J Glob Optim (2018) 71:845–869
https://doi.org/10.1007/s10898-018-0643-0

Optimization of black-box problems using Smolyak grids and polynomial approximations

Chris A. Kieslich[1] · **Fani Boukouvala**[2] · **Christodoulos A. Floudas**[3,4]

Received: 17 July 2017 / Accepted: 12 March 2018 / Published online: 3 May 2018

Abstract A surrogate-based optimization method is presented, which aims to locate the global optimum of box-constrained problems using input–output data. The method starts with a global search of the n-dimensional space, using a Smolyak (Sparse) grid which is constructed using Chebyshev extrema in the one-dimensional space. The collected samples are used to fit polynomial interpolants, which are used as surrogates towards the search for the global optimum. The proposed algorithm adaptively refines the grid by collecting new points in promising regions, and iteratively refines the search space around the incumbent sample until the search domain reaches a minimum hyper-volume and convergence has been attained. The algorithm is tested on a large set of benchmark problems with up to thirty dimensions and its performance is compared to a recent algorithm for global optimization of grey-box problems using quadratic, kriging and radial basis functions. It is shown that the proposed algorithm has a consistently reliable performance for the vast majority of test problems, and this is attributed to the use of Chebyshev-based Sparse Grids and polynomial interpolants, which have not gained significant attention in surrogate-based optimization thus far.

Keywords Surrogate-based optimization · Black-box optimization · Smolyak grids · Sparse-grids · Chebyshev · Polynomial interpolation

✉ Fani Boukouvala
fani.boukouvala@chbe.gatech.edu

1 Wallace H. Coulter Department of Biomedical Engineering, Georgia Institute of Technology, Atlanta, USA

2 School of Chemical and Biomolecular Engineering, Georgia Institute of Technology, Atlanta, USA

3 Texas A&M Energy Institute, Texas A&M University, College Station, USA

4 Department of Chemical Engineering, Texas A&M University, College Station, USA

1 Introduction

Optimization methods that are based on the exchange of designed data-streams between the solver and the problem formulation (Fig. 1), have gained significant interest recently. In the literature, one can find many contributions in the area of data-assisted optimization, such as direct-search [1], black-box optimization [2–5], derivative-free optimization [6–11], grey-box optimization [12], glass/black-box optimization [13], simulation optimization [14–18] and evolutionary or stochastic optimization [19–21]. All of the above terms are used for different classes of methods, however, they all have one thing in common: part or all of the optimization problem equations are not available in closed-form, but in the form of input–output data. Several reviews are available that outline the importance and applicability of such methods in the recent literature [1, 11, 15, 22]. Historically, data-based optimization became popular in the 1960s due to the poor computational capabilities for automatic differentiation using noisy computer codes prone to numerical errors [23, 24]. Nowadays, one of the main reasons that data-dependent optimization methods are becoming extremely valuable is on the other side of the spectrum. Specifically, our extraordinary computational capabilities now allow us to simulate complex systems with many internal degrees of freedom and increased computational cost, which cannot be directly optimized using gradient-based solvers. The number of applications of data-dependent optimization methods is already large, spanning a wide range of areas such as mechanical and structural design [25], complex flowsheet optimization and process synthesis [26, 27], geosciences [28], parameter estimation of large ODE systems [29], and many more.

Despite the aforementioned advances, several challenges still exist in terms of the computational tractability of the methods as the problem size increases (dimensionality and number of nonlinear terms), as well as the convergence properties of such methods to the global optimum. Well-established methods currently exist which guarantee convergence to local optima, if enough samples are collected to satisfy the necessary and sufficient conditions for local convergence [1, 8]. Moreover, several methods exist which guarantee asymptotic convergence to the global optimum, by ensuring that—at the limit of infinite sampling– the global optimum will eventually be found [30]. Due to the fact that the algebraic forms, and consequently the derivatives of the formulation are unknown, convergence to global optimality in a finite number of iterations or samples still remains a challenge. In addition, often there are sampling limitations due to the cost of the simulation or experiment, thus the value of theoretical guarantee to global optimality is overshadowed by the need to find the best possible solution with limited samples. As a result, black-box optimization methods that locate better solutions faster (with limited samples), but at the same time search the entire feasible space (global search) and provide some mechanism for converging to the global solution at any rate better than asymptotic will certainly make substantial impact on a number of applications.

A large fraction of the black-box optimization methods use the notion of surrogate functions, which are parametric models fitted to collected data from the simulation or experiment.

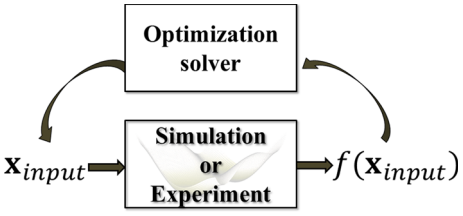

Fig. 1 Exchange of input–output data between simulation/experiment and optimization solver

The main hypothesis of surrogate-based optimization is that the optimum of a surrogate function is easy to obtain using a gradient-based solver, and this optimum is a good future sample for performing a full simulation or experiment. In other words, if the surrogate approximation captures any useful information about $f(x)$, its optimum is likely to be near the optimum of the unknown correlation $f\left(x_{glob}\right)$. Surrogate basis functions that have been explored range from simpler linear models to quadratic regression [31–33], to custom-built regression functions [16, 34], to interpolating basis functions such as kriging [4, 7, 12, 27, 35–37] and radial basis functions [5, 38–40], to artificial neural networks [41–43]. The above is not an exhaustive list of references, but this can be found in recent review articles [11, 14, 22, 44]. Typically, there are two critical decisions in order to implement the paradigm of Fig. 1, namely (a) where to sample ('design of experiment'), and (b) what type of surrogate function to fit using the obtained samples.

One potential approach for the development of surrogate models is using polynomial approximations and a set of discrete samples that lie on a Smolyak grid. Smolyak grids have not gained significant popularity in the area of black-box or derivative-free optimization thus far, and are also referred as sparse-grids or hyperbolic-cross points [45–56]. The theory was introduced in 1963, when Russian mathematician Smolyak derived a general methodology for creating sparse multidimensional grids as tensor products of one-dimensional sequences, in order to efficiently approximate and integrate functions on multidimensional hypercubes [56]. The Smolyak method has received significant attention in the integration literature [47, 48, 57–60], and approximation theory [46, 61, 62], where one can find elegant theoretical analysis on error bounds of integral approximations as a function of the number of samples and smoothness-related parameters [63].

Despite the fact that it is easy to make the connection between the similarities of the integral approximation, and derivative-free optimization using surrogate functions, the Smolyak approach has not received significant attention in the latter. There are a few notable contributions of incorporating Smolyak-grids within black-box optimization methods. Novak and Ritter [55] introduced a method which followed a direct-search approach using hyperbolic cross-points, which are a form of Smolyak grids. The work of Novak and Ritter was the first to introduce the concept of sparse-grid sampling for global optimization of complex functions, however, this work did not use any surrogate approximations directly. More recently, the search criterion of Novak and Ritter has been used with B-spline approximations for optimization of black-box functions [64]. Grimstad and Sandnes use sparse-grids and spline interpolants as constraints within a branch-and-bound algorithm for global optimization of a pump synthesis problem [65]. Finally, sparse-grid designs are used to fit local radial basis functions in a local optimization algorithm in Hulsmann and Reith [66]. Sparse-grid sampling has also been used for uncertainty quantification in stochastic optimization [59, 67, 68], however this area is outside the scope of this work.

The first aim of this contribution is to bring into the forefront some key concepts from the Smolyak theory literature to the global optimization community. Novak and Ritter did this very well in [55], however, the authors believe the recent increased interest in derivative-free optimization warrants revisiting the topic. To this end, we introduce a new algorithm that integrates some common techniques from the black-box optimization literature with a Smolyak-grid-based search algorithm. The proposed method targets problems of the form: $\left\{\min_{\mathbf{x}} f(\mathbf{x}); s.t.\ \mathbf{x}^{lo} \leq \mathbf{x} \leq \mathbf{x}^{up}; \boldsymbol{x} \in R^n\right\}$, where the closed-form of $f(x)$ is unknown. The objective of the method is to find the global optimum of $f(x)$ through sampling $f(x)$ using Smolyak grids and building Chebyshev polynomial functions $f_{sp}(x)$ as surrogate approximations of $f(x)$. Although the presented algorithm aims to find the global optimum, and

succeeds consistently for a large set of benchmark problems, theoretical convergence to the global optimum with a finite number of samples is not studied in this work.

The outline of this paper is as follows. First, basic concepts of sampling strategies and surrogate model fitting are introduced, followed by a discussion of Smolyak's method using Chebyshev polynomials. Then, a motivating example is presented, which shows the accuracy of Smolyak-based polynomial approximations when compared to surrogate functions that have been traditionally used in the black-box literature. Next, a new algorithm is described in detail, followed by the Results section, which presents the performance of the proposed method on a range of box constrained benchmark problems. The proposed algorithm is also compared to a recently developed algorithm for grey-box global optimization, ARGONAUT [12, 37].

2 Sampling and surrogate model fitting

In surrogate-based optimization, samples are typically costly to obtain, and therefore a full grid search is not possible in multidimensional spaces. In terms of the surrogate function selection, there are two competing objectives: (a) development of accurate predictive approximations, and (b) development of well-behaved, numerically stable and tractable surrogate functions, which can in turn be optimized using gradient-based optimization to identify new search directions. The first step within a surrogate-based optimization algorithm is the collection of the initial set of samples. Let set $i = 1, \ldots .N$ denote samples $x_{i,j}$ were $j = 1, \ldots, n$ represents the number of variables of the problem. Various sampling designs have been explored in surrogate-based optimization, ranging from uniform grid-based designs to space-filling designs to stochastic search. Uniform grid-based designs are deterministic; however, they suffer from the well-known curse of dimensionality. Space filling designs have gained significant popularity due to their properties of sampling uniformly within a hyper-cube, given a predefined number of samples N [69].

Latin hypercube (LH) sampling is the most commonly used design [70]. LH design routines first discretize each dimension in N levels, and then position the samples within the formed hypercubes such that each level is represented only by one sample in each dimension, and the points cover the space optimally (i.e., maximization of the minimum pair-wise distance criterion). When creating space-filling designs, there are multiple solutions to the optimality-based positioning problem and as a result, there are many equivalent LH designs with an equal number of samples. This characteristic creates a stochastic effect on the performance of any surrogate-based optimization method that uses a LH design as a starting point. Other examples of sampling designs which have been explored, are Delaunay triangulation centroid-based sampling [71], and orthogonal arrays [72], however, there is no universally optimal sampling strategy in the field of surrogate-based optimization. Sampling strategies can also be adaptive, when additional samples are added after performing iterations of surrogate model fitting and optimization based on various criteria. Typical criteria that have been used are minimization/maximization of the surrogate function or minimization of the error of prediction between the surrogate function and the unknown correlation [16, 41, 73], or hybrid criteria that aim to balance between exploration and optimization [4]. However, we consider this additional iterative sampling as part of an optimization or approximation methodology, which will be discussed later.

