Overview of Surrogate Modeling in Chemical Process Engineering

Kevin McBride[1,*] and Kai Sundmacher[1,2]

DOI: 10.1002/cite.201800091

The ability to accurately model and simulate chemical processes has been paramount to the growing success and efficiency in process design and operation. These improvements usually come with increasing complexity of the underlying models leading to substantial computational effort in their use. It may also occur that the structure of the model is sometimes unknown making optimization and study difficult. To circumvent these issues, mathematically simpler models, commonly known as surrogate models, have been designed and used to successfully replace these complex, underlying models with much success. This technique has seen increasing use within the chemical process engineering field and this article summarizes some popular surrogates and their recent use in this area.

Keywords: Process design, Process optimization, Simulations, Surrogate modeling

Received: June 26, 2018; *revised:* November 24, 2018; *accepted:* December 03, 2018

1 Introduction

In engineering practice, one must often computationally simulate a complex system. For chemical engineers, this may include modeling of chemical production processes, scheduling problems, or complex thermodynamics. These computer simulations help one to gather valuable information about the process and avoid impracticable or impossible experimental investigation. However, many of these simulations require substantial computational effort and time to evaluate or functions used in the simulator are unknown, black-box models adding to the difficulty of their analysis. Models may also be noisy or discontinuous preventing finite difference techniques from being applied. As a result, the computational demand of performing domain exploration, optimization, or sensitivity analysis becomes prohibitive due to the high number of function evaluations that are required. In some cases, a clear relationship between input variables and their responses may not be accessible to the designer [1].

A solution to this issue is the use of surrogate models (or meta-models, regression surfaces, emulators), which are mathematically simple models that map, or regress, the input-output relationships of a more complex, computationally demanding model. This is depicted in Fig. 1 for a basic, two-variable example. With their use, a more efficient evaluation of the underlying model is performed, providing a better understanding of the input-output relationships, enabling complex models from differing sources to be combined together, and increasing the speed of analyzing the design space and performing optimization. This has been especially popular in the design and analysis of computer experiments (DACE), where optimization through surrogates is key [2, 3]. This has led to many uses in the chemical engineering domain, i.e., for simplifying unit operation and process optimization, increasing the speed and ease of parameter fitting and analysis, and for identifying the feasible regions of operation for black-box process models. With a cursory scan of the literature, it would seem that other branches of the engineering sciences are much more active in surrogate model research. This is especially the case for aerospace engineering from which several detailed reviews on surrogate modeling and optimization originate [4 – 6]. Notwithstanding, several within the chemical and process engineering community are researching surrogate modeling and two important reviews were recently published by Bhosekar and Ierapetritou [7], who covered many aspects of surrogate modeling, optimization and sampling, and by Garud et al. [8], who provided extensive coverage on the design of computer experiments for surrogate model sampling.

This article presents a brief overview of surrogate modeling use in the chemical process engineering field. In the next section a short introduction to surrogate modeling design and several of the most commonly used surrogates in the chemical process engineering literature are given. Sect. 3 presents several examples of surrogate modeling followed by the conclusion.

[1]Kevin McBride, Prof. Dr.-Ing. Kai Sundmacher
mcbride@mpi-magdeburg.mpg.de
Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany.

[2]Prof. Dr.-Ing. Kai Sundmacher
Otto-von-Guericke University Magdeburg, Chair for Process Systems Engineering, Universitätsplatz 2, 39106 Magdeburg, Germany.

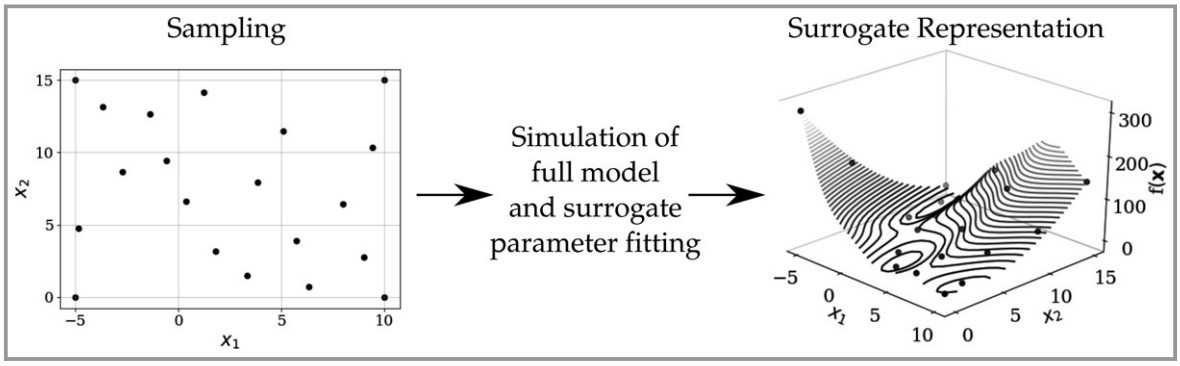

Figure 1. Surrogate model sampling, simulation, and fitting resulting in a surface representation of the original function. The sampling plan used here is a maxmin design with extreme vertices using a total of 20 samples sites (10 for each dimension).

2 Surrogate Modeling

2.1 Model Selection

One would like to replace a complex simulation with a surrogate and is faced with the question: which form should the surrogate take? There is no consensus or clear-cut guidelines on surrogate model selection in the literature, although some guidance is found in the works by Forrester [9] and Simpson [1]. It may not be known which surrogate will perform best for a given function and the decision usually requires some foundational knowledge of the underlying system [4], which may be difficult when handling black-box functions. One also needs to consider several factors such as dimensionality, simulation complexity, noisy responses, and mixed continuous and discrete data. Some model specific guidelines are given in Sect. 2.2.

Several authors attempted to address the lack of a comprehensive study on which surrogate performs best by direct comparison [10 – 12], but these are usually limited to only a couple of model types. More generally, Jones [13] stated that interpolating models are preferable to non-interpolating ones due to their better ability in matching the surface of the original function. This is preferable for deterministic computer experiments; however, non-interpolating methods are recommended for noisy simulations [14]. There is also a distinction made between global and local surrogates: global models represent the entire design space, i.e., feasibility analysis, where complete domain knowledge is desirable and for tasks involving optimization, localized models are more commonly used [15].

2.2 Design of Experiments for Surrogate Modeling

Since surrogate model quality depends strongly on the sample data used to map the input/output relationship of the underlying model, methods for identifying ideal points are necessary for building a good model. For DACE, several sampling techniques have been developed that differ from the classical design of experiments used in physical experiments. The random variation that exists in natural experiments is not a concern in deterministic computer simulations and replications are not necessary unless noise is present. The goal remains that one attempts to gather the maximum amount of information using the fewest number or limited set of simulation runs.

