted by its overall error, the sum over all models yields the final value assigned to the design point. A similar approach is *stacking*, where the weights are chosen by an additional training step.

To reduce the computation time, an ensemble of different, possibly local, models can be generated and evaluated in parallel. Each model may employ the same candidate solutions (from the population) and the corresponding observations from expensive function evaluations. Haftka et al. [34] present a review of surrogate methods that provide parallelization.

Since each of the p surrogates proposes one or even more different candidate solutions, a selection and/or a combination mechanism is required. Conventionally, surrogates are assessed and chosen according to their estimated true error [41, 61, 62]. Generally, attaining a surrogate that has minimal error is the desired feature. The *mean absolute error* (the L1 norm), the *mean square error* or its pendant, the *root mean square error*, are commonly used as performance metrics. Methods from statistics, statistical learning [36], and machine learning [53], such as the simple holdout approach, cross-validation, and the bootstrap are also important in this context.

Several selection and combination mechanisms for surrogates were developed in the last years. A simple approach determines the best surrogate, i.e., the surrogate with the smallest prediction error, and determines the next candidate solution based on that surrogate. Alternatively, candidate solutions from several surrogates can be combined. Zerpa et al. [82] use multiple surrogates and build an adaptive weighted average surrogate of the individual surrogates. Goel et al. [30] explore the possibility of using the best surrogate or a weighted average surrogate instead of one single surrogate. Model quality, i.e., the errors in surrogates, is used to determine the weights assigned to each surrogate. Sanchez et al. [58] present a weighted-sum approach for the selection of surrogate ensembles. The surrogates for the ensemble are chosen based on their performance and the weights are adaptive and inversely proportional to the local modeling errors. Tenne and Armfield [69] propose a surrogate-assisted memetic algorithm which generates accurate surrogates using multiple cross-validation tests.

Huang et al. [39] use several surrogates with different accuracies for a semiconductor manufacturing system. They propose an ordinal transformation to utilize the results from several less accurate, but computationally cheaper surrogates. The function values of all solution candidates are evaluated on every cheap surrogate and the individuals are ranked. The authors observe that despite the big bias in the results from the cheap surrogates, the relative order among solutions is actually quite accurate. This order can be used to accelerate the selection process in EAs significantly. To reduce variability and bias in the results from the cheap surrogates, the authors apply an optimal computing budget allocation scheme.

Multiple surrogates can also be used to partition the search space. The treed Gaussian process approach uses regression trees to partition the search space into separate regions and to fit local Gaussian process surrogates in each region [32]. Nelson et

al. [54] propose an algorithm which creates a tree-based partitioning of an aerodynamic design space and employs independent Kriging surfaces in each partition. Couckuyt et al. [20] propose to combine an *evolutionary model selection* (EMS) algorithm with the EI criterion in order to dynamically select the best performing surrogate type at each iteration of the EI algorithm.

Friese et al. [29] presented a method that is able to build an ensemble of heterogeneous surrogates using convex linear combinations of the predictions. The method uses cross-validation on the known set of data points to ensure that the fit of the ensemble is the best achievable. Due to the convex linear combination, it has been shown that the ensemble prediction for each point is at least as good as the prediction of the weakest single surrogate but might even exceed the performance of the strongest surrogate.

Bartz-Beielstein and Zaefferer [8] presented a survey of model-based methods and introduced a taxonomy, which is useful as a guideline for selecting adequate model-based optimization tools. Furthermore, a new approach for combining surrogate information via stacking was proposed.

Although several approaches exist, and many successful applications were mentioned before, important questions remain unsolved and are subject of current research. They can be listed as follows:

1. Evaluation
   - How to select the most adequate surrogate?
   - Which properties or features specify the most appropriate model?
   - Which is the best method to evaluate the performance of a surrogate with regard to its different applications?
2. Necessity
   - Do better surrogates always result in improved performance?
3. Integration
   - How can results from different surrogates be combined?