Once a good initial sampling set has been collected, the next step is to identify and train a surrogate function $f_{sp}(x)$. The overall goal of surrogate-based optimization is to find

$f_{sp}(x_{sp,glob})$, such that $\left|f(x_{glob}) - f_{sp}(x_{sp,\ glob})\right| = 0$, where $f_{sp}(x_{sp,glob})$ is the global optimum of the surrogate problem, while $f(x_{glob})$ is the true global optimum. One way to ensure that the global optimum will be found by a surrogate is to perform high accuracy model approximation—such that $\left|f(x) - f_{sp}(x)\right|_{\infty} \leq \varepsilon$, where ε is a very small number. However, this procedure can be very computationally expensive, and perhaps not necessary if the goal is not approximation but the location of a single point, namely x_{glob}. For the above reason, most methods adaptively refine their sampling sets and their surrogate functions in order to guide the search towards the optimum.

The types of surrogate functions that have been predominantly used can be categorized as regression or interpolation techniques. Regression models have a pre-postulated functional form and the fitted model is not required to interpolate all points exactly. Alternatively, interpolation functions are built to predict the training points exactly and do not require the pre-postulation of a functional form. Typical interpolation techniques that have been used for surrogate-based optimization are neural networks, radial basis functions and kriging functions. Regression models tend to be less complex in form and therefore regression surrogates are smoother functions, which can be optimized faster using traditional solvers. However, regression models are more rigid and if an inadequate basis function is pre-postulated, the fit is bound to have a large approximation error. Interpolation models are more flexible, however, their size and complexity increases as the number of samples increases. In other words, the closed-form equation that is developed after fitting an interpolating function may contain many nonconvex terms, making the optimization of any formulation containing such functions a limiting step of any algorithm that employs them.

The selection of the sampling design is closely connected to the surrogate function selection and both affect the overall final accuracy of the surrogate function and the overall optimization algorithm. Sampling sets must have a reasonable number of points and must cover the search space. The required number of samples increases with dimensionality, however, a full factorial search with multiple level values leads to intractable sampling designs (Table 1). A well-balanced design is a pre-requisite for the latter stage of parameter estimation for any surrogate function. However, both the selection of the appropriate surrogate type and the parameter estimation itself contribute towards the quality and accuracy of $f_{sp}(x)$. A common misconception is that more samples will inevitably lead to better approximations. However, this has been proven incorrect for the case of equidistant uniform grid designs and polynomial approximations by Runge in 1901 [74, 75]. Runge proved that for a special class of functions, when using equidistant grid-points for polynomial interpolation, the error of approximation increased as more samples were added to the interpolation set. In fact, the error showed an oscillatory behavior and thus fitting higher order polynomials lead to lower quality predictions. In terms of approximation error bounds for space-filling design sampling and interpolation with kriging and radial basis functions, there is no theoretical proof that more samples lead to a minimax approximation, to the best of the authors' knowledge. In fact, it will be shown in the next section through a motivating example that increasing the size of a Latin-Hypercube set and fitting interpolation or regression functions does not necessarily lead to better approximations.

2.1 Constructing Smolyak grids using Chebyshev polynomials

The sampling design in this work is based on Smolyak's theory for creating tensor products of unidimensional sampling rules [56]. One advantage of Smolyak grids (SG) is that they do not suffer from the curse-of-dimensionality as severely as uniform grids (Table 1). The second reason for exploring this sampling strategy, is the rich theory that is available for the

Table 1 Number of points in full grid versus Smolyak grid of levels 1-4

n (dimension)	Full tensor product grid with 4 levels (4^n)	Smolyak grid			
		Level $\mu=1$	Level $\mu=2$	Level $\mu=3$	Level $\mu=4$
2	16	5	13	29	65
5	1024	11	61	241	801
10	1,048,576	21	221	1581	8801
20	1,099,511,627,776	41	841	11,561	120,401

convergence of the interpolating functions fitted using these grids to the true $f(x)$, with a more tractable rate and a weaker dependence on dimensionality [62, 63, 76, 77]. In order to create a Smolyak grid, one has to select the unidimensional sampling design and then apply Smolyak's rule to transform it into a multidimensional grid. Another decision in creating a Smolyak grid is the level (μ), which controls the size of the n-dimensional grid and is associated with the level of approximation accuracy that is required. The selection of the unidimensional grid is linked to the surrogate function that interpolates the sampled data, since these grid points are roots or extrema of orthogonal polynomials [78]. Moreover, by selecting a non-equidistant 1D grid such as the Chebyshev-based grid, Runge's phenomenon is no longer observed. Once the 1D basis/points and the level of approximation is selected, the multidimensional grid can be constructed as a tensor product which satisfies a constraint that filters out a subset of the full tensor product, based on their significance. Finally, in order to fit the multidimensional polynomial based on the grid points, Lagrange interpolation is used to identify the unique parameters of the surrogate function.

Before any more details are provided, it is important to make clear that the algorithm proposed in this work is not just based on constructing full Smolyak grids and polynomial interpolants in a one-stage non-adaptive approach. However, a full Smolyak grid of a specified μ_{start} level is used as a starting sampling design in the proposed algorithm, which is followed by adaptive sampling and bound refinement within promising regions. Details about this implementation can be found in the Algorithm section. The proposed algorithm's performance heavily relies on the initial sampling and the properties of the initial interpolating polynomial, similar to the extent that many other surrogate-based optimization methods rely on the initial LH design and the selected surrogate function. In addition, even despite its adaptive nature, the algorithm is restrained to use points that belong only in Smolyak grids of various levels and bounded search spaces. For this reason, in this section we provide some details about how the full-grids are constructed, as well as some discussion on the convergence properties of these polynomial functions to the true $f(x)$. Since our algorithm relies on sampling from this fixed set of grid points, the supporting theory for Chebyshev-based SG sampling is important for understanding the improved properties and performance of the developed method. Adaptive SG sampling has been previously studied for integration and approximation [47, 50, 52], verifying that one can still observe good convergence to low approximation errors by selectively sampling subsets of SGs.

Smolyak's original contribution is based on the concept that some elements of a full tensor product grid are less important than others in approximating multidimensional functions [45, 56]. Extrema or roots of different types of polynomials have been used as grid points in a single dimension. In the proposed algorithm, extrema of Chebyshev polynomials are used in the $[-1, 1]$ domain, which is also commonly referred to as Clenshaw–Curtis interpolation.

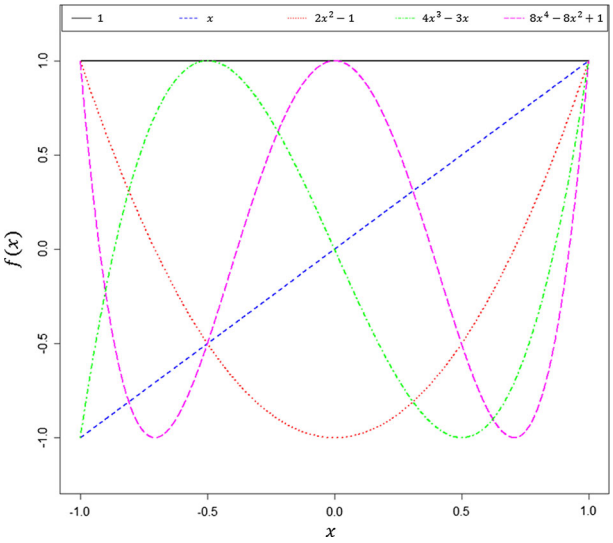

Fig. 2 Chebyshev polynomials of degree zero to four in the $[-1, 1]$ domain

Chebyshev polynomial basis functions are based on the recursive formula $T_0(x) = 1$, $T_1(x) = x$, $T_m(x) = 2xT_{m-1}(x) - T_{m-2}(x)$ for $m \geq 2$. When the degree of the Chebyshev polynomial is $m - 1$, there are m extrema which are given by $\zeta_l^m = -\cos\left(\frac{\pi(l-1)}{m-1}\right)$, for $l = 1, \ldots, m$, while for $l = 0$. Chebyshev polynomials of zeroth to fourth degree are plotted in Fig. 2, where one can clearly observe the nested nature of certain degrees of Chebyshev extrema. Specifically, the extrema of degree two are $[-1, 0, 1]$ and this set includes all of the extrema of the previous two levels. The same is true for degree four which is set: $\left[-1, -\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}, 1\right]$ and it includes all of the extrema of the previous levels. This nesting property allows the multidimensional grids that are based on these basis functions to be adaptively augmented, since every level contains all of the points of the previous level which have already been sampled.

Any multidimensional Smolyak grid of a specific level μ, is a tensor product of unidimensional grid points. These sets S_k must be nested and their cardinality is $card(S_k) = 2^{k-1} + 1$ for $k \geq 2$ and $card(S_{k=1}) = 1$. For example, for $k = 2$, the number of grid points in the unidimensional set are equal to $2^1 + 1 = 3$ and this is $[-1, 0, 1]$, while for $k = 3$, there are $2^2 + 1 = 5$ grid points which correspond to $\left[-1, -\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}, 1\right]$. To construct the multidimensional grid, the overall level of approximation μ must be set and this dictates the tensor product combinations which will be used for the grid. Specifically, the combinations of points that are kept in a SG satisfy the following constraint:

$$n \leq |k| \leq n + \mu \text{where } |k| = \sum_{i=1}^{n} k_i \tag{1}$$

As an example, if $n = 2$ and level $\mu = 1$ based on Eq. 1: $2 \leq k_1 + k_2 \leq 3$, which corresponds to the following combinations: (a) $k_1 = 1$, $k_2 = 1$, (b) $k_1 = 1$, $k_2 = 2$, (c) $k_1 = 2$, $k_2 = 1$. The above combinations lead to a grid with the following points in the $[-1, 1]^2$ hypercube: $x_1 = (0, 0)$, $x_2 = (0, 1)$, $x_3 = (0, -1)$, $x_4 = (1, 0)$, $x_5 = (-1, 0)$.

