Optim Lett (2017) 11:895–913
DOI 10.1007/s11590-016-1028-2

ORIGINAL PAPER

ARGONAUT: AlgoRithms for Global Optimization of coNstrAined grey-box compUTational problems

Fani Boukouvala[1,2] · **Christodoulos A. Floudas**[1,2]

Received: 12 June 2015 / Accepted: 9 March 2016 / Published online: 12 April 2016

Abstract The algorithmic framework ARGONAUT is presented for the global optimization of general constrained grey-box problems. ARGONAUT incorporates variable selection, bounds tightening and constrained sampling techniques, in order to develop accurate surrogate representations of unknown equations, which are globally optimized. ARGONAUT is tested on a large set of test problems for constrained global optimization with a large number of input variables and constraints. The performance of the presented framework is compared to that of existing techniques for constrained derivative-free optimization.

Keywords Grey-box optimization · Surrogate modeling · Variable selection · Derivative-free optimization · General constraints · Nonlinear programming

1 Introduction

We present a framework for the global optimization of general non-linear constrained grey-box problems. Grey-box problems are characterized by partial or total lack of closed-form equations describing the constraints and the objective of the problem. Grey-box problems arise in various fields which rely on expensive simulations, input-

Electronic supplementary material The online version of this article (doi:10.1007/s11590-016-1028-2) contains supplementary material, which is available to authorized users.

✉ Christodoulos A. Floudas
floudas@tamu.edu

[1] Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77843, USA

[2] Texas A&M Energy Institute, Texas A&M University, College Station, TX 77843, USA

output data, proprietary codes, or phenomena which have not yet been defined by physics-based mathematical equations. Important applications of grey-box modeling and optimization stem from various disciplines (e.g., chemical engineering, financial management, mechanical and aerospace engineering, geosciences, molecular engineering, and material science), where certain phenomena are accurately described by expensive finite-element or partial-differential equation based simulations, or even depend on unknown correlations for which solely input-output data exists [1–16]. In many applications, there is a need to efficiently integrate known closed-form models with explicitly unknown correlations in order to locate the globally optimal solution. The direct use of deterministic global optimization methods is prohibitive in the above cases due to the lack of analytical mathematical forms, which does not allow for the calculation of derivative information or the derivation of rigorous upper and lower bounds. Moreover, the computational cost of the model for obtaining the input-output data for any unknown correlations limits the total number of samples which can be obtained in order to optimize the overall grey-box system efficiently [17–19,21].

ARGONAUT (AlgoRithms for Global Optimization of coNstrAined Grey-Box compUTational systems) acts as a link between information provided in the form of input-output data and deterministic global optimization. This is achieved through the development of well-behaved, tractable surrogate approximations which accurately represent any unknown correlations. Surrogate models aim to guide the search towards the true optimum of the original model [17,22–34]. Surrogate models serve as analytical approximations to the underlying unknown equations, which are inexpensive to evaluate and allow for the use of derivative-based optimization. However, surrogate functional forms are often multimodal non-convex functions. For this reason, global optimization is required in order to solve grey-box optimization problems efficiently [35,36].

The paper is structured as follows. The general formulation of general constrained grey-box problems is defined in Sect. 2. Section 3 and its subsections describe the components of the algorithmic framework ARGONAUT. An illustrative example is presented in Sect. 4, followed by the presentation of the performance of ARGONAUT on a large set of benchmark problems. We also perform a comparative analysis of the performance of ARGONAUT with existing software for constrained derivative-free optimization. Conclusions and future work are described in Sect. 5.

2 Problem definition

A general Constrained Grey-Box NonLinear optimization problem (CGB-NLP) is defined:

$$\begin{aligned}
&\min_{x} f(\mathbf{x}) \\
&s.t. \quad g_m(\mathbf{x}) \leq 0 \quad \forall m \in \{1, \ldots, M\} \\
&g_k(\mathbf{x}) \leq 0 \qquad\quad \forall k \in \{1, \ldots, K\} \\
&\mathbf{x} \in \Re^C
\end{aligned} \tag{CGB − NLP}$$

where C represents the number of continuous variables with known finite bounds $\left[x_i^L, x_i^U\right]$. K and M represent the total number of constraints with known closed-form and unknown closed-form respectively. It is assumed that the form of the objective function $f(x)$ is unknown since this is the typical case in the vast majority of grey-box optimization applications. Thus, $f(x)$ and $g_m(x)$ are not known explicitly and are approximated by surrogate functions.

3 Constrained grey-box global optimization framework

The main components of the ARGONAUT framework are described in the following subsections and are shown graphically in Fig. 1. A large portion of the algorithm has been introduced in detail in [37], motivated by the optimization of a nonlinear algebraic partial differential system of equations for pressure swing adsorption. In this work, we generalize the methodology and present results for a wide range of test problems with a higher number of dimensions and constraints.