There are two types of sampling strategies relevant to surrogate design: the one-shot design (static, a priori, non-adaptive, domain based) and the sequential (adaptive, online, model-based) methods. A flowchart of both methods is shown in Fig. 2. The first of these selects sample sites from the complete design space without considering model knowledge using space-filling designs that attempt to uniformly spread the sample sites throughout the design space. The latter selects samples sites based on information provided by the surrogate model after an initial model has been developed. One may consider two extremes: regressing a surrogate model using only samples with a-priori space-filling designs or using the bare-minimum of samples to fit the initial model and improve it with targeted sample sites via

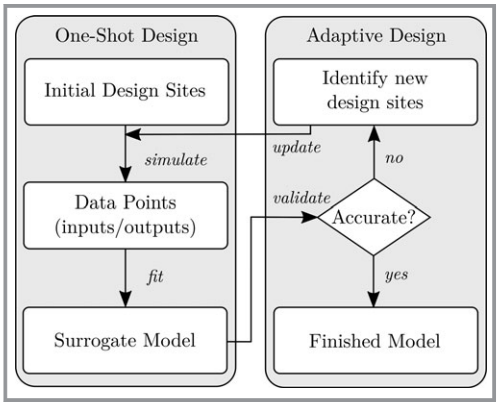

Figure 2. General procedure for the one-shot and adaptive methods for global surrogate model design.

adaptive sampling. Most applications use a compromise of both techniques. The reason is that for higher dimensional problems, the space-filling designs will become unreasonably large and computationally prohibitive or fail to sample adequately in the higher dimensional space [8]. On the other hand, excessive adaptive sampling leads to frequent refitting of the surrogate model parameters becoming itself computationally demanding [16]. Some researchers have explicitly investigated the performance of such trade-offs, such as by Wang and Ierapetritou [12], who show that performance is dependent upon the surrogate used.

Garud et al. [8] and Yondo et al. [6] compressively reviewed space-filling designs for computer experiments. The consensus is that simply increasing the number of samples does not necessarily lead to better surrogate design and that there is no universal rule for selecting a space-filling design. Many researchers have successfully used Latin hypercube sampling (LHS) [17] or a maxmin [18] design for this task. Some also include the extreme vertices of the design space to prevent extrapolation in these corners. Such a plan is shown in Fig. 1. The number of initial sample sites is often based on experience with the underlying model or by a recommendation from Jones et al. [19], who suggested using ten times the model dimension. In an attempt to add some order to sample plan selection, Garud et al. [8] presented one of the first comparisons of the space-filling capabilities of several popular static designs across various dimensions and sample sizes and made several recommendations for sample plan selection based on these criteria. Additionally, Bhosekar and Ierapetritou [7] compared the effects of initial sample size and sampling plan on performance for several kriging and RBF surrogates.

Use of adaptive sampling techniques tends to improve surrogate model performance compared to only using a single static plan while also requiring fewer functional evaluations [20]. Surrogates built adaptively also result in customized models and eliminate guesswork by the designer that may lead to unsatisfactory accuracy. Two general types of adaptive sampling techniques exist: one for improving the surrogate globally and one for improving the accuracy of localized regions of the domain for use in surrogate-based optimization [13]. When improving the surrogate model, one must consider the balance between addressing undersampled regions of the design space (exploration) and handling highly complex and non-linear areas (exploitation). Several global adaptive sampling techniques have been developed to address this, such as the modern LOLO-Voronoi by Crombecq et al. [20] and the Smart Sampling Algorithm (SSA) by Garud et al. [21]. A comprehensive list of specific methods plus their surrogate model dependencies is provided by Garud et al. [22]. Here, they also compared the performance of modern adaptive sampling techniques in developing surrogates to represent several chemical flow sheet simulations and show that SSA performed the best in almost all cases. Localized adaptive sampling differs in that it is accompanied by contractions and translations of the design around current optima where new surrogates are fit to sample sites in these localized regions around, a process that repeats until the specified stopping criteria are met. This is the fundamental use of surrogates for derivative free optimization (DFO) including trust-region methods [23]. A representative depiction of this procedure is given in Fig. 3.

In general, there is no rigorous, all-encompassing analysis of surrogate model selection, sampling strategy, and underlying model; however, several groups actively pursuing various pieces of this puzzle, e.g., Boukouvala et al. [10], Nuchitprasittichai and Cremaschi [24], Eason and Cremaschi [25], Sikorski et al. [11], Cozad et al. [26, 27], Wang and Ierapetritou [12], Bhosekar and Ierapetritou [7], Garud et al. [8, 22]. An overall discussion of current progress in these areas of surrogate modeling is presented by Bartz-Beielstein and Zaefferer [28].

2.3 Surrogate Models

This section presents the most commonly used surrogate models found in the chemical process engineering literature and short descriptions of polynomials, kriging, and artificial neural networks, radial basis functions (RBFs) are given. Other surrogate models that are used to a lesser degree in the chemical engineering literature not discussed here, but also of importance, are high dimensional model representation (HDMR) [29], support vector regression [30], and multivariate adaptive regression splines [31]. There are also many other models used in machine learning that may be of interest to the reader but are also beyond the scope of this article and a recommended source is Murphey [32].

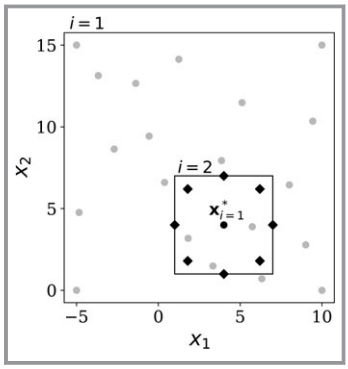

Figure 3. Hypothetical surrogate-based optimization approach. In the first iteration (i = 1), the global design space is sampled and mapped with a surrogate. This model is used in optimization and the local optimum is found at $x^*_{i=1}$. In the second iteration (i = 2), the design space is contracted and centered around the current optimum. New samples sites are identified using DoE techniques and a new, local surrogate model fit to the new data plus the current optimum.

2.3.1 Polynomials

Polynomial functions are the most commonly used surrogate model in engineering practice [5]. This class of regression model has an extensive history in classical experimental design where system information was often unknown, and it became necessary to develop methods that were able to explore the significance of the main process variables and their interactions [33]. Polynomials reveal this information conveniently through the magnitude of their coefficients and so provide general knowledge about the design problem and the underlying process. They are the computationally simplest models for regression purposes and should be used with less complex underlying models. They are usually restricted to main effects and first order interactions, such as in Eq. (1) since higher-order interactions usually lack significance and require more data to fit the additional parameters [1]. Polynomials work well for low-dimensional problems; however, high dimensional and highly non-linear systems are commonly encountered in engineering practice. In such a case, they may not represent the response surface reliably [13, 34] and are usually restricted to local regions. If a small data set is used, overfitting the polynomial becomes a concern and several regularization techniques exist to prevent this [35].

$$\hat{y}(x) = \beta_0 + \sum_{i=1}^{d} \beta_i x_i + \sum_{i=1}^{d} \beta_{ii} x_i^2 + \sum_{i=1}^{d} \sum_{j \le i}^{d} \beta_{ij} x_i x_j \tag{1}$$

2.3.2 Kriging

One of the most popular surrogate models is kriging, developed originally to describe spatial distributions between ore deposits based on the inverse distance weighting of known sample compositions [36]. Kriging was later introduced into deterministic computer experiments to describe the input and output relationships of computationally demanding simulations [2, 3, 19]. In the literature, kriging is usually the go-to wrench in the surrogate toolbox. Its popularity extends from its flexibility in being able to model many kinds of functions and interpolate the data while requiring only a handful of fitted parameters, which are due to its mathematical basis as a Gaussian process model [37]. The model consists of two parts: a deterministic polynomial term that describes the global trend of the data, and the realization of a stochastic process that accounts for the lack of fit in the polynomial term (Eq. (2)).