Table 2 Tensor product of grid points for a two variable problem using extrema of Chebyshev polynomials in the unidimensional space

	$S_{k1}\|S_{k2}$	$k_2 = 1$ [0]	$k_2 = 2$ [−1, 0, 1]
$k_1 = 1$	[0]	(0, 0)	(0, −1), (0, 0), (0, 1)
$k_1 = 2$	[−1, 0, 1]	(−1, 0), (0, 0), (1, 0)	(−1, −1), (−1, 0), (−1, 1) (0, −1), (0, 0), (0, 1)

The tensor product operations which lead to the above grid can be better shown through Table 2.

As seen in Table 2, the grid for $n = 2$ and level $\mu = 1$ is the non-redundant union of the sets (0, −1), (0, 0), (0, 1) and (−1, 0), (0, 0), (1, 0). A very recent efficient implementation of Smolyak grid construction overcomes the issue of redundancies in the tensor products allowing the direct identification of the non-redundant union of grid points [79]. Each of the grid points of the SG correspond to a term of the Chebyshev basis function. In the example of $n = 2$ and level $\mu = 1$, point (0, 0) is associated with a constant term (1), point (−1, 0) corresponds to basis function x_1, point (1, 0) corresponds to basis function $2x_1^2 - 1$, point (0, −1) corresponds to basis x_2 and point (0, 1) corresponds to point $2x_2^2 - 1$. As the level of approximation μ increases, terms of higher order and multiple variable terms start to appear (i.e., bilinear, trilinear, etc.).

The equation used to form the polynomial function, which will interpolate the Smolyak grid points is given by:

$$f_{sp}^{\mu}(x_1, x_2, \ldots, x_n) = \sum_{\max(n,\mu+1) \le |k| \le d+\mu} (-1)^{d+\mu-|k|} \binom{d-1}{d+\mu-|k|} p^{|k|}(x_1, x_2, \ldots, x_n) \tag{2}$$

where:

$$p^{|k|}(x_1, x_2, \ldots, x_n) = \sum_{l_1}^{card(S_{k1})} \cdots \sum_{l_n}^{card(S_{kn})} b_{l_1 \ldots l_n} \psi_{l1}(x_1) \cdots \psi_{ln}(x_n) \tag{3}$$

and $\psi_{lj}(x_j)$ is the l_j-unidimensional Chebyshev basis function in dimension j; $\psi_{l1}(x_1) \ldots \psi_{ln}(x_n)$ is the n-dimensional basis function and $b_{l_1 \ldots l_n}$ the corresponding coefficients. In this work, we have implemented an algorithm based on the method described in [79], in order to construct the grid and corresponding polynomial functions very efficiently.

The surrogate functions created by this approach have the following form, which is a compact form of Eq. 3:

$$f_{sp}^{\mu}(x_1, x_2, \ldots, x_n) = \sum_{l=1}^{M} b_l \Psi_l(x_1, x_2, \ldots, x_n) \tag{4}$$

where there is a total of M composite polynomial terms and M total points in the grid. After the multidimensional grid has been created and the basis functions of polynomial $f_{sp}(x)$ have been identified, the actual values of the black-box function $f(x)$ must be obtained at all grid points. Finally, based on the implementation of Judd et al. [79], the parameters of $f_{sp}(x)$ are calculated by solving a system of linear equations using Lagrange interpolation.

2.2 Error bounds of Smolyak grids using Chebyshev polynomials

Overall, polynomials have not been popular in the surrogate-based optimization literature, partly due to several misconceptions that are highlighted by Trefethen in [80]. For example, not all polynomial interpolants diverge as $N \rightarrow \infty$, especially when the points are not equidistant. In fact, the choice of Chebyshev interpolants will always converge to the true function $f(x)$, as long as the true function is relatively smooth. If a function is Lipschitz continuous, $\{|f(x) - f(y)| \leq L|x - y|\}$ for a constant L, this convergence is guaranteed. It is true that in most cases of black-box optimization, no information is available regarding the smoothness or derivatives of the unknown function. However, as very well described in Novak and Ritter [55], our aim should be to develop algorithms that perform as well as possible on the average case. It is also important to develop algorithms that have a consistent performance and that are less dependent on the random set of initial samples, which is an undesired characteristic of most surrogate-based optimization approaches.

One very important finding of SG based interpolation with N points, is that the error bound of $\varepsilon_{\mathrm{N}} = \left|f\left(x_{glob}\right) - f_{sp}\left(x_{sp,\ glob}\right)\right|$ has a weaker dependence on dimensionality n rather than the bound for uniform grid search, or stochastic search [55, 63, 77]. If any information regarding the bounds on the derivatives of the function are available or can be assumed, then tighter bounds on these errors can be derived [55]. In simpler terms, these polynomial approximations have good predictive abilities for a wide range of functions with some relative smoothness. Most importantly, as points are added to the grid (or terms are added to the interpolating function) there is supporting theory in the literature that shows that the unique polynomial representations will continuously improve in error and converge to the true function. Of course, this methodology is still affected by the curse of dimensionality, which is why adaptive methods are sought to selectively pick the best subsets of a Smolyak Grid as the training points of the interpolant [47, 49, 50, 52, 79].

3 Motivating example

The purpose of this section is to show the approximating properties and complexity of SG-based polynomial functions, when compared to traditionally used kriging, radial basis functions and quadratic models. In order to compare the approximating efficiencies and convergence properties of different grids and surrogate functions, the two-dimensional Branin function is used (Eq. 5):

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

$$\text{where } -5 \leq x_1 \leq 10,\ 0 \leq x_2 \leq 15.$$

The true contour and three-dimensional surface of Eq. 5 are shown in Fig. 3. This function is selected as a motivating example, due to its dimensionality and the presence of the trigonometric term, which is not included in any of the surrogates that will be fitted.

First, three levels of Smolyak grids are used (Fig. 4), to fit polynomial functions based on Chebyshev basis functions (Fig. 5). The full Smolyak Grids have 13, 29 and 65 points respectively. For comparison, Latin Hypercube designs with the same number of samples are collected and these points are used to fit quadratic, kriging and radial basis functions. The three sizes of LH designs cover a range that is comparable to standard practice in surrogate-based optimization of starting with $\sim 10n$ points. The LH designs are shown in Fig. 6, while

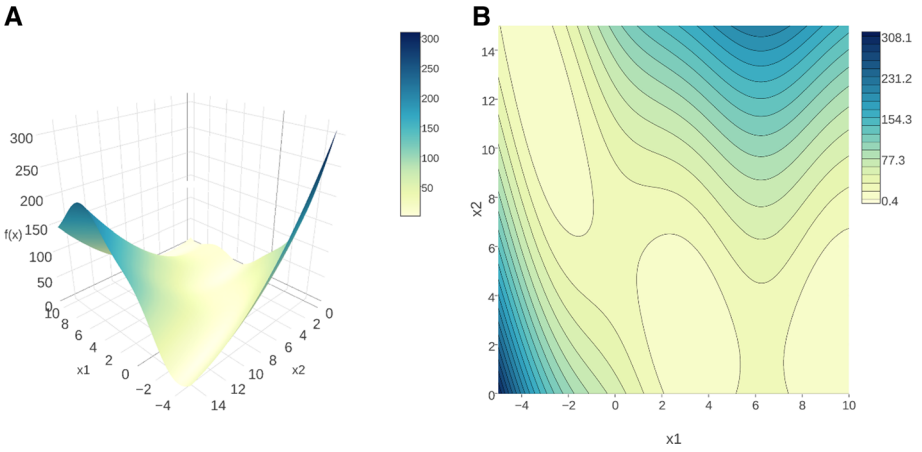

Fig. 3 True Branin function. **a** Three dimensional plot, **b** two dimensional contour plot

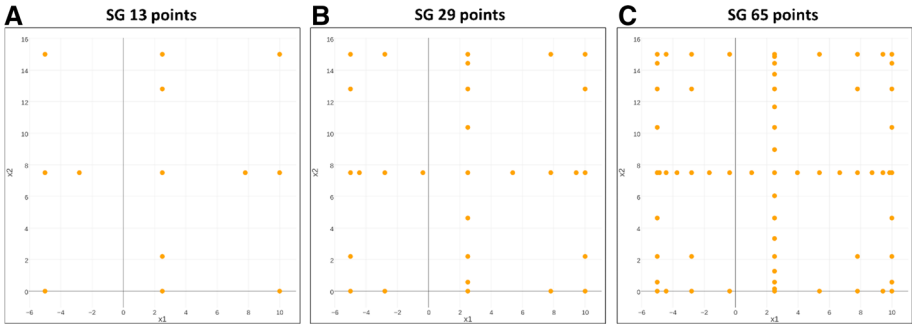

Fig. 4 Full sparse grid points using Chebyshev basis functions **a** level $\mu = 2$ with $N = 13$ points, **b** level $\mu = 3$ with $N = 29$, **c** level $\mu = 4$ with $N = 65$ points

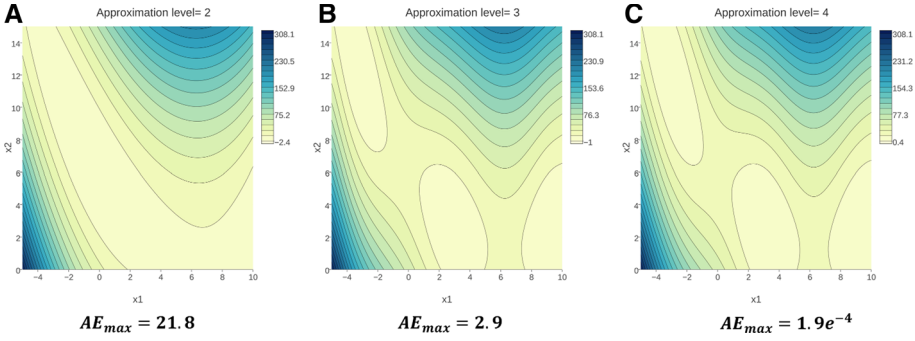

Fig. 5 Approximations of Branin function using full SG for **a** level $\mu = 2$ with $N = 13$ points, **b** level $\mu = 3$ with $N = 29$ points, **c** level $\mu = 4$ with $N = 65$ points

Figs. 7 through 9 show the corresponding fitted surrogates. For all of the fitted functions, the maximum Absolute Error: $AE_{max} = \max_i \left(\left| f_{sp}\left(\mathbf{x}^{(i)}\right) - f\left(\mathbf{x}^{(i)}\right) \right| \right)$ of approximation is calculated and reported over a dense grid of points (300×300 points).