5 Discrete Problems

Most publications on the use of surrogates in optimization focus on problems where the search space is continuous, i.e., $\mathbb{R}^n$. There is no reason to assume that only continuous problems can be expensive-to-evaluate black boxes. Rather, discrete or combinatorial optimization problems may also be subject to these challenges. Examples for discrete search spaces are, e.g., string, graph, categorical integer, binary or any kind of mixed space. Due to the difficulty that such discrete data structures may impose, and the limited amount of existing research, discrete problems are a major challenge for surrogate-assisted optimization. Some of the few published applications

in the field of surrogate-assisted combinatorial optimization include examples from the engineering domain [2, 14, 55, 68, 74], bio-informatics [22, 57] and computer science [64, 67].

In an abstract and simplified way, surrogate-assisted optimization algorithms can be split into two interconnected modules: (i) a modeling technique that uses data to learn the problem structure, and (ii) an optimization algorithm that exploits the model to determine promising solutions. Clearly, there is a large variety of optimizers available for discrete and combinatorial search spaces, e.g., various meta-heuristics like Evolutionary Algorithms (EAs). Hence, the real challenge does seem to reside with the required modeling techniques.

A recent survey [8] classified potential modeling techniques for discrete problems into six categories (called strategies). The survey also provides an overview of works that apply these strategies. The six strategies are as follows:

- The *naive approach* ignores the discrete structure of the data. This is a viable method when the discrete problem itself is rather simple and vector-valued. For example, ordinal integer variables may form discrete search spaces, yet may often be modeled with standard machine learning approaches.
- *Custom-fit solutions* can be developed for specific problems. As some recent examples show [2, 74], surrogate models may be developed to fit specific applications. While they may not be easily transferred to other applications, their exploitation of problem knowledge renders them a potentially powerful choice.
- *Inherently discrete models*, e.g., based on regression trees, may help provide intuitive solutions to discrete problems. They may be problematic when discrete variables are mixed with continuous variables, or when more complex objects are the subject of the optimization algorithm.
- *Mapping approaches* try to derive or learn some mapping function, that transfers from the complex, discrete search space to a space that is more easily handled by classical machine learning techniques. Dummy variables or contrasts are simple examples of this approach, while more sophisticated methods such as auto-encoders have also been proposed in this context [31].
- A modeling algorithm can also *extract numeric features* of the optimized objects. Afterwards, these features may be used within a classical modeling framework. Examples can be found in the context of genetic programming [37].
- Finally, some of the most frequently applied methods in surrogate-assisted optimization include models that exploit *measures of similarity* or dissimilarity, e.g., Kriging and support vector machines. If continuous or Euclidean measures of (dis)similarity are replaced by their respective discrete equivalents, these models provide a natural approach towards modeling of discrete data [45, 81].

While not to be dismissed in general, the first three of these strategies are the most limited ones. Firstly, the naive approach will fail for any more complex problem, and it is plainly infeasible if the data structure is not vector-valued. Secondly, custom-fit methods will require significant problem knowledge, which may conflict with the black-box nature of many problems. Thirdly, inherently discrete models often are

limited to certain structures, and may become problematic, e.g., when facing mixed problems.

On the other hand, the latter three strategies are quite promising, mainly due to their greater versatility. Mapping approaches comprise very classical approaches such as dummy variable mappings and also state-of-the-art approaches such as auto-encoders. Feature extraction allows to easily exploit knowledge about the problem, data-structure or properties of candidate solutions. Finally, similarity-based models allow to intuitively transfer approaches like Efficient Global Optimization [45, 81] to the discrete domain.

Despite these promising developments, the discrete domain still poses serious challenges. Clearly, the discrete case mirrors the continuous case in the sense that all other challenges described in this chapter apply to both cases. Some of these issues are even more pressing in the discrete case—the problem of developing benchmarks that are relevant to real-world problems is one example. While benchmarking is also quite challenging in the continuous case, the emerging-field nature of discrete surrogate-assisted optimization is one reason for the lack of openly available test-cases.

Another issue that may be especially challenging in the discrete domain is dimensionality. As in the continuous case, discrete search spaces of high dimensionality are especially difficult to deal with. At the same time, some questions lack answers in the discrete domain: When does combinatorial data become too high-dimensional for a certain modeling technique? How can we achieve dimensionality reduction of complex, combinatorial structures such as graphs or trees?