$$\hat{y}(\mathbf{x}) = p^{\mathrm{T}}(\mathbf{x})\beta + Z(\mathbf{x}) \tag{2}$$

$$cov\left(Z(x_i), Z(x_j)\right) = \sigma^2 R\left(x_i, x_j\right) \tag{3}$$

$$R\left(x_i, x_j\right) = \sigma^2 \exp\left(-\sum_{k=1}^{n} \theta_k x_i^k - x_j^{k2}\right) \tag{4}$$

$$\hat{y}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\hat{\beta} + r(\mathbf{x})^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y}_{\mathrm{D}} - \mathbf{F}\hat{\beta}\right) \tag{5}$$

$$s^2(\mathbf{x}) = \hat{\sigma}^2\left(1 - r(\mathbf{x})^{\mathrm{T}}\mathbf{R}^{-1}r(\mathbf{x}) + \frac{(\mathbf{F}\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) - \mathbf{p}(\mathbf{x}))^{\mathrm{T}}(\mathbf{F}\mathbf{R}^{-1}\mathbf{r}(\mathbf{x}) - \mathbf{p}(\mathbf{x}))}{\mathbf{F}^{\mathrm{T}}R^{-1}\mathbf{F}}\right) \tag{6}$$

The stochastic part requires selection of a correlation function (Eq. (3)) a-priori or by fitting a semivariogram to trends in the data [38]. The former usually take the form of a kernel function that performs a transformation on the Euclidean distance between two sample sites. The correlation tends towards unity for two nearby sites and reduces towards zero as the points move further apart. The Gaussian kernel (also known as the squared-exponential or radial basis function kernel) is very common in the chemical engineering literature due to its smoothness (Eq. (4)). A prediction for an unsampled point is a linear function of the observed data is found by Eq. (5) and its prediction error is calculated using Eq. (6). Regions in the sample space where uncertainty is high can be identified and new samples added to the observed data to increase the model's performance. Fig. 4 depicts this for a univariate function. This feature has led to the development of several kriging-specific adaptive sampling techniques aimed explicitly at this goal [13, 39, 40]. One must keep in mind that these predictions are made assuming that the model parameters fit from the observed data are correct; the predicted variance is itself only a prediction of the expected model uncertainty.

Kriging is recommended for problems with a dimensionality below 20, when the variables are continuous, and the underlying function is smooth. If there are discontinuities, this will lead to poor results due to the stationary covariance assumption of the correlation [41]. Fitting the kriging model (Eq. (5)) involves a matrix inversion and becomes

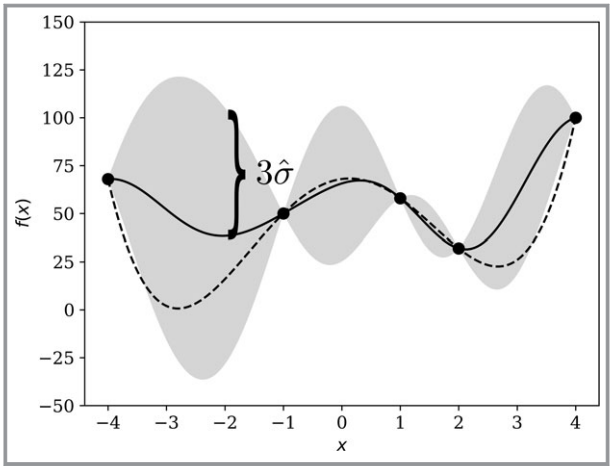

Figure 4. Kriging surrogate example showing the predicted model error (3σ) for the unsampled sites of a univariate function. The solid line is the surrogate model, the dashed line is the original model, and the dots represent the observed sites used in fitting the surrogate.

computationally demanding when the observed set is large. There are many different types of kriging than that presented here, and a list of most variants is found in Yondo et al. [6].

2.3.3 Artificial Neural Networks

Artificial neural networks (ANN) are based on the idea of how information is processed in actual biological neural networks, like the brain [42]. They have come to deviate substantially from their original inspiration, however, and are at their core a series of tensor operations performed on data, more like a geometric transformation. They are composed of units called artificial neurons, where the tensor transforms occur, that are arranged in a series of connected layers. Neurons receive numerical information from each neuron in the previous layer and output an analogous response that is provided as input to each neuron in the subsequent layer. Each of these transmissions is accompanied by a weight, which are fit to the training data using an optimization algorithm most often via backpropagation. In this manner the model learns to map the input data to the process responses. Fig. 5 depicts this idea for a small, simple network.

These surrogates are capable of being fit to many different systems and have provided some profound results for many different tasks [43]. They are able to adequately represent the global nature of the design space for high-dimension, nonlinear systems. Despite this, the downside to neural network modeling is in designing the appropriate network architecture, of which there are potentially infinite possibilities. This entails designing the network layout and defining its hyperparameters, such as learning rate, the transfer functions, regularization methods, etc., a process that may result in additional training and validation expense. Quite often a copious amount of data is needed to fit the usually large number of weights without overfitting. Thus, ANNs are recommended when significant volumes of data are present or cheap to generate (e.g., glass box models). This prohibits their use in computationally demanding simulations, where the many function calls would become impractical, unless small networks of only a few neurons are used. In addition, discovering correlations between the inputs and the responses is not quantifiable, reducing some potential model reduction possibilities. However, when these problems are insignificant or can be avoided, ANNs provide some of the most powerful surrogate models available.

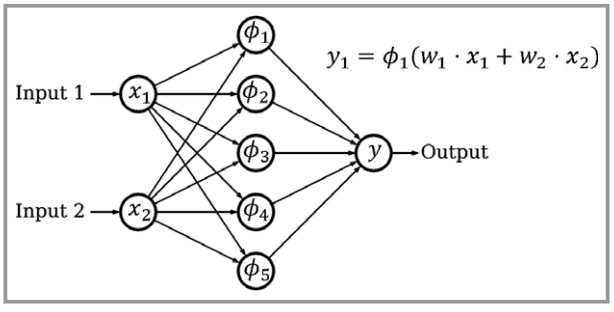

Figure 5. Basic structure of a neural network with two inputs, five neurons in the hidden layer, and a single output response.

2.3.4 Radial Basis Functions

RBFs consist of a weighted linear combination of local univariate functions performed on some measure of the distance from a point to a specified center [44]. The RBF approximation takes the form of Eq. (7), where x_i denotes the i^{th} center of n basis functions, $\varphi(x)$, which can take several forms, evaluated at the Euclidean distances between the prediction sites and the basis function centers, and λ_i are scalar weights fit during regression. This form of the RBF is identical to an ANN with a single hidden layer with radial basis functions [4]. Some authors may also include a first-order polynomial expansion to help find a unique, interpolating solution. Generally, RBFs are applicable to situations where kriging surrogates may be used, but are not as often used in the chemical engineering literature because the parameterized basis function of kriging (which may be considered a special form of RBF) is preferred due to its higher accuracy, flexibility, and ability to make predictions of model variance. Wang and Ierapetritou [12] recently addressed this issue by developing an adaptive sampling technique for cubic RBFs. In several cases, they showed that cubic RBFs improved flexibility in exploring the design space with higher accuracy while using fewer samples than kriging.

$$\hat{\mathbf{y}}(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i \, \phi(\, \mathbf{x} - \mathbf{x}_{i2} \,) \tag{7}$$

3 Applications of Surrogate Modeling in Chemical Process Engineering

In this section, different applications of surrogate modeling relevant to the chemical engineering field are presented. These include surrogate models used in optimization, dynamic process modeling, feasibility analysis, parameter estimation and sensitivity analysis, and scheduling.

3.1 Surrogate-Based Optimization

One of the most important uses of surrogate modeling has been in DFO [23, 45]. Optimization using derivative-based methods is more efficient than DFO and surrogates provide a surface with which one can derive derivative information [46]. This strategy is known by many names, including surrogate (or meta)-based optimization, DACE, and response-surface methodology.