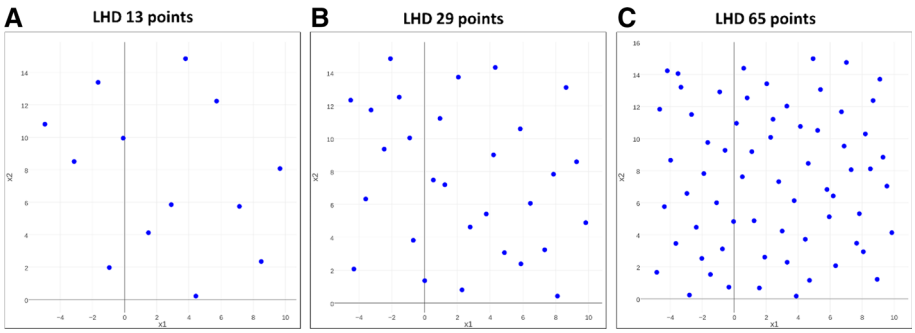

Fig. 6 Latin hypercube designs with **a** 13 points, **b** 29 points, **c** 65 points

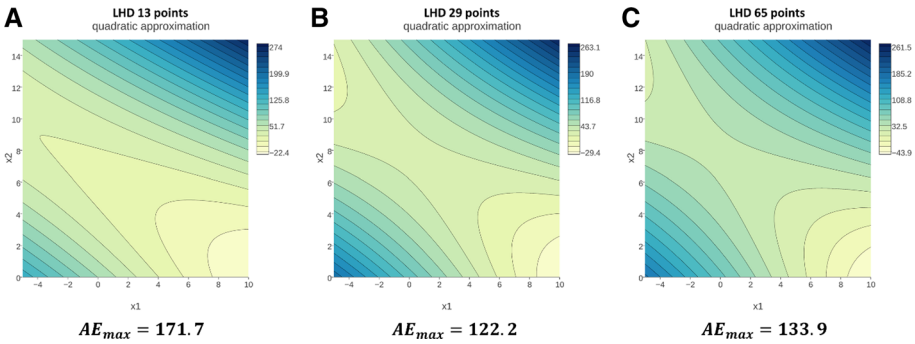

Fig. 7 Quadratic approximations of the Branin function based on the designs of Fig. 5. **a** LH design with 13 points, **b** LH design with 29 points, **c** LH design with 65 points

Even though no optimization is performed in this section, it is undeniable that the performance and efficiency of any surrogate-based optimization algorithm depends on the accuracy and complexity of the surrogate function, thus we believe that this comparison is very relevant. All of the parameter fitting is performed on scaled input–output data in the $[-1, 1]$ domain, in order to be able to assess the importance of the fitted parameters. The simplest form of all surrogate functions is compared, and this is achieved by removing any of the terms for which the parameters were found to have an absolute value lower than $1e-10$. All of the figures and absolute errors are reported based on actual values of the $f(x)$.

Based on Fig. 5, the Chebyshev based Smolyak-Grid approach (SG-Cheb) approximates the Branin function with a maximum absolute error of 21.8 when using 13 points, while for the next level with 29 points the maximum error decreases to 2.9. By the fourth level the function is almost perfectly represented with a maximum approximation error in the order of $1e^{-4}$ with 65 points collected. Even though this is a single-stage sampling/fitting procedure, what is important to observe is that these polynomial functions converge to the true function.

Based on Fig. 7, it can be observed that LH sampling followed by fitting of a quadratic regression model (LH-quad), performs worst in terms of maximum approximation error. This is expected, since the quadratic model does not contain the necessary terms to capture the nonlinearity of the Branin function and this will not improve if we add more points.

The fitted kriging (LH-krig) function is a good approximation of the Branin function, however, the approximation with 65 points has a maximum absolute error of 27.6 from the actual, which is higher than that with 13 points using the SG-Cheb approach. Moreover, in

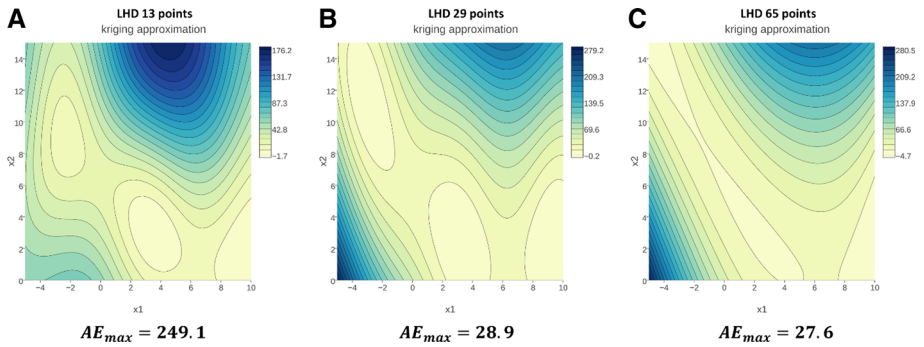

Fig. 8 Kriging approximations of the Branin function based on the designs of Fig. 5. **a** LH design with 13 points, **b** LH design with 29 points, **c** LH design with 65 points

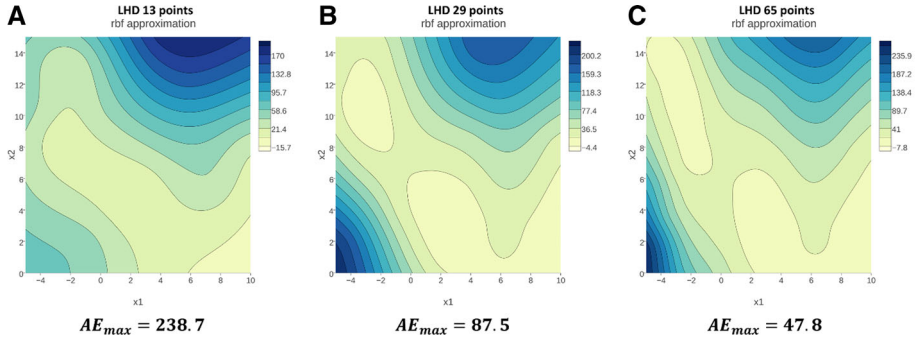

Fig. 9 Radial basis function approximations of the Branin function based on the designs of Fig. 5. **a** LH design with 13 points, **b** LH design with 29 points, **c** LH design with 65 points

the example shown here, as the samples increase from 29 to 65 only a small improvement in the maximum error is observed (Fig. 8). Finally, the radial basis functions consistently improve as more points are used, however, the best approximation has a maximum absolute error of 47.8 when using 65 points (Fig. 9). It must be mentioned that if multiple initial LH designs are used to fit kriging and radial basis functions, the final surrogate functions may have a lower or higher maximum absolute error. This phenomenon is the root cause of the variability in the performance of many surrogate-based optimization methods.

Finally, it is important to show the closed-form representation of all of the fitted functions developed for this motivating case. In Table 3, the expressions of the four different methods are shown for the first level of 13 points. In terms of the SG-Cheb, there are 9 parameters, polynomial terms up to the 4th degree, one bilinear and one trilinear term. The kriging and radial basis functions contain as many terms as the number of samples and each term is a composite nonlinear term. A widely used form of radial basis functions is used, namely first order and spline kernel radial basis. The parameters of each function are denoted by b_i, c_i, ν, θ_i, while $x^{(i)}$ corresponds to a specific value of x for grid point i. None of the terms of the quadratic, kriging or radial basis functions were found to be negligible.

Table 3 Comparison of functional forms for Branin function approximations using 13 points

Function type	Functional form
SG-Chebyshev	$f_{sp}(x) = b_0 + b_1 x_1 + b_2\left(2x_1^2 - 1\right) + b_3 x_2 + b_4\left(2x_2^2 - 1\right)$
LH-quadratic	$f_{sp}(x) = b_0 + b_1 x_1 + b_2 x_2 + b_{11} x_1^2 + b_{12} x_1 x_2 + b_{22} x_2^2$
LH-kriging	$f_{sp}(x) = v + \sum_{i=1}^{13} c_i exp\left(\sum_{j=1}^{2} -\theta_j \left(x_j^{(i)} - x_j\right)^2\right)$
LH-RBF (1st order, spline)	$f_{sp}(x) = v + b_1 x_1 + b_2 x_2 + \sum_{i=1}^{13} c_i \sum_{j=1}^{2}\left[\left(x_j^{(i)} - x_j\right)^2 ln\left(x_j^{(i)} - x_j\right)\right]$

4 Algorithmic description

4.1 Motivation

Most existing surrogate-based optimization algorithms use non-restrained adaptive sampling, where there is no restriction on the position of samples that can be added at a given iteration of the algorithm. Although non-restrained sampling offers the advantage of increased flexibility to sample any point in space, one cannot guarantee to develop surrogate functions in which the approximation error is always reduced with the addition of new samples. In developing the proposed algorithm, we wished to take advantage of the superior convergence properties of Chebyshev polynomials on Smolyak grids to build surrogate functions whose approximation error would improve with the addition of new samples, while efficiently searching high dimensional spaces. To take advantage of the sampling properties of Smolyak grids our sampling scheme would need to be restricted to the fixed grid points of the Smolyak grid that are defined by the choice of unidimensional grid. Additionally, samples should be added following the hierarchy of the Smolyak algorithm, meaning that at least locally, all points of a previous level should be added prior to adding points of a higher level. Hierarchically adding points will serve to limit the complexity of the surrogate function, since higher level points in the Smolyak grid are associated with higher order polynomial terms.