With the introduced strategies in mind, another important question is how practitioners should choose the right strategy for a certain use case. Clearly, expert knowledge may help to resolve this in some cases, for example, when the approaches require some kind of problem specific prior knowledge, e.g., a suitable map from the discrete structures to a more manageable one, expressive/information-rich features, or an appropriate measure of similarity. Depending on what is available, a strategy can be chosen. If the available prior knowledge is insufficient to make a choice between potential modeling strategies, this issue can be framed in the context of ensemble methods, again mirroring the continuous case.

Finally, there is one particular challenge that arises in the context of combinatorial optimization with similarity-based models—definiteness. Models such as support vector machines or Gaussian process regression may require positive semi-definite kernels [79]. In the discrete case, we will often deal with kernels that are not proven to be definite. Furthermore, designing definite kernels may be equally infeasible. Adapting the modeling technique to drop the definiteness requirement may have severe drawbacks, such as an increased computational effort or a loss of accuracy. In principle, indefinite kernels could also arise in continuous optimization. Yet, this is much less an issue due to the availability of many well-established and powerful kernels in this field.

To summarize, issues like dimensionality, benchmarking, strategy selection and definiteness will be vital for the further progress in the field of discrete surrogate-assisted optimization. Importantly, more effort should be spent on the consolidation

of theory and practice. An increased number of studies on real-world applications would be very helpful to highlight strengths and weaknesses of the methods that have been proposed in this field.

6 Multi-objective Optimization

Highly demanding optimization problems often yield more than one objective. These multiple objectives are usually aggregated to a single objective by an aggregation function, e.g., a weighted sum. This way, a single objective function is received that ordinary (single-objective) optimization algorithms can deal with. An alternative approach is to consider all objectives in parallel which particularly makes sense if objectives are conflicting like quality and price in production or lift and drag in airfoil design. Of course, surrogates are also used for multiple objective optimization problems if such problems are highly demanding, e.g., with respect to computational resources.

In the case of multiple objectives, the mathematical formulation of an objective function reads like:

$$f : \mathbb{R}^n \to \mathbb{R}^m, \quad f(x) = (f_1(x), \ldots, f_m(x)), \quad x \in \mathbb{R}^n.$$

In such cases, the concept of Pareto dominance plays a major role when solutions $x, y \in \mathbb{R}^n$ are compared based on their objective function values $f(x), f(y) \in \mathbb{R}^m$. A solution x is said to dominate another solution y ($x <_p y$ in case of minimization), if and only if the following holds:

$$\begin{aligned} \forall i: &\quad f_i(x) \leq f_i(y) \qquad (i = 1, \ldots, m), \\ \exists j: &\quad f_j(x) < f_j(y) \qquad (j = 1, \ldots, m). \end{aligned}$$

Based on this definition of dominance, special sets in search as well as in objective space can be identified. The Pareto set is the set of all non-dominated solutions in the search space. The image of the Pareto set under f in the objective space is then called the Pareto front. For more details the interested reader is referred to [18, 19, 23].

Although quite a few algorithms for multi-objective surrogate-assisted optimization have already been proposed, the field lacks a common repository where approaches are collected, benchmarked, or compared. The reason for this is rather straightforward. The development of such algorithms is application-oriented and the algorithms have usually been implemented to solve a special industrial or real-word optimization task. The inventors stem from different research fields having different scientific backgrounds, speaking different scientific languages, and are often even not aware of other scientists trying to solve similar optimization tasks. Thus, algorithms

developed for similar tasks are rather different like ParEGO and RASM, which will both be introduced in short in the following.

ParEGO (Pareto Efficient Global Optimization) by Knowles [46] is a kind of an aggregation approach to directly apply the famous EGO algorithm (Efficient Global Optimization by Jones et al. [45]) to multi-objective problems. In this sense, it is completely straightforward. The different objective values of a solution are converted into a single one using an augmented Tchebycheff function. Here, different weight vectors, drawn randomly, are chosen in each iteration. This allows gradually approximating the entire Pareto front.