Palmer and Realff [47, 48] were the first to optimize chemical flow sheets using surrogates. For this they used

polynomial and kriging surrogates due to their ability to handle limited data sets since they were determined to use as little sampling as possible. Optimization was performed using the initial models followed by a contraction or translation of the design space around the current optimum, as well as possible model reduction if input variables were found to be insignificant. With this method they achieved process configurations comparable to those found in the literature. Davis and Ierapetritou [49] introduced one of the first surrogate-based optimization approaches, Kriging-RSM, using a sequential design method to model noisy black-box functions with deterministic feasible regions. The algorithm begins with a kriging model to describe better the global domain and for performing the initial optimization. Once local optima have been identified, sub-regions using a geometrically adapted sampling plan around each local optimum are defined and new, local surrogate models are fit using second-order polynomials [50]. The best solution from the set of localized polynomials is taken as the overall optimum. The idea is to exploit the beneficial properties of each model type to reduce computational expense while ensuring better convergence to the global optimum. In a subsequent article, Davis and Ierapetritou [51] added a branch and bound method to the algorithm to handle integer variables allowing process synthesis and design problems to be considered. In a third iteration, Davis and Ierapetritou [52] incorporated discrete variables into the black-box models themselves. They successfully optimized a process for purifying alcohol dehydrogenase and one for the production of *tert*-butyl methacrylate, including their superstructures, using kriging models for the NLP subproblems.

However, in many of these examples the processes modeled had low dimensionality. It would, however, become impractical to model complete flow sheets as the dimensionality increases and a modular, or distributed, approach is often preferable. Caballero and Grossmann [53] used surrogate models to replace individual units within a process flow sheet model that were computationally expensive, difficult to obtain derivatives for, or noisy black-box models. This beneficially lowered the dimensionality of any given surrogate. They recommended a maximum dimensionality of nine or ten for each surrogate to keep sampling from becoming the computationally limiting step and to maintain surrogate accuracy. Their reasoning for using a kriging surrogate is that at each iteration, as opposed to a polynomial, the predicted variance of the model is maintained allowing for a convenient determination of stopping criteria and constraint feasibility. This allows for the noise generated by the simulation models to be taken into account and used as convenient stopping criteria for the optimization. These benefits come as a trade-off from the higher computational costs associated with fitting and updating the kriging models at each iteration, as compared to polynomials. Thus, they explicitly developed the algorithm to not update the kriging parameters in each iteration and attempted to generate satisfactory models initially that would require little adaptive improvement. They presented a series of optimal distillation column design examples where convergence with the original model proved difficult, but where their kriging-based method delivered reliable and robust solutions.

Quirante et al. [54] adapted this method to include discrete variables for distillation design and presented several optimization examples that included tray number, feed location, and in one case the superstructure of a blending process. Quirante and Caballero [55] then applied this method to an existing sour gas stripping plant in Germany to find the optimal operating conditions that would minimize operating costs. Here, they stressed not only the replacement of problematic units with surrogates as before, but that if necessary, explicit functions (especially the recycle streams) could be included as part of the NLP formulation. They were able to show that optimal operation of the plant could reduce operating costs by about 46 %. More recently, Quirante et al. [16] published an example of their method exemplified on the successful superstructure optimization of a vinyl chloride monomer production process. Lin et al. [56] applied a similar modular approach to compare two alternatives, reactive extraction and reactive adsorption, for the production of levulinic acid and hydromethylfurfural from glucose. In this example, kriging surrogates replaced both reactor configurations and all flash units found in both flow sheet variants.

Hasan et al. [57] compared the total annualized cost of vacuum swing and pressure swing adsorption in carbon capture by optimizing their respective superstructures. These models are very complex and difficult to solve because of the nonlinear algebraic and partial differential equation system used to model the non-isothermal adsorption. The solution was to use a global kriging surrogate in its place. Optimal points were simulated and added to the kriging surrogate until the cost improvement fell below a given tolerance. Hasan et al. [58] performed a similar study but incorporated the performance of the chosen zeolite into the design problem. Not only that, they also improved the screening step by performing many quick simulations to find the best set of initial design sites for full-scale simulations used to fit the kriging surrogate. First et al. [59] again applied this technique to optimize processes for the separation of CO_2 from methane while comparing the use of several different, novel zeolite materials.

Modular surrogate modeling has been used for thermodynamic equilibrium, which often leads to convergence issues during optimization when solved implicitly. To avoid calculating the liquid-liquid equilibrium during the optimization of a process for the hydroformylation of 1-dodecene, McBride et al. [60] replaced the decanter in the flow sheet with a kriging surrogate. Kaiser et al. [61] then used this same model for optimal reactor design in a similar process. Nentwich and Engell [62] also considered the optimization of a hydroformylation of 1-dodecene process but used ANN

surrogates to describe both the gas solubility in the reactor and the liquid-liquid equilibrium in a downstream decanter.

In most of the examples so far, the Gaussian process-based kriging method was implemented. However, like in the last example, ANNs have also been used extensively in the chemical engineering literature. Nascimento et al. [63] studied ANNs in the optimization of an industrial extruder for the polymerization of nylon and in the minimization by-products in an acetic anhydride plant. Meert and Rijckaert [64] have used them for polymerization modeling and Mujtaba et al. [65] applied an ANN in the optimal control of batch reactors. Bloch and Denoeux [66] used them for optimal control of coagulation and hot-dip galvanization and Fernandes [67] in the optimization of a Fischer-Tropsch reactor. Additionally, Henao and Maravelias [68] outlined a full featured methodology for a surrogate-based approach to superstructure optimization exemplified using ANNs to replace unit operations found in a process flowsheet. Fahmi and Cremeschi [69] optimized the superstructure of a biodiesel plant with surrogates replacing each unit operation, the thermodynamics, and mixing models. Smith et al. [70] replaced expensive CFD simulations in order to optimize a packed-bed reactor for algae growth. Nuchitprasittichai and Cremaschi [24] introduced a framework to determine the number of samples needed for fitting an ANN a-priori and exemplified this method with the optimization of a carbon capture process using aqueous amines. Eason and Cremaschi [25] continued in this vein by developing two adaptive sampling algorithms for ANNs, both of which were used to model the same carbon capture process. With their new methods, they reduced the amount of sampling required to produce accurate ANN surrogates.

The ANNs in several of these works are restricted to simple network architectures with a single hidden layer with only a handful of neurons. The benefit of such networks is that they are not difficult to fit and do not require large amounts of data. However, they lack the predictive power of larger networks. Naturally, larger networks require more data to prevent overfitting which may become counterproductive when limited computational expense is desirable. Also, several ANNs published in chemical engineering literature do not seem to have been rigorously regressed and show signs of being overfit. This strongly affects the predictive performance of such models that more attention should be applied to building high-quality ANNs when selected as a surrogate model. There is much to learn from the machine learning community on this topic [43].

One of the more interesting cases of DFO using surrogate modeling is the ALAMO framework [26]. This method designs mathematically simple surrogates from a set of basis functions using the least amount of data possible. Cozad et al. [27] added constrained regression to the method, which places bounds on the surrogate output, making extrapolation more reliable – an important feature for modeling physical or safety limitations in chemical processes. They exemplified ALAMO on several relevant examples, i.e., for a flash drum, a carbon capture adsorber, and a bound constrained batch reactor. It was also shown that the adaptive sampling method used by ALAMO results in similar accuracy as space-filling designs but requires fewer sampling points, reducing the number of complex simulations needed [71].

In a similar thread, Boukouvala and Floudas [72] developed ARGONAUT, a framework for optimizing constrained global DFO problems using surrogate models. Here, surrogates are automatically chosen from a list of possibilities (i.e., polynomials, RBF, kriging) based on need to limit complexity and maintain accuracy of the objective and constraints of the underlying gray-box models.