The typical approach for surrogate-based optimization is to identify new candidate samples by optimizing the current surrogate function, evaluate the objective value of the candidate samples, and finally, add the newly collected samples to the sample pool that is used to update the surrogate function for the next iteration. The developed algorithm generally follows this paradigm. However, to use the Smolyak-based sampling scheme described above, the only points that can be considered as candidate samples are points that lie on the Smolyak grid and have a level (μ) that is near the maximum level of any member of the sample pool. These restrictions mean that as the algorithm adaptively identifies new sampling locations, there exists a finite set of candidate samples to be considered, eliminating the need for optimizing the surrogate function to identify candidate samples at every iteration. At the same time, the entire premise of surrogate-based optimization is that optimization of the surrogate function will lead to the identification of the global optimum, especially if the surrogate function is highly accurate. Our algorithm alternates between iterations that add points simply from a pool of potential promising candidate SG points, and iterations that optimize surrogate functions to identify optimal points that do not lie on the predefined Smolyak Grid. To update the surrogate function with a point that is not a candidate point of a current Smolyak grid, the algorithm updates the bounds of the grid search to be centered around optimal samples

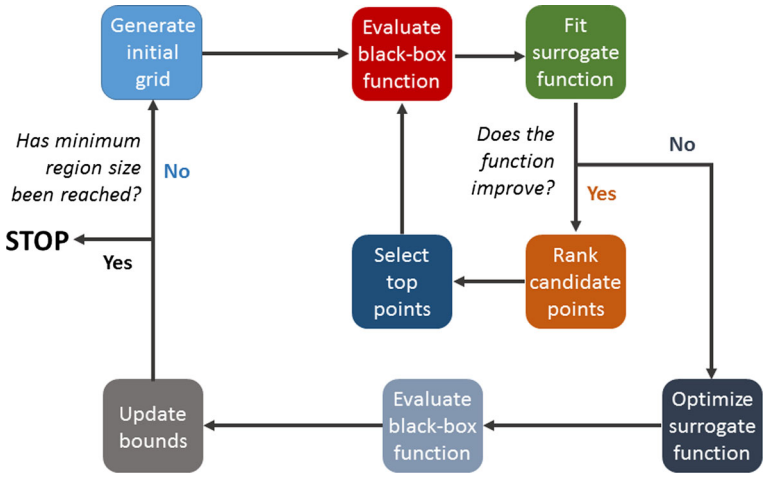

Fig. 10 Smolyak grid-based optimization algorithm (SGO algorithm)

identified by a previous iteration. Bound refinement is a standard procedure in many black-box algorithms, so the proposed algorithm adaptively builds surrogate functions, continuing to add points/terms until the function no longer improves, and subsequently optimizes the function and reduces the bounds to allow new points to be sampled.

In brief, the algorithm is composed of two loops (Fig. 10). An inner loop that adaptively builds a Smolyak approximation and an outer loop which refines the bounds based on the geometric center of the current best point(s) and a specified reduction factor. The inner loop starts with a base Smolyak grid of a specified level given the current bounds, and evaluates the black-box function at these grid points to build an initial approximation. The inner loop continues to adaptively add points/terms to the approximation, until the approximation no longer improves. Subsequently, the outer loop optimizes the arrived at approximation, and evaluates the original (or black-box) function at the identified points. Due to the fact that training points must be Smolyak Grid points, no optimization is performed within the inner loop. Instead of optimization, the fitted polynomial is used to predict the values of the black-box function at points of various level SG points that have not been sampled yet, and a ranking score is used to identify the next samples. The outer loop updates the bounds and continues to iterate until the updated bounds reach a specified minimum hyper-volume. The overview of the algorithm is shown in Fig. 10, while Fig. 11 is a more detailed description of the algorithmic steps and the default settings. The following paragraphs will describe key elements of the algorithm in greater detail.

4.2 Initializing algorithm

The algorithm is initialized by setting the number of variables (n), their upper and lower bounds $\left([\boldsymbol{x}^{lo}, \boldsymbol{x}^{up}]\right)$, and the initial level of the Smolyak Grid (μ_{init}). This is the only information that is required to initialize the algorithm, which will first scale the original search space within the $[-1, 1]^n$ domain; create the grid; collect the initial set of samples by calling the "black-box", and finally fit the resulting polynomial. The algorithm has some additional parameters, which can be tuned depending on the characteristics of the problem. These are the bound refinement rate (brr) and the hyper-volume of the search space for

Provide: black-box function $f(x)$ and variable bounds $[x^{lo}, x^{up}]$.
Initialize parameters: $\mu_{init}, \mu_{max}, brr, ss_{final}, w_{grid}, w_{obj}, \varepsilon_{improv}, N_{max}$

While $ss \geq ss_{final}$
  Set $\mu = \mu_{init}$. Create μ_{init} grid: $\mathbf{X}_{\text{init}} \in \mathrm{R}^{[\mathrm{n} \times \mathrm{d}]}$.
  Collect $f(\mathbf{X}_{\text{init}})$.
  Calculate parameters of surrogate approximation $f_{sp}(\mathrm{x})$.
  Create pool of potential candidate samples $\mathbf{X}_{\text{pool}}$ by including SG points up to level $\mu_{init} + 1$.

  While $\varepsilon \geq \varepsilon_{improv}$ and $\mathbf{X}_{\text{pool}} \neq \emptyset$
    Estimate $f_{sp}(\mathrm{x})$ at $\mathbf{X}_{\text{pool}}$.
    Rank all points within $\mathbf{X}_{\text{pool}}$ using consensus rank $w_{grid} r_{grid} + w_{obj} r_{obj}$.
    Select m best candidate samples $\mathbf{X}_{\text{best}} \subset \mathbf{X}_{\text{pool}}$ and evaluate $f(\mathrm{x})$.
    Update $f_{sp}(\mathrm{x})$ parameters.
    If $\mathbf{X}_{\text{pool}} = \emptyset$ and $\mu < \mu_{max}$, set $\mu = \mu + 1$ and update $\mathbf{X}_{\text{pool}}$.
  End

  Multi-start local optimization of $f_{sp}(x)$. Identify l unique optima $\mathbf{X}_{\text{opt}}$.
  Collect $f(\mathbf{X}_{\text{opt}})$.
  Update $f_{incumbent}$ as best sample out of all samples.
  Update search space center at $f_{\text{incumbent}}$.
  Reduce search space radius ss by brr. Update bounds $[\mathrm{x}^{\mathrm{lo}}, \mathrm{x}^{\mathrm{up}}]$.
End

Fig. 11 Detailed steps of proposed algorithm

convergence (ss_{final}). The bound refinement rate controls the rate at which the search space will be reduced at each iteration. The algorithm converges when the hyper-volume of the search space is very small, which is a convergence criterion that is theoretically linked to convergence to local stationary and gradient matching between the surrogate function and the black-box function [8]. Parameter ss_{final} controls how small this region should be relative to the original search space and as expected, the smaller its value, the longer the algorithm will need to converge.

4.3 Adaptive surrogate generation

As shown in Fig. 10, the surrogate function is updated adaptively within the inner loop, until it does not further improve. The surrogate model is considered to no longer improve when the sum of the absolute values of the surrogate model parameters for all new points added at a given iteration, is below a small threshold $(\varepsilon_{improv} \leq 1e^{-9})$. As described earlier, no further improvement in this type of polynomial interpolation suggests that the surrogate fit is an accurate representation of the black-box function.

In order to identify the points which will be added per iteration, we have formulated a ranking score that aims to balance between adding lower order points (lower level SG points), points with low predicted objective function value and points in the proximity of the current incumbent solution. The main restriction that we have is that new points must be points of a SG of the studied bounded region. For example, if the algorithm starts with Smolyak Grid of level 1, then any points of grid of level 2 or more could be candidate points, and since we have a surrogate approximation we can certainly estimate the objective function value at those points. However, as the level of the grid increases, this is also associated with higher order polynomial terms in the surrogate function, which should be avoided. The ranking function

is a consensus rank, or a weighted sum $\left(w_{grid}r_{grid} + w_{obj}r_{obj},\ \text{where } w_{grid} + w_{obj} = 1\right)$ of the respective ranks of each point based on the SG level $\left(r_{grid}\right)$ and the predicted objective function $\left(r_{obj}\right)$. For example, each of the points in the candidate pool of new samples has a rank to represent its grid level (lowest level rank is equal to 1), and a rank to represent its predicted objective function value (point with best objective function value has a rank equal to 1). The weights of the consensus rank are algorithmic parameters that can be tuned, depending on whether surrogate model simplicity is favored over exploring promising regions. In other words, as $w_{obj} \to 1$, the algorithm prefers points with good objective function values, while as $w_{grid} \to 1$, the algorithm would prefer points with the lowest level. To maintain diversity in the new samples, 10% of the lowest ranked points are added to update the surrogate function.

4.4 Surrogate optimization and bounds refinement

When an inner loop has converged, it is assumed that we have built a polynomial function that is as best as possible for the specific search space. Thus, to move forward the algorithm optimizes the best approximation, reduces the search space and repeats the procedure outlined above. At this stage, optimization is used to identify whether there are points predicted to have better objective function values that are not SG points. In this work, we have used a local optimizer initialized multiple times with a diverse set of existing samples with promising objective function values. All uniquely identified local optima are sampled using the black-box simulator, and if a better solution is found, we update the incumbent solution. The search space is reduced by the bound refinement fraction around the incumbent solution at this stage of the algorithm. Once the bounds are updated, the SG is re-initialized and the entire procedure is repeated until the search space reaches the minimum hyper-volume required for convergence. Due to the nature of the SG grids, as soon as the bounds are refined, it is highly unlikely that points from the previous search space will be used in the refined space and as a result, an entirely new SG grid needs to be created. This is one drawbacks of using fixed grids, and it may be a source of increased sampling requirements of the algorithm. However, this is necessary in order to maintain the predictive properties of SG grids. Moreover, it is important to mention that as the search-space is reduced, the algorithm typically requires less samples of lower SG levels (μ) in order to accurately approximate the black-box function.

5 Computational results

5.1 Motivating example revisited

As an introduction to the evolution and performance of the proposed algorithm we will continue with our motivating example, the Branin function. Figure 12a, b illustrate the adaptive sampling of the SGO algorithm with respect to the iteration in which the sample was collected and the objective value of the sample, respectively. Starting with an initial grid of level 1, which includes 5 samples, the first iteration of the adaptive surrogate generation collected 19 samples (Fig. 12a—red points). For the Branin function, the approximation following the first iteration of the algorithm (Fig. 12c) has a maximum absolute error of 2.9 (Fig. 12d). To achieve an equivalent approximation using full Smolyak grids an approximation level of 3 (Fig. 5b), corresponding to 29 samples, is required. Local optimization of the surrogate model of the first iteration (Fig. 12c) identifies a point that has a Euclidean distance less than 0.3 from one of the global optima of the Branin function, which defined the first set of refined bounds. For this motivating example we have used the default parametric values of bound