In contrast to the ParEGO approach, approaches like RASM (Rank-based Aggregated Surrogate Models [49]) exist. This is a mono-surrogate approach, i.e., only one surrogate model is learned during the execution of the algorithm. For other approaches, multiple surrogate models are learned for the different fitness function values. In RASM, the surrogate models the Pareto dominance relation by invoking a rank-SVM framework [44]. Generated offspring are thereby filtered in terms of the approximated Pareto dominance relation.

In contrast to the mono-surrogate approaches, Emmerich et al. [24] proposed a way of integrating surrogates into an evolutionary algorithm based on local surrogate models. These surrogate models based on Kriging are set up for each individual which is to be evaluated by the costly objective function. Such local surrogates base on a predefined number of near neighbors and are used for preselecting. The preselection then leads to a reduced set of offspring to be evaluated by the costly objective function.

The mentioned approaches show only a small fraction of possible methods and approaches already being used in surrogate-assisted multi-objective optimization. A short overview of existing libraries was set up during the Surrogate-Assisted Multi-Criteria Optimization (SAMCO) workshop at the Lorentz Center in Leiden, NL.[1] Although this might look like a good first result, there are a lot of open issues to be dealt with in the future.

As a first and rather important open issue, this simple collection is missing a list of algorithms contained in each library. Moreover, a detailed comparison of strengths and weaknesses on every list entry would be highly appreciated. In addition, a lot of promising research areas exist, e.g.:

- Exploration of multiple objectives with different response surfaces approaches, in particular focusing on specific requirements of set- and indicator-based optimization techniques.
- Integration and testing of new variants of models.
- Integration and testing of new infill criteria.
- Exploration of approaches beyond one model per objective function, in particular considering model dominance relations and/or model performance indicator landscapes.
- Integration of ensembles of surrogates.

[1] See https://www.lorentzcenter.nl/lc/web/2016/764/info.php3?wsid=764 for SAMCO's website and http://samco.gforge.inria.fr/doku.php for the list of libraries (both accessed on 30. 11. 2017).

Finally, benchmarking surrogate-assisted multi-objective optimization lacks rigorousness. This would be a highly needed first step to better understand the strengths and weaknesses of each approach and, thus, provide algorithm recommendations for users.

7 Further Topics

7.1 Dynamic Problems

A frequently encountered challenge in real-world optimization are dynamic problems. Here, the location of optima or Pareto fronts change over time. Surrogate models need to implement new adaptive strategies to identify and track these changes [42]. In this case, we need to rethink the classical use of a surrogate model for the global approximation of the fitness landscapes. For instance, a static global model, which is initially fit and sequentially updated, is not applicable as it models a steady, non-dynamic state. Sequential updates of a model have to consider the dynamic changes of the landscape over time. A possible approach is to change their purpose from the pure approximation of the fitness landscape towards a prediction model of future changes in the landscape. For instance, models from time series forecasting can support a direct optimization of the objective [25]. Another approach arises from the usage of dedicated sub-populations in evolutionary algorithms to track moving optima. These can be assisted by dynamically constructed local regression models to reduce computational effort in well-explored areas [10, 12, 72]. Further, surrogates can be utilized to detect drifts in dynamic environments and trigger new optimization runs [70].

7.2 Dimensionality

If the dimensionality of a problem is large, this has a significant impact on the computation time of surrogates. Dimensionality has two main repercussions for surrogate-assisted optimization. Firstly, the search on the surrogate model becomes costly, due to the large number of variables that need be handled by the optimization algorithm. Secondly, the surrogate model building procedure itself becomes difficult, because the dimensionality will increase the number of model parameters. To tackle these problems, different strategies have emerged [16, 60]. The first is *decomposition* of a high-dimensional problem in a set of smaller, independent sub-problems [1, 75]. The second is *variable screening* to identify the importance of variables by analyzing their main, interaction and higher order effects. This allows for a reduction of the search space by elimination of insignificant effects [33, 76]. *Mapping* techniques transform the variable space to a feature space of lower dimension, for instance by

means of a principal component analysis [3, 56]. The last common technique is *space reduction*, where the aim is to reduce the search space, e.g., by changing the boundary constraints of the problem [59].