3.2 Surrogates for Dynamic Processes

Surrogates that can accurately represent dynamic systems may be a promising tool for online control. These dynamic surrogates are modeled as explicit functions of time with a discrete time step. The dynamic system of equations for the process is solved for this specified time interval at various initial conditions or parameters. The prediction of a current state now requires prior state information in addition to the current control variable inputs.

Hernandez and Gallivan [73] presented an early work using dynamic surrogates studying the use of kriging surrogates for modeling a second-order elementary reaction with a single reactant. Hernandez and Grover [74] later expanded this work to the practical problem of optimally producing platinum nanoparticles. Shokry and Espuña [75] considered the dynamic optimization of simple chemical process examples using kriging surrogates and found that selection of the proper time step is critical for accuracy. The larger the time step, the less sensitive is the model is to rapid changes in the system. The trade-off in increasing accuracy by reducing the time step is the increase in computational time and cost for initial sampling, fitting the surrogate, and performing the desired optimization.

A main area of focus for dynamic surrogate modeling has been found in the pharmaceutical process engineering, where kriging and polynomials had been previously investigated for steady-state operation of roller compaction by Boukouvala et al. [76] and feeder design by Jia et al. [77]. Using dynamic surrogates is of interest here because first-principle models of the process are often unknown and can be considered black-box models. Boukouvala et al. [10] explored how well kriging and ANN surrogates were in capturing the behavior of a dynamic roller compactor process. The kriging surrogates were able to recreate accurately the dynamic system behavior, even with large perturbations in the control variables. Due to the statistical nature of the kriging model, predictions made during operation could be incorporated into the observation database if within a specified prediction variance multiple, which increased the speed and accuracy of the dynamic kriging model. In this work,

the ANN surrogates were unable to capture the nature of the system leading to poor predictions. In addition, since ANNs are also not able to make model error predictions they were not able to add new points to the sample database. However, they stated clearly that only 20 neurons were used in a single hidden layer, which is quite restrictive, and that larger networks would make better predictions at the expense of higher sampling. Rogers and Ierapetritou [78] extended surrogate-based feasibility analysis to dynamic problems. They also noticed that time dependency becomes an issue with adaptive sampling, limiting its effective use because adding sample sites at a specific time is only possible after integrating from the initial time, making their identification difficult. By limiting the number of sample sites taken across the time horizon and by calculating the expected improvement for specific times, they were able to overcome these inconveniences and successfully model the dynamic feasible region of a roller compactor process with dynamic kriging models.

Sciascio and Amicarelli [79] estimated biomass concentration over time in a fermentation process using a Gaussian process. This proved to be more reliable in fitting the data than conventional methods based on extended Kalman filters. Likar and Kocijan [80] demonstrated the use of a Gaussian process for model predictive control in an actual lab-scale gas-liquid separation unit. They mention that a key handicap of Gaussian processes for dynamic systems is the computational burden of inverting the covariance matrix, which in this case consisted of up to one thousand samples. As with kriging, if too many samples are used, the model may become too computationally expensive itself. Despite this, they state that the controller is still robust due to the model's predicted variance, which – similar to the pharmaceutical examples in the next section – provides more operational safety.

3.3 Surrogates in Feasibility Analysis

Process feasibility describes the capability of the process to comply with all constraints, whether they relate to operation, product quality, production requirements, or some other performance aspect, when faced with uncertainty in the process parameters or variables. Feasibility analysis is interested in establishing the range in which process operation may occur without violation of these constraints, otherwise known as the feasible region, and can be described by the uncertain parameters. The roots of this problem go back to the flexibility index developed by Swaney and Grossmann [81] to numerically quantify the size of the parameter space in which feasible steady-state process operation is attainable using the control variables. A recent review on this topic is found in Grossmann et al. [82]. When considering complex black-box models, evaluating the feasible region is challenging and several researchers have addressed this by using surrogate models to map it.

Banerjee et al. [83] presented an early work in this area using HDMR to map feasible regions and then applying them to a materials design problem. Boukouvala and Ierapetritou [84] performed similar feasibility analysis using kriging surrogates with a special focus on initial sampling techniques. These initial surrogates were updated using a standard EI adaptive sampling technique and the authors were successful in defining the feasible regions for a roller compactor used in the production of solid dosage pharmaceuticals. They then adapted the EI method to account for variable noise [85] and later introduced an EI technique specific to feasibility analysis that required fewer sampling than previously to map adequately the feasible region [39]. Recently, Wang and Ierapetritou [86] developed two new adaptive sampling strategies for feasibility analysis in modeling stochastic black-box functions with stochastic kriging surrogates.

Rogers and Ierapetritou [78, 87] considered in a series of publications the feasibility analysis and flexibility analysis of black-box functions with an updated algorithm using an adjustable feasibility, which is the predicted feasibility less the square root of the prediction variance at some point in the design space. This is similar to the possible constraint feasibility seen in Caballero and Grossmann [53], but made the constraint boundary more conservative by creating a buffer between predicted feasibility and the real boundary. They argue that pharmaceutical products must adhere to strict standards, and that it is wise to be cautious around constraint boundaries.

Wang and Ierapetritou [12] introduced a novel method for surrogate-based feasibility analysis using cubic RBFs fit using a newly developed, model specific adaptive sampling technique. With this new method, they were able to conduct extensive comparisons of how sampling affects the performance of kriging and cubic RBF surrogates. In almost all of the cases they investigated, the cubic RBF method required far fewer sampling points than kriging for the same level of accuracy. They also report that cubic RBFs are less likely to become stuck exploiting regions and show some glaring examples where kriging is trapped when using a low-resolution grid sample plan. For higher dimensional examples, (one with five and one with six) both kriging and cubic RBF surrogates ran into the *curse of dimensionality* and the desired accuracy suffered. They concluded that adaptive sampling simply requires more iterations for complex surfaces in higher dimensions and recommend using LHS to sample for the initial model. This helps to find potential disjoint feasible regions than when only using the extreme vertices, but they found that this tends to weaken the conservativeness of the surrogate's predictions. The total number of sampling points is dependent on the original function and how many samples are required is still an open question.

3.4 Surrogates in Parameter Estimation and Sensitivity Analysis

When the original or first principles model is computationally demanding, parameter optimization may become cumbersome. To avoid the direct optimization of a complex

granulation population balance model, Braumann et al. [88, 89] performed parameter optimization using localized first and second-order polynomial surrogates of the full-scale model. Braumann et al. [90] updated their methodology by applying a more global approach to parameter estimation beginning with a quasi-random screening of the parameter domain to find the best set of parameters for the model by comparing their performance to experimental data. A second-order polynomial was then fit to the localized region around the current parameter set and the parameters of the granulation model were optimized using the projection method in Braumann et al. [89] and a Bayesian approach. In the latter, the Markov chain Monte Carlo (MCMC) sampling required to develop the posterior is achieved much more efficiently using the surrogates than if the full granulation model were used. Mosbach et al. [91] and Kastner et al. [92] also apply similar methods for reducing uncertainty in complex, computationally demanding models for an internal combustion engine and for jet-milling, respectively, by means of experimental design. Since parameter estimation may require many evaluations, second-order polynomials are used to represent the objective in the parameter space followed by Bayesian parameter estimation. Here, the surrogates reduced the computational burden of the MCMC sampling allowing for different MCMC algorithms to be compared [92]. To study the relative importance of model parameters in parameter optimization, Menz et al. [93] explored the use of an HDMR surrogate in the global sensitivity analysis of a population balance model for silicon nanoparticle synthesis. Sikorski et al. [11] presented a key paper addressing some open questions with surrogate modeling. Up until their work, neither a detailed assessment of how well surrogates describe industrial processes had been compiled, nor a comparison of surrogates for such tasks undertaken. Using the energy demand of a biodiesel plant as an example, they measured the performance of various surrogate models based on polynomials and HDMRs and the effects that dimensionality and domain size have on their accuracies. They showed that higher order polynomials are not necessarily better suited for higher dimensional data and that HDMR is not only effective as a surrogate, but also in identifying which variables are insignificant, leading to possible model reduction strategies.