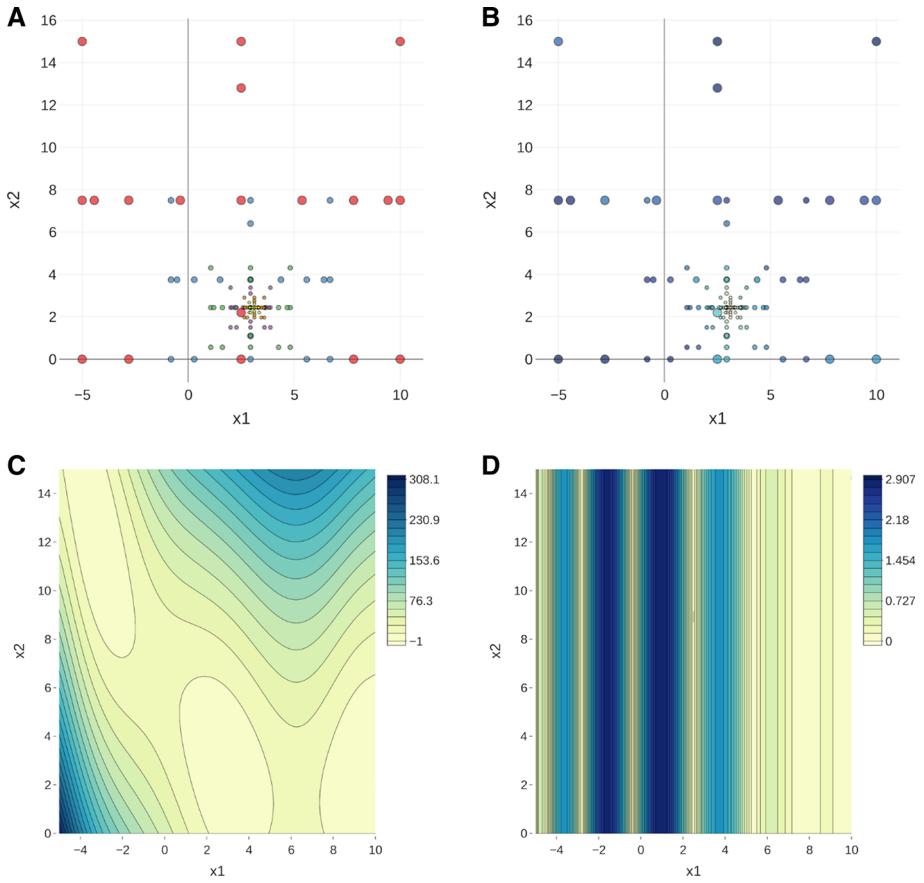

Fig. 12 Algorithmic example using the Branin function. **a** Samples colored by iteration, **b** Samples colored by objective value, **c** Approximation following the first iteration using the full bounds, **d** error of first approximation

reduction factor of 0.5 and minimum region size that is 5% of the hyper-volume defined by the initial full bounds. Figure 12a shows how the sampling focused with each of the 6 iterations, with the color changing for each iteraction and the radius of the points reducing when going from the first to the last iteration. Figure 12b illustrates how the objective function improved with each iteration. Ultimately, the SGO algorithm identifies a point that satisfies the solution criteria in 40 samples, and converges to one of the global optima of the Branin function in 112 samples.

5.2 Testing on bound-constrained benchmark problems

The performance of the algorithm is tested on a set of benchmark problems for bound constrained optimization, out of the library of Rios and Sahinidis [11]. 95 of the test problems have 2–3 variables, 74 problems have 4–9 variables and 20 have 10–30 variables. The test problems include both convex and nonconvex problems. It must be noted that a reasonable number of the benchmark problems have a global optimum in the middle of the multidimensional search space and it was realized that this point is by default included as a sample in

the first level SG. In order to avoid this and truly test the performance of the method, each problem's search space was slightly perturbed at initialization. Following this approach, we avoid sampling the global optimum during the initial sampling and test the ability of the algorithm in converging to the true optimum.

Several performance metrics are used to assess the performance of the methodology. First, both the absolute error: $Err_{abs} = abs\left(f_{incumbent}(x) - f_{global}(x)\right)$ and relative error $Err_{rel} = \frac{abs\left(f_{incumbent}(x) - f_{global}(x)\right)}{median\left(f_{sampled}(x)\right) - f_{global}(x)}$ (normalized by the distance of the median sampled value to the global optimum) between the incumbent solution at convergence ($f_{incumbent}(x)$) from the global optimum $\left(f_{global}(x)\right)$ are used to assess how close we are to identifying the global optimum. Lastly, we report performance curves in terms of the number of problems solved, where a solved problem is considered to be one that either the relative or absolute error are less or equal to 1%.

To better quantify the performance of the SGO algorithm, we have tested the algorithm on a set of benchmark problems and provide a side-by-side comparison with a black-box optimization solver, ARGONAUT [12, 37]. The ARGONAUT framework has recently been developed for grey-box optimization problems with complex constraints, and it is based on optimized sampling, bound refinement and identifying the best surrogate function for each of the objective and constraints out a set of potential basis functions (linear, quadratic, kriging, signomial, radial basis functions). The ARGONAUT algorithm is run at its default settings three times for each test problem, where each time a different initial LHD design is used. The computational time and total samples reported for ARGONAUT are the respective sums of the three runs. The reason for performing multiple iterations of the ARGONAUT framework, is due to the fact that each time the algorithm is initialized with a slightly different sampling set and this sometimes leads the algorithm to converge to different local optima. This does not happen with the current algorithm, which is purely deterministic. For fairness, we report the best obtained solution for both algorithms and the total number of samples required to obtain this result.

The two most important parameters that affect the performance and cost of the algorithm are the initial Smolyak grid level and the bounds reduction factor. Therefore, an investigation of these parameters was performed. We have performed a series of tests and identified two suggested settings, namely a conservative case and a non-conservative case. In both the conservative and non-conservative versions of the algorithm, the bound reduction factor is 50%, while the final search space is the distinguishing parameter and was set to equal to 5% and 0.05%, respectively. This means that at every outer loop iteration, the search space is reduced to 50% of the size of the current iteration around the incumbent solution at that point, while the algorithm converges when the search region hyper-volume is less than 5% or 0.05% of the original search space. This "conservative" setting refers to the fact that the final search space hyper-volume is generous leading to faster convergence, while in the "non-conservative" case, many more samples are collected since the final search space hyper-volume is 100 times smaller. We have also performed tests with μ_{max} ranging from 3 to 5.

In order to test the global convergence performance of the SGO algorithm we did not place any limits on the number of samples collected or the computational run time. In its default settings, the SGO algorithm solved 89 of the 95 problems of 2–3 variables, while ARGONAUT solved 86, corresponding to success rates of 94% and 91% respectively (Fig. 13a, b). The SGO algorithm solved 67 of the 74 problems of 4–9 variables, while ARGONAUT solved 61, corresponding to success rates of 91% and 82% respectively (Fig. 13c, d). The main advantage of the ARGONAUT algorithm is the reduced number of samples that are used for problems of 4–9 variables, however, due to the properties of the surrogate functions consid-

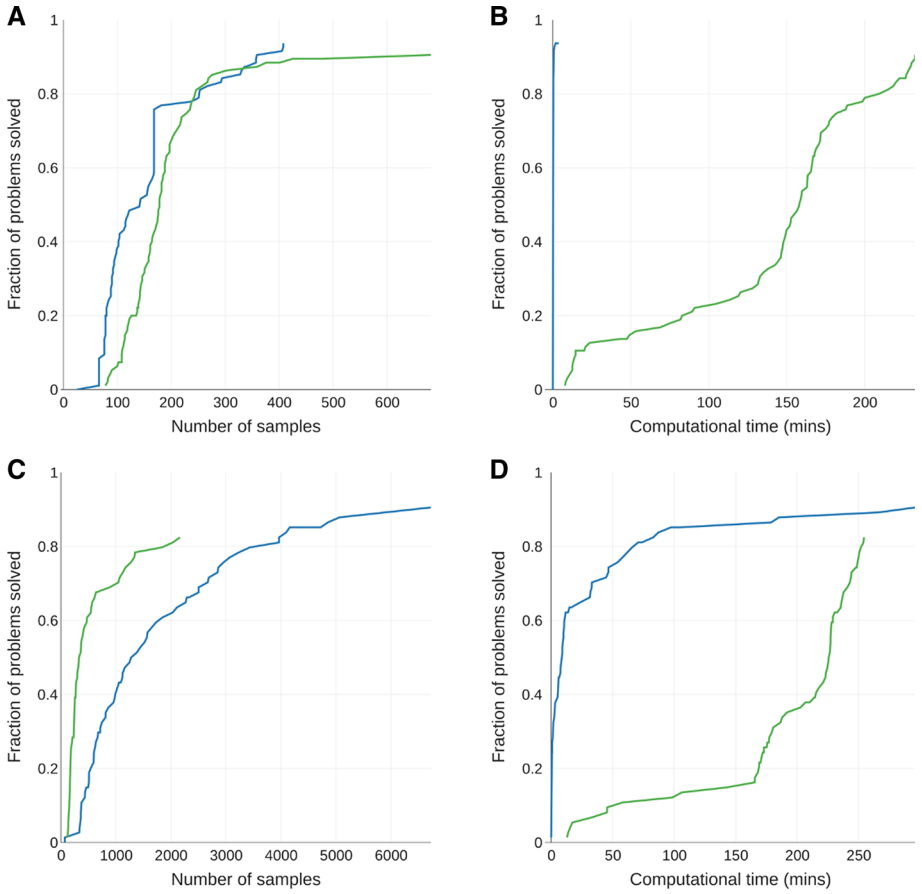

Fig. 13 Performance comparison between ARGONAUT (green) and SGO (blue). SGO uses conservative sampling ($\mu_{\max} = 3$) and bounds convergence. **a**, **b** Problems with 2 or 3 variables, **c**, **d** problems with 4–9 variables. (Color figure online)

ered it converges prematurely. The increased computational cost of ARGONAUT is due to the use of global optimization for parameter estimation, sampling selection and optimization of the grey-box problems, however, this cost would pay off significantly as the cost of the individual simulation sample increases. Finally, it must be mentioned that the benchmark problems for bound constrained optimization are on average highly complex (multimodal and nonconvex) and thus they are excellent for testing the good approximating capabilities of SG-based polynomials, when compared to commonly used interpolating surrogate methods.

To evaluate the performance of the SGO algorithm on problems with greater than 9 variables we applied the SGO algorithm to a small set of problems with 10–30 variables, which is summarized by Table 4. Of the 20 problems of 10–30 variables, only one problem, hs110, was not solved using our default parameter values. Understandably, the SGO algorithm required more samples for the larger problems, with the number of samples used for the 10–30 variable problems ranging from approximately 2000 to over 50,000 samples. To put this in perspective, full tensor grids, with 4 possible values in each dimension, would contain over 1 million or 1 quintillion samples in 10 or 30 dimensions, respectively (Table 1).