7.3 Noisy Problems

In noisy problems, the fitness evaluations are subject to uncertainties and deviations. These are common in real-world situations, for instance as an effect of sensory measurement errors or random effects in computer simulations [9, 43]. Server availability may not be guaranteed, which results in different calculation times per job. An integration of resource handling in the optimization/simulation algorithm is needed.

Classical procedure include repetitions of design points and modeling by regression, such as linear least-squares models. Another example of such a method is the so-called *nugget effect* in Kriging. Without this effect, Kriging interpolates the data by reproducing observations, which is not applicable in noisy environments. With the nugget effect, it allows deviations from the interpolation by applying regression [21]. A current approach to deal with noisy problems in evolutionary and surrogate-assisted optimization is *sequential resampling* with *optimal computational budget allocation* [6, 7, 15].

8 Conclusion

Although surrogate-assisted optimization has been a hot topic for more than a decade, a lot of research directions still need to be investigated in more detail. This chapter tried to highlight several issues where the authors see a demand for further investigation. The chapter outlines these open issues and provides information on how these fields have already been addressed in the recent past. By that, we hope that users and developers can learn about recent scientific progress and get an idea where further research will be required in the future.

The most important of the presented topics turned out to be benchmarking. Benchmarking reoccurs in several sections of this chapter, e.g., in multi-objective optimization as well as in discrete optimization. Moreover, references and recommendations for people new to the field, who are trying to apply surrogate assisted algorithm to their problems, should be derived from benchmarking results. Thus, benchmarking is an open issue throughout the whole field of surrogate-assisted optimization and a mandatory step towards identifying strengths and weaknesses of approaches.

The first steps of choosing the right approach, algorithm or model should be made easier. The authors of surrogate-assisted optimization approaches originate from a lot of different fields. On the one hand, this is a problem since they have different backgrounds, speak different scientific languages and name things differently, prefer different solutions and so on. On the other hand, the diversity of the field enables

the potential of finding solutions inspired by a wide range of disciplines. Providing solutions that are relevant and applicable in many different fields is one of the greatest achievements one can reach in science.

Acknowledgements This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement no. 692286. Tea Tušar acknowledges financial support from the Slovenian Research Agency (project no. Z2–8177).

References

1. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data, pp. 94–105 (1998)
2. Asadi, M., Goldak, J.: Combinatorial optimization of weld sequence by using a surrogate model to mitigate a weld distortion. Int. J. Mech. Mater. Des. **7**(2), 123–139 (2011)
3. Bandler, J.W., Cheng, Q.S., Dakroury, S.A., Mohamed, A.S., Bakr, M.H., Madsen, K., Sondergaard, J.: Space mapping: the state of the art. IEEE Trans. Microw. Theory Tech. **52**(1), 337–361 (2004)
4. Bartz-Beielstein, T.: How to create generalizable results. In: Springer Handbook of Computational Intelligence, pp. 1127–1142. Springer, Berlin (2015)
5. Bartz-Beielstein, T.: Stacked generalization of surrogate models - a practical approach. Technical Report 5/2016, TH Köln (2016)
6. Bartz-Beielstein, T., Blum, D., Branke, J.: Particle swarm optimization and sequential sampling in noisy environments. In: Metaheuristics, pp. 261–273 (2007)
7. Bartz-Beielstein, T., Friese, M., Zaefferer, M., Naujoks, B., Flasch, O., Konen, W., Koch, P.: Noisy optimization with sequential parameter optimization and optimal computational budget allocation. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO **2011**, pp. 119–120 (2011)
8. Bartz-Beielstein, T., Zaefferer, M.: Model-based methods for continuous and discrete global optimization. Appl. Soft Comput. **55**, 154–167 (2017)
9. Beyer, H.G., Sendhoff, B.: Robust optimization-a comprehensive survey. Comput. Methods Appl. Mech. Eng. **196**(33), 3190–3218 (2007)
10. Bird, S., Li, X.: Improving local convergence in particle swarms by fitness approximation using regression. In: Computational Intelligence in Expensive Optimization Problems, pp. 265–293 (2010)
11. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT92, pp. 144–152 (1992)
12. Branke, J., Kaußler, T., Smidt, C., Schmeck, H.: A multi-population approach to dynamic optimization problems. In: Evolutionary Design and Manufacture, pp. 299–307. Springer, Berlin (2000)
13. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC Press (1984)
14. Brownlee, A.E., Wright, J.A.: Constrained, mixed-integer and multi-objective optimisation of building designs by NSGA-II with fitness approximation. Appl. Soft Comput. **33**, 114–126 (2015)
15. Chen, C.h., Lee, L.H.: Stochastic Simulation Optimization: An Optimal Computing Budget Allocation, vol. 1, World Scientific (2011)
16. Chen, X., Diez, M., Kandasamy, M., Zhang, Z., Campana, E.F., Stern, F.: High-fidelity global optimization of shape design by dimensionality reduction, metamodels and deterministic particle swarm. Eng. Optim. **47**(4), 473–494 (2015)