3.5 Surrogates in Scheduling

Several scheduling and inventory management examples have also been simplified using surrogate models. Wan et al. [34] presented one of the first applications of kriging surrogate model use in supply chain management in order to reduce the computational burden compared to the modern methods at the time. Sahay and Ierapetritou [94] modeled a multi-enterprise supply chain network considering interactions between entities using kriging surrogates as part of a derivative free optimization approach. The expected improvement of the kriging method was used to move towards and ultimately identify the optimal configuration for the entire supply chain. Shi and You [95] used piece-wise linear surrogate models to represent the dynamic behavior of batch reactors in a scheduling problem reducing the original combined scheduling and dynamic optimization problem from an MIDO to an MINLP. The surrogates were iteratively updated during optimization to ensure they retained high fidelity without frequent sampling. This method proved to be much faster at converging to an optimal solution than by solving the full-scale model. Ye and You [96] investigated solving the optimization of an inventory system under uncertainty by modeling the performance of each inventory node as a kriging model in a localized manner. The multiple surrogates were aggregated together in order to optimize the entire network. The benefit of using surrogates in this way is analogous to the modular approach to flow sheet optimization in that each surrogate model's dimensionality can be reduced and accuracy maintained. Their region-wise surrogate modeling and optimization framework was able to identify better solutions in a shorter time than by using a conventional genetic algorithm in the two cases studies presented.

4 Conclusion

Surrogate modeling has many benefits including simplifying the object of study, reducing the computational burden for optimization, hastening parameter fitting and analysis, allowing for sensitivity analysis to be performed, modeling unknown black-box processes, and many more. Their use has helped in solving many difficult problems in the chemical and process engineering fields, several of which were mentioned in this article. It was shown that many of the open questions concerning surrogate model selection, sampling, and accuracy are being addressed by the chemical and process engineering community. In other branches, surrogate modeling is being taken to new levels by enhancing various models or combining them into ensemble methods [28]. All of these efforts combine to define a set of guidelines for surrogate model use and development, which and this will lead to a more systematic and structured approach to surrogate modeling. This will help lead to solutions that are more efficient, enabling more difficult processes to be modeled, optimized, and studied in the near future.

This work was conducted in part in cooperation with the Collaborative Research Centre "Integrated Chemical Processes in Liquid Multiphase Systems". The financial support from the German Research Foundation (DFG) under the grant SFB/TRR 63 is gratefully acknowledged.

Kevin McBride is a research associate in the group Process Systems Engineering at the Max Planck Institute for Dynamics of Complex Technical Systems, headed by Prof. Dr.-Ing. Kai Sundmacher. He studied Chemical Engineering at Texas Tech University before obtaining his diploma at Clausthal University of Technology. Dr. McBride received his PhD from the Otto-von-Guericke University Magdeburg. His research focuses on computer-aided molecular design of solvents and on modeling chemical production processes.

Kai Sundmacher studied Mechanical and Process Engineering at the University of Hanover and at the TU Clausthal (doctorate 1995, habilitation 1998). After a postdoctoral phase at the University of Newcastle, UK, he was appointed to the Chair of Process Systems Engineering at Otto-von-Guericke University Magdeburg in 1999. Since 2001 he has also been Director of the Max Planck Institute for Dynamics of Complex Technical Systems in Magdeburg. His research focuses on the modeling, analysis and synthesis of complex processes in chemical production, biotechnology, and chemical energy conversion.

Abbreviations

ALAMO	Automatic Learning of Algebraic Models for Optimization
ANN	artificial neural network
ARGONAUT	AlgoRithms for Global Optimization of coNstrAined gray-box compUTational problems
DACE	design and analysis of computer experiments
DFO	derivative-free optimization
EI	expected improvement
HDMR	high-dimensional model representation
LHS	Latin hypercube sampling
LLE	liquid-liquid equilibrium
MCMC	Markov chain Monte Carlo
MIDO	mixed-integer dynamic optimization
MINLP	mixed-integer nonlinear programming
NLP	nonlinear programming
RBF	radial basis function
RSM	residual surface map
SSA	smart sampling algorithm
SVR	support vector regression

References

[1] T. W. Simpson, F. Mistree, *AIAA J.* **2001**, *39 (12)*, 2233. DOI: https://doi.org/10.2514/2.1234

[2] J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn, *Stat. Sci.* **1989**, *4 (4)*, 409. DOI: https://doi.org/10.1214/ss/1177012413

[3] J. P. Kleijnen, *Design and Analysis of Simulation Experiments*, 2nd ed., Springer, Berlin **2015**.

[4] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P. Kevin Tucker, *Prog. Aerosp. Sci.* **2005**, *41 (1)*, 1. DOI: https://doi.org/10.1016/j.paerosci.2005.02.001

[5] A. I. J. Forrester, A. J. Keane, *Prog. Aerosp. Sci.* **2009**, *45 (1 – 3)*, 50. DOI: https://doi.org/10.1016/j.paerosci.2008.11.001

[6] R. Yondo, E. Andres, E. Valero, *Prog. Aerosp. Sci.* **2017**, *96*, 23. DOI: https://doi.org/10.1016/j.paerosci.2017.11.003

[7] A. Bhosekar, M. Ierapetritou, *Comput. Chem. Eng.* **2018**, *108*, 250. DOI: https://doi.org/10.1016/j.compchemeng.2017.09.017

[8] S. S. Garud, I. A. Karimi, M. Kraft, *Comput. Chem. Eng.* **2017**, *106*, 71. DOI: https://doi.org/10.1016/j.compchemeng.2017.05.010

[9] A. I. J. Forrester, A. Sobester, A. J. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley, Hoboken, NJ **2008**.

[10] F. Boukouvala, F. Muzzio, M. G. Ierapetritou, *Ind. Eng. Chem. Res.* **2011**, *50 (11)*, 6743. DOI: https://doi.org/10.1021/ie102305a

[11] J. J. Sikorski, G. Brownbridge, S. S. Garud, S. Mosbach, I. A. Karimi, M. Kraft, *Comput. Chem. Eng.* **2016**, *95*, 108. DOI: https://doi.org/10.1016/j.compchemeng.2016.06.019

[12] Z. Wang, M. Ierapetritou, *AIChE J.* **2017**, *63 (2)*, 532. DOI: https://doi.org/10.1002/aic.15362

[13] D. R. Jones, *J. Global Optim.* **2001**, *21 (4)*, 345. DOI: https://doi.org/10.1023/A:1012771025575

[14] A. I. J. Forrester, A. J. Keane, N. W. Bressloff, *AIAA J.* **2006**, *44 (10)*, 2331. DOI: https://doi.org/10.2514/1.20068

[15] K. Crombecq, *Ph.D. Thesis*, Ghent University, Belgium **2011**.