Table 4 Performance on test problems with 10–30 variables

Problem name	Dim	Global Opt	Incumbent Opt	Absolute error	relative error	Incumbent Opt sample	Total samples	Time (min)
arglinb	10	4.63E+00	4.63E+00	1.58E−04	1.74E−11	5093	7821	160.1
brownal	10	0.00E+00	9.94E−01	9.94E−01	4.62E−04	7780	9324	413.3
dixon3dq	10	0.00E+00	1.04E−01	1.04E−01	2.90E−04	7333	7916	69.2
extrosnb	10	0.00E+00	1.27E−02	1.27E−02	1.04E−08	446	3915	10.0
hilberta	10	6.00E−10	1.46E−05	1.46E−05	4.74E−07	1432	8275	95.1
hs110	10	−4.58E+01	−3.34E+01	1.24E+01	6.42E−02	9018	9467	1081.0
s281	10	1.45E−05	1.28E−01	1.28E−01	2.92E−03	9412	9480	236.3
s282	10	0.00E+00	9.43E−01	9.43E−01	1.40E−06	560	3757	10.8
s283	10	6.00E−10	2.42E+00	2.42E+00	1.20E−08	4410	4784	78.1
s291	10	0.00E+00	1.00E−07	1.00E−07	1.67E−10	319	1908	0.5
s295	10	3.99E+00	3.99E+00	2.10E−04	1.36E−10	1781	4528	32.9
osborneb	11	4.01E−02	6.88E−01	6.48E−01	3.02E−04	8972	10,846	1076.7
qudlin	12	−7.20E+03	−7.20E+03	0.00E+00	0.00E+00	3738	4428	15.6
s296	16	0.00E+00	5.43E−03	5.43E−03	4.13E−09	3532	13,012	192.5
s286	20	0.00E+00	3.56E−02	3.56E−02	3.94E−08	2231	21,829	276.4
s287	20	0.00E+00	9.16E−04	9.16E−04	1.37E−09	9902	21,899	496.6
s288	20	0.00E+00	1.75E+01	1.75E+01	1.27E−04	20,439	27,399	4101.6
3pk	30	1.72E+00	4.70E+00	2.98E+00	8.37E−07	39,452	49,692	2179.0
s292	30	0.00E+00	2.11E−06	2.11E−06	9.53E−10	7260	43,554	420.4
s297	30	0.00E+00	6.78E+00	6.78E+00	3.44E−06	7260	52,454	1508.0

SGO uses conservative sampling ($\mu_{\max} = 3$) and final search region hyper-volume

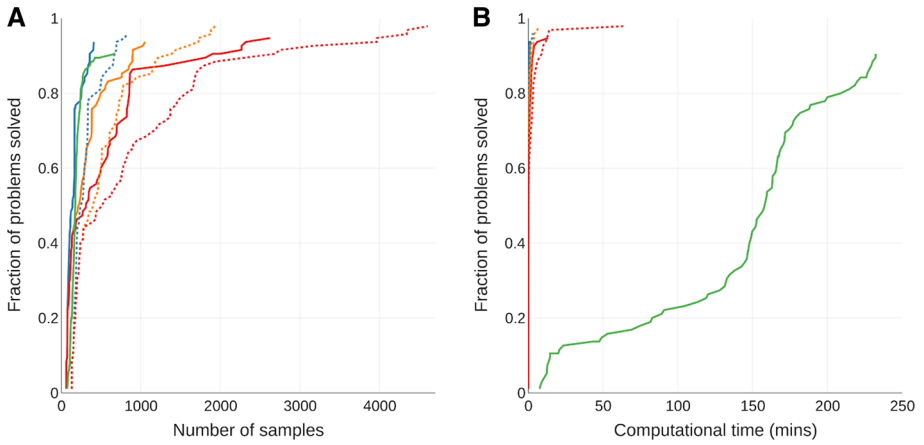

Fig. 14 Comparison of algorithmic parameters. Blue: $\mu_{\max} = 3$, orange: $\mu_{\max} = 4$, red: $\mu_{\max} = 5$. Solid lines are conservative bound reduction, while dashed lines are non-conservative bound reduction. Solid green line is ARGONAUT. (Color figure online)

What is important to note is that by modifying slightly the three main parameters of the algorithm the algorithm was able to locate the global optimum of all problems, based on the tolerance that we have set (Fig. 14). All but two of the 95 problems containing two or three variables were solved by one of the 6 runs, which have three different settings for μ_{max} and ss_{final}. For example, one problem that wasn't solved (s202) was solved by changing the bounds reduction factor from 0.50 to 0.10. Parameter tuning of derivative-free optimization solvers is a valid concern, however, this algorithm is advantageous in this respect since it has few parameters that need to be tuned. In addition, these parameters have a direct link to the nature of the problem and one could use this knowledge (if available) to tune the parameters based on (a) the cost of the black-box simulation and (b) the expected non-smoothness and multimodality of the problem. For example, we have found that starting with a higher SG level (μ_{init}) has a positive effect on performance. This would be especially suggested for problems where the cost of the simulation is not a very limiting factor. In other words, starting with a better approximating function with more investment in the initial sampling leads to improved performance. This may seem like an obvious conclusion, however the balance between initial versus adaptive sampling is an open question in surrogate-based optimization. In this work, this conclusion further supports the superior performance of the Smolyak-grid based polynomials, because by increasing μ, there is certainty that the approximation will be better. Also, we observe that non-conservative bound reduction (smaller brr and ss_{final}) increases time to convergence and sampling requirements significantly, but increases the success rate of the algorithm. Thus, if it can be afforded computationally, it provides a more reliable and consistent convergence behavior, especially when the black-box function is highly nonlinear.

The major advantage of the SGO algorithm is its ability to continue to sample the problem in order to converge consistently to the global optimum. It is believed that this is directly linked to the low approximation errors and convergence properties of the polynomials used. Another main positive aspect of this algorithm is the simplicity in its operations. In fact, once the SG generation and polynomial interpolation algorithms are implemented, the algorithm consists of further sampling on a predefined set of grid point locations, solving systems of linear equations and solving a few local optimization sub-problems. This simplicity in

operations is indicative of its low overall computational cost. Moreover, this cost can be significantly reduced by parallelizing the collection of samples and pre-computing and storing the Smolyak grids and the corresponding polynomial basis functions. The current version of the algorithm may not be applicable to very computationally expensive simulations in higher dimensions, due to the high number of samples required. In fact, by studying at which point of the algorithm the best solution was found versus the total number of samples required for convergence, it is obvious that the algorithm finds the global optimum with a significantly lower number of samples (Table 4). This result implies that there is room for improvement in terms of identifying more efficient convergence criteria. In addition, this result implies that if the algorithm stopped prior to convergence due to a maximum number of samples limitation, the best solution can still be a good one.

6 Conclusions

We have presented a surrogate-based optimization algorithm that samples the bound constrained space based on a Smolyak-grid, and interpolating polynomial approximations. The first aim of this work is to highlight the good properties of Sparse-Grids and polynomial functions, which make them great candidates for surrogate-based optimization. We believe that this is a timely contribution due to the recent increase in the academic and industrial interest in methods that can reliably optimize problems using input–output data streams. The second aim of this work is the introduction and testing of a new method that uses SG-based sampling and polynomial approximations, and borrows concepts from the derivative-free optimization field to converge to an optimal solution. The results that are reported on 189 test problems show that this algorithm, when well-tuned, can find the global optimum for all problems. The algorithm is shown to converge to the global solutions with low computational cost, while the number of samples for convergence is low for low-dimensional problems, but increases with dimensionality. One of the advantages of this algorithm is that it is a self-contained R-based algorithm that we plan to make available as a package to the community in the near future. However, one of the drawbacks of this implementation is that the polynomial approximations were not globally optimized using a deterministic solver. Although the multi-start optimization approach has a very good performance, future work will investigate the effects of global optimization on the efficiency and computational cost of the algorithm. Future work will also involve using error bound analysis to study the convergence properties of the algorithm, as well as enhancing the capabilities of the algorithm to be able to handle constraints.

Acknowledgements The authors would like to acknowledge the contribution of the late Professor Christodoulos A Floudas who introduced the authors to this area. CAF was a visionary and pioneer in the areas of global optimization, computational biology, energy systems, grey-box optimization and machine learning. This work was initiated when CAK and FB were postdoctoral associates in the laboratory of CAF, but was completed after his passing when they joined Georgia Institute of Technology.

References

1. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**(3), 385–482 (2003)
2. Conn, A.R., Le Digabel, S.: Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. Optim. Methods Softw. **28**(1), 139–158 (2013)

3. Davis, E., Ierapetritou, M.: A kriging-based approach to MINLP containing black-box models and noise. Ind. Eng. Chem. Res. **47**(16), 6101–6125 (2008)
4. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Glob. Optim. **13**(4), 455–492 (1998)
5. Regis, R.G.: Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Eng. Optim. **46**(2), 218–243 (2014)
6. Audet, C., Dennis Jr., J.E.: A progressive barrier for derivative-free nonlinear programming. SIAM J. Optim. **20**(1), 445–472 (2009)
7. Boukouvala, F., Ierapetritou, M.G.: Derivative-free optimization for expensive constrained problems using a novel expected improvement objective function. AIChE J. **60**(7), 2462–2474 (2014)
8. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. SIAM, Philadelphia (2009)
9. Echebest, N., Schuverdt, M.L., Vignau, R.P.: A derivative-free method for solving box-constrained underdetermined nonlinear systems of equations. Appl. Math. Comput. **219**(6), 3198–3208 (2012)
10. Le Thi, H.A., Vaz, A.I.F., Vicente, L.N.: Optimizing radial basis functions by d.c. programming and its use in direct search for global derivative-free optimization. Top **20**(1), 190–214 (2012)
11. Rios, L.M., Sahinidis, N.V.: Derivative-free optimization: a review of algorithms and comparison of software implementations. J. Glob. Optim. **56**(3), 1247–1293 (2013)
12. Boukouvala, F., Floudas, C.A.: ARGONAUT: AlgoRithms for global optimization of coNstrAined grey-box compUTational problems. Optim. Lett. **11**(5), 895–913 (2017)
13. Eason, J.P., Biegler, L.T.: A trust region filter method for glass box/black box optimization. AIChE J. **62**(9), 3124–3136 (2016)
14. Amaran, S., et al.: Simulation optimization: a review of algorithms and applications. 4OR **12**(4), 301–333 (2014)
15. Amaran, S., et al.: Simulation optimization: a review of algorithms and applications. Ann. Oper. Res. **240**(1), 351–380 (2016)
16. Cozad, A., Sahinidis, N.V., Miller, D.C.: Learning surrogate models for simulation-based optimization. AIChE J. **60**(6), 2211–2227 (2014)
17. Jakobsson, S., et al.: A method for simulation based optimization using radial basis functions. Optim. Eng. **11**(4), 501–532 (2010)
18. Quan, N., et al.: Simulation optimization via kriging: a sequential search using expected improvement with computing budget constraints. IIE Trans. **45**(7), 763–780 (2013)
19. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**(1), 4–31 (2011)
20. Egea, J.A., Martí, R., Banga, J.R.: An evolutionary method for complex-process optimization. Comput. Oper. Res. **37**(2), 315–324 (2010)
21. Ong, Y.S., Nair, P.B., Keane, A.J.: Evolutionary optimization of computationally expensive problems via surrogate modeling. Aiaa J. **41**(4), 687–696 (2003)
22. Boukouvala, F., Misener, R., Floudas, C.A.: Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO. Eur. J. Oper. Res. **252**(3), 701–727 (2016)
23. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. J. ACM **8**(2), 212–229 (1961)
24. Nedler, J.A., Mead, R.: A simplex method for function minimization. Comput. J. **7**, 308–313 (1965)
25. Booker, A.J., et al.: A rigorous framework for optimization of expensive functions by surrogates. Struct. Multidiscip. Optim. **17**(1), 1–13 (1999)
26. Boukouvala, F., Ierapetritou, M.G.: Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing. J. Pharm. Innov. **8**(2), 131–145 (2013)
27. Caballero, J.A., Grossmann, I.E.: An algorithm for the use of surrogate models in modular flowsheet optimization. AIChE J. **54**(10), 2633–2650 (2008)
28. Ciaurri, D.E., Mukerji, T., Durlofsky, L.J.: Derivative-free optimization for oil field operations. In: Yang, X.-S., Koziel, S. (eds.) Computational Optimization and Applications in Engineering and Industry, pp. 19–55. Springer, Berlin (2011)
29. Egea, J.A., et al.: Scatter search for chemical and bio-process optimization. J. Glob. Optim. **37**(3), 481–503 (2007)
30. Torn, A., Zilinskas, A.: Global Optimization. Springer, New York (1989)
31. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. J. Glob. Optim. **21**(4), 345–383 (2001)