17. Coello Coello, C.A.: Constraint-handling techniques used with evolutionary algorithms. In: Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, pp. 675–701 (2017)
18. Coello Coello, C.A., Lamont, G.B.: Applications of Multi-Objective Evolutionary Algorithms. World Scientific, New Jersey (2004)
19. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn, Springer, Berlin (2007)
20. Couckuyt, I., De Turck, F., Dhaene, T., Gorissen, D.: Automatic surrogate model type selection during the optimization of expensive black-box problems. In: Proceedings of the 2011 Winter Simulation Conference, WSC 2011, pp. 4269–4279 (2011)
21. Cressie, N.: Spatial prediction and ordinary kriging. Math. Geol. **20**(4), 405–421 (1988)
22. Custódio, F.L., Barbosa, H.J., Dardenne, L.E.: Full-atom ab initio protein structure prediction with a genetic algorithm using a similarity-based surrogate model. Proceedings of the IEEE Congress on Evolutionary Computation, CEC **2010**, 1–8 (2010)
23. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley (2001)
24. Emmerich, M.T.M., Giannakoglou, K.C., Naujoks, B.: Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. IEEE Trans. Evol. Comput. **10**(4), 421–439 (2006)
25. Filipiak, P., Michalak, K., Lipinski, P.: Infeasibility driven evolutionary algorithm with arima-based prediction mechanism. Intell. Data Eng. Autom. Learn.-IDEAL **2011**, 345–352 (2011)
26. Fischbach, A., Zaefferer, M., Stork, J., Friese, M., Bartz-Beielstein, T.: From real world data to test functions. In: Proceedings of the 26. Workshop Computational Intelligence, pp. 159–177 (2016)
27. Forrester, A., Keane, A.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. **45**, 50–79 (2009)
28. Forrester, A., Sobester, A., Keane, A.: Engineering Design Via Surrogate Modelling: A Practical Guide. Wiley (2008)
29. Friese, M., Bartz-Beielstein, T., Emmerich, M.: Building ensembles of surrogates by optimal convex combination. Bioinspired Optim. Methods Their Appl. BIOMA **2016**, 131–143 (2016)
30. Goel, T., Haftka, R.T., Shyy, W., Queipo, N.V.: Ensemble of surrogates. Struct. Multidiscip. Optim. **33**(3), 199–216 (2006)
31. Gómez-Bombarelli, R., Duvenaud, D.K., Hernández-Lobato, J.M., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., Aspuru-Guzik, A.: Automatic chemical design using a data-driven continuous representation of molecules. arXiv:1610.02415 (2016)
32. Gramacy, R.B.: TGP: an R package for Bayesian nonstationary, semiparametric nonlinear regression and design by treed Gaussian process models. J. Stat. Softw. **19**(9), 1–46 (2007)
33. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
34. Haftka, R.T., Villanueva, D., Chaudhuri, A.: Parallel surrogate-assisted global optimization with expensive functions–a survey. Struct. Multidiscip. Optim. **54**(1), 3–13 (2016)
35. Hansen, N., Auger, A., Mersmann, O., Tušar, T., Brockhoff, D.: COCO: a platform for comparing continuous optimizers in a black-box setting. arXiv:1603.08785 (2016)
36. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn, Springer, Berlin (2009)
37. Hildebrandt, T., Branke, J.: On using surrogates with genetic programming. Evol. Comput. **23**(3), 343–367 (2015)
38. Ho, Y., Pepyne, D.: Simple explanation of the no-free-lunch theorem and its implications. J. Optim. Theory Appl. **115**(3), 549–570 (2002)
39. Huang, E., Xu, J., Zhang, S., Chen, C.H.: Multi-fidelity model integration for engineering design. Procedia Comput. Sci. **44**, 336–344 (2015)
40. Hussein, R., Deb, K.: A generative kriging surrogate model for constrained and unconstrained multi-objective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016, pp. 573–580 (2016)