[16] N. Quirante, J. Javaloyes-Anton, J. A. Caballero, *Chem. Eng. Res. Des.* **2018**, *132*, 766. DOI: https://doi.org/10.1016/j.cherd.2018.02.032

[17] M. D. McKay, R. J. Beckman, W. J. Conover, *Technometrics* **1979**, *21 (2)*, 239. DOI: https://doi.org/10.1080/00401706.1979.10489755

[18] M. E. Johnson, L. M. Moore, D. Ylvisaker, *J. Stat. Plan. Inference* **1990**, *26 (2)*, 131. DOI: https://doi.org/10.1016/0378-3758(90)90122-B

[19] D. R. Jones, M. Schonlau, W. J. Welch, *J. Global Optim.* **1998**, *13 (4)*, 455. DOI: https://doi.org/10.1023/A:1008306431147

[20] K. Crombecq, D. Gorissen, D. Deschrijver, T. Dhaene, *SIAM J. Sci. Comput.* **2011**, *33 (4)*, 1948. DOI: https://doi.org/10.1137/090761811

[21] S. S. Garud, I. A. Karimi, M. Kraft, *Comput. Chem. Eng.* **2017**, *96*, 103. DOI: https://doi.org/10.1016/j.compchemeng.2016.10.006

[22] S. S. Garud, I. A. Karimi, G. Brownbridge, M. Kraft, *Comput. Chem. Eng.* **2018**, *108*, 276. DOI: https://doi.org/10.1016/j.compchemeng.2017.09.016

[23] A. R. Conn, K. Scheinberg, L. N. Vicente, *Introduction to Derivative Free Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, PA **2009**.

[24] A. Nuchitprasittichai, S. Cremaschi, *AIChE J.* **2013**, *59 (3)*, 805. DOI: https://doi.org/10.1002/aic.13871

[25] J. Eason, S. Cremaschi, *Comput. Chem. Eng.* **2014**, *68*, 220. DOI: https://doi.org/10.1016/j.compchemeng.2014.05.021

[26] A. Cozad, N. V. Sahinidis, D. C. Miller, *AIChE J.* **2014**, *60 (6)*, 2211. DOI: https://doi.org/10.1002/aic.14418

[27] A. Cozad, N. V. Sahinidis, D. C. Miller, *Comput. Chem. Eng.* **2015**, *73*, 116. DOI: https://doi.org/10.1016/j.compchemeng.2014.11.010

[28] T. Bartz-Beielstein, M. Zaefferer, *Appl. Soft Comput.* **2017**, *55*, 154. DOI: https://doi.org/10.1016/j.asoc.2017.01.039

[29] H. Rabitz, Ö. F. Aliş, *J. Math. Chem.* **1999**, *25 (2 – 3)*, 197. DOI: https://doi.org/10.1023/A:1019188517934

[30] A. J. Smola, B. Schölkopf, *Stat. Comput.* **2004**, *14 (3)*, 199. DOI: https://doi.org/10.1023/B:STCO.0000035301.49549.88

[31] J. H. Friedman, *Ann. Stat.* **1991**, *19 (1)*, 1. DOI: https://doi.org/10.1214/aos/1176347963

[32] K. P. Murphey, *Machine Learning A Probalistic Perspective*, The MIT Press, Cambridge, MA **2012**.

[33] G. E. Box, N. R. Draper, *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, New York **1987**.

[34] X. Wan, J. F. Pekny, G. V. Reklaitis, *Comput. Chem. Eng.* **2005**, *29*, 1317. DOI: https://doi.org/10.1016/j.compchemeng.2005.02.018

[35] D. Ruppert, M. P. Wand, R. J. Carroll, *Semiparametric Regression*, Cambridge University Press, Cambridge **2003**.

[36] D. G. Krige, *Master's Thesis*, University of Witwatersrand, South Africa **1951**.

[37] C. Rasmussen, C. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, MA **2006**.

[38] N. Cressie, *Statistics for Spatial Data*, John Wiley & Sons, Hoboken, NJ **2015**. DOI: https://doi.org/10.1002/9781119115151

[39] F. Boukouvala, M. G. Ierapetritou, *AIChE J.* **2014**, *60 (7)*, 2462. DOI: https://doi.org/10.1002/aic.14442

[40] Z. Wang, M. Ierapetritou, *Comput. Chem. Eng.* **2018**, *118*, 210. DOI: https://doi.org/10.1016/j.compchemeng.2018.07.016

[41] J. Stepenson, K. Gallagher, C. C. Holmes, *Geol. Soc. London Spec. Publ.* **2004**, *239*, 195. DOI: https://doi.org/10.1144/GSL.SP.239.01.13

[42] S. S. Haykin, *Neural Networks and Learning Machines*, Vol. 3, Pearson, Upper Saddle River, NJ **2009**.

[43] F. Chollet, *Deep Learning with Python*, Manning Publishing, Shelter Island, NY **2018**.

[44] H. Fang, M. F. Horstemeyer, *Eng. Optim.* **2006**, *38 (4)*, 407. DOI: https://doi.org/10.1080/03052150500422294

[45] S. Amaran, N. V. Sahinidis, B. Sharda, S. J. Bury, *Ann. Oper. Res.* **2016**, *240 (1)*, 351. DOI: https://doi.org/10.1007/s10479-015-2019-x

[46] O. Kramer, D. E. Ciaurri, S. Koziel, *Derivative-free optimization in Computational optimization, methods and algorithms*, 61, Springer, Berlin **2011**.

[47] K. Palmer, M. Realff, *Chem. Eng. Res. Des.* **2002**, *80 (7)*, 773. DOI: https://doi.org/10.1205/026387602320776849

[48] K. Palmer, M. Realff, *Chem. Eng. Res. Des.* **2002**, *80 (7)*, 760. DOI: https://doi.org/10.1205/026387602320776830

[49] E. Davis, M. Ierapetritou, *AIChE J.* **2007**, *53 (8)*, 2001. DOI: https://doi.org/10.1002/aic.11228

[50] E. Davis, M. Ierapetritou, *AIChE J.* **2010**, *56 (1)*, 220. DOI: https://doi.org/10.1002/aic.11881

[51] E. Davis, M. Ierapetritou, *J. Global Optim.* **2009**, *43 (2 – 3)*, 191. DOI: https://doi.org/10.1007/s10898-007-9217-2

[52] E. Davis, M. Ierapetritou, *Ind. Eng. Chem. Res.* **2008**, *47 (16)*, 6101. DOI: https://doi.org/10.1021/ie800028a

[53] J. A. Caballero, I. E. Grossmann, *AIChE J.* **2008**, *54 (10)*, 2633. DOI: https://doi.org/10.1002/aic.11579

[54] N. Quirante, J. Javaloyes, J. A. Caballero, *AIChE J.* **2015**, *61 (7)*, 2169. DOI: https://doi.org/10.1002/aic.14798

[55] N. Quirante, J. A. Caballero, *Comput. Chem. Eng.* **2016**, *92*, 143. DOI: https://doi.org/10.1016/j.compchemeng.2016.04.039

[56] Z. Lin, J. Wang, V. Nikolakis, M. Ierapetritou, *Comput. Chem. Eng.* **2017**, *102*, 258. DOI: https://doi.org/10.1016/j.compchemeng.2016.09.012

[57] M. M. F. Hasan, R. C. Baliban, J. A. Elia, C. A. Fluodas, *Ind. Eng. Chem. Res.* **2012**, *51*, 15665. DOI: https://doi.org/10.1021/ie301572n

[58] M. M. F. Hasan, E. L. First, C. A. Fluodas, *Phys. Chem. Chem. Phys.* **2013**, *15*, 17601. DOI: https://doi.org/10.1039/c3cp53627k