32. Powell, M.J.D.: A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Gomez, S., Hennart, J.-P. (eds.) Advances in Optimization and Numerical Analysis, p. 51–67. Springer, Dordrecht (1994)
33. Powell, M.J.: The BOBYQA Algorithm for Bound Constrained Optimization Without Derivatives. Cambridge NA Report NA2009/06. University of Cambridge, Cambridge (2009)
34. Wilson, Z.T., Sahinidis, N.V.: The ALAMO approach to machine learning. Comput. Chem. Eng. **106**, 785–795 (2017)
35. Davis, E., Ierapetritou, M.: A kriging based method for the solution of mixed-integer nonlinear programs containing black-box functions. J. Glob. Optim. **43**(2–3), 191–205 (2009)
36. Palmer, K., Realff, M.: Optimization and validation of steady-state flowsheet simulation metamodels. Chem. Eng. Res. Des. **80**(7), 773–782 (2002)
37. Boukouvala, F., Hasan, M.F., Floudas, C.A.: Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption. J. Global Optim. **67**(1–2), 3–42 (2017)
38. Gutmann, H.-M.: A radial basis function method for global optimization. J. Glob. Optim. **19**(3), 201–227 (2001)
39. Muller, J., Shoemaker, C.A., Piche, R.: SO-MI: a surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems. Comput. Oper. Res. **40**(5), 1383–1400 (2013)
40. Regis, R.G., Shoemaker, C.A.: A quasi-multistart framework for global optimization of expensive functions using response surface models. J. Glob. Optim. **56**(4), 1719–1753 (2013)
41. Eason, J., Cremaschi, S.: Adaptive sequential sampling for surrogate model generation with artificial neural networks. Comput. Chem. Eng. **68**, 220–232 (2014)
42. Fahmi, I., Cremaschi, S.: Process synthesis of biodiesel production plant using artificial neural networks as the surrogate models. Comput. Chem. Eng. **46**, 105–123 (2012)
43. Henao, C.A., Maravelias, C.T.: Surrogate-based process synthesis. Comput. Aided Chem. Eng. **28**, 1129–1134 (2010)
44. Martelli, E., Amaldi, E.: PGS-COM: a hybrid method for constrained non-smooth black-box optimization problems: Brief review, novel algorithm and comparative evaluation. Comput. Chem. Eng. **63**, 108–139 (2014)
45. Novak, E., et al.: Smolyak/sparse grid algorithms. Tractability Multivar. Probl. Std. Inf. Funct. **12**, 320–397 (2010)
46. Plaskota, L., Wasilkowski, G.W.: Smolyak's algorithm for integration and L-1-approximation of multivariate functions with bounded mixed derivatives of second order. Numer. Algorithms **36**(3), 229–246 (2004)
47. Bungartz, H.J., Dirnstorfer, S.: Multivariate quadrature on adaptive sparse grids. Computing **71**(1), 89–114 (2003)
48. Bungartz, H.J., Dirnstorfer, S.: Higher order quadrature on sparse grids. In: Computational Science—Iccs 2004. In: Bubak, M., et al. (Ed.) Proceedings, p. 394–401 (2004)
49. Bungartz, H.-J., Pfluger, D., Zimmer, S.: Adaptive sparse grid techniques for data mining. Springer, Berlin (2008)
50. Harding, B.: Adaptive Sparse Grids and Extrapolation Techniques. In: Garcke, J., Pfluger, D. (eds.) Sparse Grids and Applications—Stuttgart 2014. Lecture Notes in Computational Science and Engineering, vol 109. Springer, Berlin (2016)
51. Jiang, Y., Xu, Y.S.: B-spline quasi-interpolation on sparse grids. J. Complex. **27**(5), 466–488 (2011)
52. Pfluger, D., Peherstorfer, B., Bungartz, H.J.: Spatially adaptive sparse grids for high-dimensional data-driven problems. J. Complex. **26**(5), 508–522 (2010)
53. Sickel, W., Ullrich, T.: Spline interpolation on sparse grids. Appl. Anal. **90**(3–4), 337–383 (2011)
54. Zenger, C.: Sparse grids. In: Hackbusch, W. (ed.) Parallel algorithms for partial differential equations. Proceedings of the Sixth GAMM Seminar. Notes on numerical fluid mechanics. Braunschweig, Verlag Vieweg, vol 31 (1991)
55. Novak, E., Ritter, K.: Global optimization using hyperbolic cross points. In: Floudas, C.A., Pardalos, P.M. (eds.) State of the Art in Global Optimization: Computational Methods and Applications. Springer, Boston (1996)
56. Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. In: *Dokl. Akad. Nauk SSSR* (1963)
57. Gerstner, T., Griebel, M.: Numerical integration using sparse grids. Numer. Algorithms **18**(3–4), 209–232 (1998)
58. Dung, D.: Sampling and cubature on sparse grids based on a B-spline quasi-interpolation. Found. Comput. Math. **16**(5), 1193–1240 (2016)

59. Tang, J.J., et al.: Dimension-adaptive sparse grid interpolation for uncertainty quantification in modern power systems: probabilistic power flow. IEEE Trans. Power Syst. **31**(2), 907–919 (2016)
60. Peherstorfer, B., et al.: Selected recent applications of sparse grids. Numer. Math. Theory Methods Appl. **8**(1), 47–77 (2015)
61. Gajda, P.: Smolyak's algorithm for weighted L-1-approximation of multivariate functions with bounded rth mixed derivatives over R-d. Numer. Algorithms **40**(4), 401–414 (2005)
62. Xu, G.Q.: On weak tractability of the Smolyak algorithm for approximation problems. J. Approx. Theory **192**, 347–361 (2015)
63. Wasilkowski, G.W., Wozniakowski, H.: Explicit cost bounds of algorithms for multivariate tensor product problems. J. Complex. **11**(1), 1–56 (1995)
64. Valentin, J., Pfluger, D.: Hierarchical gradient-based optimization with b-splines on sparse grids. In: Garcke, J., Pfluger, D. (Eds.) Sparse Grids and Applications—Stuttgart 2014. Lecture Notes in Computational Science and Engineering, vol 109. Springer, Cham (2016)
65. Grimstad, B., Sandnes, A.: Global optimization with spline constraints: a new branch-and-bound method based on B-splines. J. Glob. Optim. **65**(3), 401–439 (2016)
66. Hulsmann, M., Reith, D.: SpaGrOW-A derivative-free optimization scheme for intermolecular force field parameters based on sparse grid methods. Entropy **15**(9), 3640–3687 (2013)
67. Sankaran, S.: Stochastic optimization using a sparse grid collocation scheme. Probab. Eng. Mech. **24**(3), 382–396 (2009)
68. Chen, P., Quarteroni, A.: A new algorithm for high-dimensional uncertainty quantification based on dimension-adaptive sparse grid approximation and reduced basis methods. J. Comput. Phys. **298**, 176–193 (2015)
69. Pronzato, L., Müller, W.G.: Design of computer experiments: space filling and beyond. Stat. Comput. **22**(3), 681–701 (2012)
70. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics **21**(2), 239–245 (1979)
71. Davis, E., Ierapetritou, M.: A centroid-based sampling strategy for kriging global modeling and optimization. AIChE J. **56**(1), 220–240 (2010)
72. Owen, A.B.: Orthogonal arrays for computer experiments, integration and visualization. Stat. Sin. **2**(2), 439–452 (1992)
73. Garud, S.S., Karimi, I.A., Kraft, M.: Smart sampling algorithm for surrogate model development. Comput. Chem. Eng. **96**, 103–114 (2017)
74. Fornberg, B., Zuev, J.: The Runge phenomenon and spatially variable shape parameters in RBF interpolation. Comput. Math Appl. **54**(3), 379–398 (2007)
75. Runge, C.: Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. Zeitschrift für Mathematik und Physik **46**(224–243), 20 (1901)
76. Xiang, S., Chen, X., Wang, H.: Error bounds for approximation in Chebyshev points. Numer. Math. **116**(3), 463–491 (2010)
77. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. Adv. Comput. Math. **12**(4), 273–288 (2000)
78. Davis, P.J., Rabinowitz, P.: Methods of Numerical Integration. Dover Publications (2007)
79. Judd, K.L., et al.: Smolyak method for solving dynamic economic models: lagrange interpolation, anisotropic grid and adaptive domain. J. Econ. Dyn. Control **44**, 92–123 (2014)
80. Trefethen, N.: Six myths of polynomial interpolation and quadrature. Math. Today **47**, 184–188 (2011)