41. Jin, R., Chen, W., Simpson, T.W.: Comparative studies of metamodelling techniques under multiple modelling criteria. Struct. Multidiscip. Optim. **23**(1), 1–13 (2001)
42. Jin, Y.: Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol. Comput. **1**(2), 61–70 (2011)
43. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments-a survey. IEEE Trans. Evol. Comput. **9**(3), 303–317 (2005)
44. Joachims, T.: A support vector method for multivariate performance measures. In: Proceedings of the Twenty-Second International Conference on Machine Learning, ICML 2005, pp. 377–384 (2005)
45. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Glob. Optim. **13**, 455–492 (1998)
46. Knowles, J.: Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans. Evol. Comput. **10**(1), 50–66 (2006)
47. Knowles, J., Nakayama, H.: Meta-modeling in multiobjective optimization. In: Multiobjective optimization, pp. 245–284. Springer, Berlin (2008)
48. Le, M.N., Ong, Y.S., Menzel, S., Jin, Y., Sendhoff, B.: Evolution by adapting surrogates. Evol. Comput. **21**(2), 313–340 (2013)
49. Loshchilov, I., Schoenauer, M., Sebag, M.: Dominance-based Pareto-surrogate for multi-objective optimization. In: Proceedings of the 8th International Conference on Simulated Evolution and Learning, SEAL 2010, LNCS, vol. 6457, pp. 230–239 (2010)
50. Mezura-Montes, E., Coello Coello, C.A.: A survey of constraint-handling techniques based on evolutionary multiobjective optimization (2006)
51. Montgomery, D.C.: Design and Analysis of Experiments, 5th edn, Wiley (2001)
52. Müller, J., Shoemaker, C.A.: Influence of ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization problems. J. Glob. Optim. **60**(2), 123–144 (2014)
53. Murphy, K.P.: Machine Learning: A Probabilistic Perspective. MIT press (2012)
54. Nelson, A., Alonso, J., Pulliam, T.: Multi-fidelity aerodynamic optimization using treed meta-models. In: Proceedings of the 25th AIAA Applied Aerodynamics Conference, pp. 1–19. American Institute of Aeronautics and Astronautics (2012)
55. Rao, S.V.N., Manju, S.: Optimal pumping locations of skimming wells. Hydrol. Sci. J. **52**(2), 352–361 (2007)
56. Robinson, T., Eldred, M., Willcox, K., Haimes, R.: Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. Aiaa J. **46**(11), 2814–2822 (2008)
57. Romero, P.A., Krause, A., Arnold, F.H.: Navigating the protein fitness landscape with Gaussian processes. Proc. Natl. Acad. Sci. **110**(3), E193–E201 (2013)
58. Sanchez, E., Pintos, S., Queipo, N.V.: Toward an optimal ensemble of kernel-based approximations with engineering applications. In: Proceedings of the 2006 IEEE International Joint Conference on Neural Network, pp. 2152–2158 (2006)
59. Shan, S., Wang, G.G.: Space exploration and global optimization for computationally intensive design problems: a rough set based approach. Struct. Multidiscip. Optim. **28**(6), 427–441 (2004)
60. Shan, S., Wang, G.G.: Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. Struct. Multidiscip. Optim. **41**(2), 219–241 (2010)
61. Shi, L., Rasheed, K.: A survey of fitness approximation methods applied in evolutionary algorithms. In: Computational Intelligence in Expensive Optimization Problems, pp. 3–28. Springer, Berlin (2010)
62. Simpson, T., Toropov, V., Balabanov, V., Viana, F.: Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come—or not. In: Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, pp. 1–22 (2012)
63. Singh, P., Couckuyt, I., Ferranti, F., Dhaene, T.: A constrained multi-objective surrogate-based optimization algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 3080–3087 (2014)