[59] E. L. First, M. M. F. Hasan, C. A. Fluodas, *AIChE J.* **2014**, *60 (5)*, 1767. DOI: https://doi.org/10.1002/aic.14441

[60] K. McBride, N. M. Kaiser, K. Sundmacher, *Comput. Chem. Eng.* **2017**, *105*, 212. DOI: https://doi.org/10.1016/j.compchemeng.2016.11.019

[61] N. M. Kaiser, M. Jokiel, K. McBride, R. J. Flassig, K. Sundmacher, *Ind. Eng. Chem. Res.* **2017**, *56 (40)*, 11507. DOI: https://doi.org/10.1021/acs.iecr.7b01939

[62] C. Nentwich, S. Engell, in *Proc. of Int. Joint Conference Neural Networks*, IEEE, Piscataway, NJ **2016**, 1291. DOI: https://doi.org/10.1109/ijcnn.2016.7727346

[63] C. A. O. Nascimento, R. Giudici, R. Guardani, *Comput. Chem. Eng.* **2000**, *24*, 2303. DOI: https://doi.org/10.1016/s0098-1354(00)00587-1

[64] K. Meert, M. Rijckaert, *Comput. Chem. Eng.* **1998**, *22*, 587. DOI: https://doi.org/10.1016/s0098-1354(98)00104-5

[65] I. M. Mujtaba, N. Aziz, M. Hussain, *Chem. Eng. Res. Des.* **2006**, *84 (8)*, 635. DOI: https://doi.org/10.1205/cherd.05096

[66] G. Bloch, T. Denoeux, *ISA Trans.* **2003**, *42 (1)*, 39. DOI: https://doi.org/10.1016/s0019-0578(07)60112-8

[67] F. A. Fernandes, *Chem. Eng. Technol.* **2006**, *29 (4)*, 449. DOI: https://doi.org/10.1002/ceat.200500310

[68] C. A. Henao, C. T. Maravelias, *AIChE J.* **2011**, *57 (5)*, 1216. DOI: https://doi.org/10.1002/aic.12341

[69] I. Fahmi, S. Cremeschi, *Comput. Chem. Eng.* **2012**, *46*, 106. DOI: https://doi.org/10.1016/j.compchemeng.2012.06.006

[70] J. D. Smith, A. A. Neto, S. Cremaschi, D. W. Crunkleton, *Ind. Chem. Eng. Res.* **2013**, *52*, 7181. DOI: https://doi.org/10.1021/ie30248d

[71] Z. T. Wilson, N. V. Sahinidis, *Comput. Chem. Eng.* **2017**, *106*, 785. DOI: https://doi.org/10.1016/j.compchemeng.2017.02.010

[72] F. Boukouvala, C. A. Floudas, *Optim. Lett.* **2017**, *11 (5)*, 895. DOI: https://doi.org/10.1007/s11590-016-1028-2

[73] A. F. Hernandez, M. G. Gallivan, in *2008 American Control Conference*, IEEE, Piscataway, NJ **2008**, 3993. DOI: https://doi.org/10.1109/acc.2008.4587117

[74] A. F. Hernandez, M. A. Grover, *Ind. Eng. Chem. Res.* **2011**, *50 (3)*, 1379. DOI: https://doi.org/10.1021/ie1007954

[75] A. Shokry, A. Espuña, *Proc. Technol.* **2014**, *15*, 376. DOI: https://doi.org/10.1016/j.protcy.2014.09.092

[76] F. Boukouvala, F. J. Muzzio, M. G. Ierapetritou, *AIChE J.* **2010**, *56 (11)*, 2860. DOI: https://doi.org/10.1002/aic.12203

[77] Z. Jia, E. Davis, F. J. Muzzio, M. G. Ierapetritou, *J. Pharm. Innov.* **2009**, *4 (4)*, 174. DOI: https://doi.org/10.1007/s12247-009-9070-6

[78] A. Rogers, M. Ierapetritou, *Chem. Eng. Sci.* **2015**, *137*, 986. DOI: https://doi.org/10.1016/j.ces.2015.06.014

[79] F. Sciascio, A. N. Amicarelli, *Comput. Chem. Eng.* **2008**, *32 (12)*, 3264. DOI: https://doi.org/10.1016/j.compchemeng.2008.05.015
[80] B. Likar, J. Kocijan, *Comput. Chem. Eng.* **2007**, *31 (3)*, 142. DOI: https://doi.org/10.1016/j.compchemeng.2006.05.011
[81] R. E. Swaney, I. E. Grossmann, *AIChE J.* **1985**, *31 (4)*, 621. DOI: https://doi.org/10.1002/aic.690310412
[82] I. E. Grossmann, B. A. Calfa, P. Garcia-Herreros, *Comput. Chem. Eng.* **2014**, *70*, 22. DOI: https://doi.org/10.1016/j.compchemeng.2013.12.013
[83] I. Banerjee, S. Pal, S. Maiti, *Comput. Chem. Eng.* **2010**, *34*, 1515. DOI: https://doi.org/10.1016/j.compchemeng.2010.02.016
[84] F. Boukouvala, M. G. Ierapetritou, *Comput. Chem. Eng.* **2012**, *36*, 358. DOI: https://doi.org/10.1016/j.compchemeng.2011.06.005
[85] F. Boukouvala, M. G. Ierapetritou, *J. Pharm. Innov.* **2013**, *8*, 131. DOI: https://doi.org/10.1007/s12247-013-9154-1
[86] Z. Wang, M. Ierapetritou, *J. Glob. Optim.* **2018**, *71*, 957. DOI: https://doi.org/10.1007/s10898-018-0615-4
[87] A. Rogers, M. Ierapetritou, *Chem. Eng. Sci.* **2015**, *137*, 1005. DOI: https://doi.org/10.1016/j.ces.2015.06.026
[88] A. Braumann, M. Kraft, P. R. Mort, *Powder Technol.* **2010**, *197*, 196. DOI: https://doi.org/10.1016/j.powtec.2009.09.014
[89] A. Braumann, P. L. W. Man, M. Kraft, *Ind. Eng. Chem. Res.* **2010**, *49*, 428. DOI: https://doi.org/10.1021/ie901230u
[90] A. Braumann, P. L. W. Man, M. Kraft, *AIChE J.* **2011**, *57 (11)*, 3105. DOI: https://doi.org/10.1002/aic.12526
[91] S. Mosbach, A. Braumann, P. L. W. Man, C. A. Kastner, G. P. E. Brownbridge, M. Kraft, *Combust. Flame* **2012**, *159*, 1303. DOI: https://doi.org/10.1016/j.combustflame.2011.10.019
[92] C. A. Kastner, A. Braumann, P. L. W. Man, S. Mosbach, G. P. E. Brownbridge, J. Akroyd, M. Kraft, C. Himawan, *Chem. Eng. Sci.* **2013**, *89*, 244. DOI: https://doi.org/10.1016/j.ces.2012.11.027
[93] W. J. Menz, G. P. E. Brownbridge, M. Kraft, *J. Aerosol Sci.* **2014**, *76*, 188. DOI: https://doi.org/10.1016/j.jaerosci.2014.06.011
[94] N. Sahay, M. Ierapetritou, *AIChE J.* **2016**, *62 (9)*, 3392. DOI: https://doi.org/10.1002/aic.15399
[95] H. Shi, F. You, *AIChE J.* **2015**, *61 (12)*, 4191. DOI: https://doi.org/10.1002/aic.14974
[96] W. Ye, F. You, *Comput. Chem. Eng.* **2016**, *87*, 164. DOI: https://doi.org/10.1016/j.compchemeng.2016.01.015