64. Smith, J., Stone, C., Serpell, M.: Exploiting diverse distance metrics for surrogate-based optimisation of ordering problems: A case study. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2016, pp. 701–708 (2016)
65. Smola, A., Vapnik, V.: Support vector regression machines. Adv. Neural Inf. Process. Syst. **9**, 155–161 (1997)
66. Sutton, R.S., Barto, A.G.: Introduction to Reinforcement Learning, 1st edn, MIT Press (1998)
67. Swersky, K., Duvenaud, D., Snoek, J., Hutter, F., Osborne, M.: Raiders of the lost architecture: kernels for Bayesian optimization in conditional parameter spaces. arXiv:1409.4011 (2014)
68. Teixeira, C., Covas, J.A., Stützle, T., Gaspar-Cunha, A.: Multi-objective ant colony optimization for the twin-screw configuration problem. Eng. Optim. **44**(3), 351–371 (2012)
69. Tenne, Y., Armfield, S.W.: A versatile surrogate-assisted memetic algorithm for optimization of computationally expensive functions and its engineering applications. In: Success in Evolutionary Computation, pp. 43–72. Springer Berlin Heidelberg (2008)
70. Theiler, J., Galdrikian, B., Longtin, A., Eubank, S., Farmer, J.D.: Using surrogate data to detect nonlinearity in time series. Technical Report, Los Alamos National Laboratory, NM (United States) (1991)
71. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Methodolgical) **58**(1), 267–288 (1996)
72. Ursem, R.K.: Multinational gas: Multimodal optimization techniques in dynamic environments. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2000, pp. 19–26 (2000)
73. Vapnik, V.N., Chervonenkis, A.J.: Theory of Pattern Recognition. Nauka (1974)
74. Voutchkov, I., Keane, A., Bhaskar, A., Olsen, T.M.: Weld sequence optimization: the use of surrogate models for solving sequential combinatorial problems. Comput. Methods Appl. Mech. Eng. **194**(30–33), 3535–3551 (2005)
75. Wall, M.E., Rechtsteiner, A., Rocha, L.M.: Singular value decomposition and principal component analysis. In: A Practical Approach To Microarray Data Analysis, pp. 91–109. Springer, Berlin (2003)
76. Wang, H.: Forward regression for ultra-high dimensional variable screening. J. Am. Stat. Assoc. **104**(488), 1512–1524 (2009)
77. Whitley, L.D., Mathias, K.E., Rana, S., Dzubera, J.: Building better test functions. In: Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 239–246 (1995)
78. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)
79. Zaefferer, M., Bartz-Beielstein, T.: Efficient global optimization with indefinite kernels. In: Proceedings of the 14th International Conference Parallel Problem Solving from Nature, PPSN XIV, pp. 69–79 (2016)
80. Zaefferer, M., Fischbach, A., Naujoks, B., Bartz-Beielstein, T.: Simulation-based test functions for optimization algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2017, pp. 905–912 (2017)
81. Zaefferer, M., Stork, J., Friese, M., Fischbach, A., Naujoks, B., Bartz-Beielstein, T.: Efficient global optimization for combinatorial problems. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2014, pp. 871–878 (2014)
82. Zerpa, L.E., Queipo, N.V., Pintos, S., Salager, J.L.: An optimization methodology of alkaline-surfactant-polymer flooding processes using field scale numerical simulation and multiple surrogates. J. Pet. Sci. Eng. **47**(3–4), 197–208 (2005)