Firstly, the algorithm requires the following inputs: (a) the number of continuous input variables (C), (b) number of unknown constraints (M), (c) the number of known constraints and their analytical form (K), (d) finite bounds on all of the input variables $([x_i^L, x_i^U])$, and (e) a simulation/model which provides the value of all of the unknown functions (constraints and objective) at any given combination of the input variables within $[x_i^L, x_i^U]$. ARGONAUT is developed for problems for which the objective and a subset of, or all of the constraints of the problem do not have analytical forms. The values of the objective and the left hand side of the constraints in set M are provided after a function-call to the simulation/model.

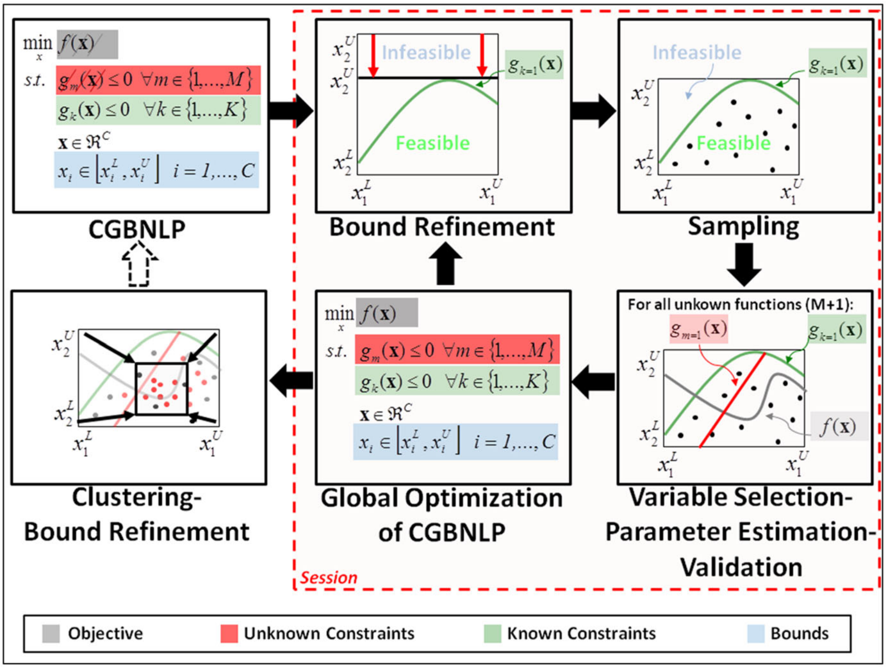
$\min_x f(\mathbf{x})$
s.t. $g_m(\mathbf{x}) \le 0 \;\; \forall m \in \{1,\ldots,M\}$
$g_k(\mathbf{x}) \le 0 \;\; \forall k \in \{1,\ldots,K\}$
$\mathbf{x} \in \Re^C$
$x_i \in [x_i^L, x_i^U] \;\; i = 1,\ldots,C$
CGBNLP
Infeasible
Feasible
$g_{k=1}(\mathbf{x})$
x_2^U
x_2^L
x_1^L
x_1^U
Bound Refinement
Sampling
For all unkown functions (M+1):
$g_{m=1}(\mathbf{x})$
$f(\mathbf{x})$
Variable Selection-Parameter Estimation-Validation
Global Optimization of CGBNLP
Clustering-Bound Refinement
Session
Objective
Unknown Constraints
Known Constraints
Bounds

Fig. 1 Algorithmic building blocks of ARGONAUT

ARGONAUT is an iterative framework, where a single iteration is defined by bounds tightening, sampling, surrogate function selection, global optimization of the CGB-NLP problem and collection of new sampling points in order to repeat the entire procedure until certain convergence criteria are met. Moreover, ARGONAUT has the option of repeating the entire iterative procedure (from initial sampling to convergence) in refined subspaces. We define an entire run of the algorithm within a single bound constrained region as a "session" (Fig. 1).

The following steps are performed during one iteration of ARGONAUT. Firstly, the proposed method exploits any information that can be extracted from provided known information in forms of data or equations. For example, in [37] we showed that prior reduced-order data can be used to select improved sampling sets using scenario reduction. In this work, we have incorporated an optimality-based-bound-tightening step (OBBT), using all of the constraints in the set K. Once the bounds on the input variables are identified, a sampling set is designed using Latin Hypercube sampling. Sampling reduction may be necessary at this point, in order to select an optimal subset of design points within feasible regions. Subsequently ARGONAUT performs variable selection for each of the constraints and the objective function. During this step, a subset of the variables may be eliminated from the prediction model of any output, if it is statistically identified that there is no correlation between the variables and the specific output. Based on the obtained samples, model sparsity, and bounds, the best surrogate function is selected for each of the M unknown constraints and the objective function ($M + 1$) out of a pool of possible surrogate basis function types (Table 1). The selections are performed by solving a series of parameter estimation and cross-validation problems. When this is completed, the algorithm

Table 1 Surrogate function forms incorporated within ARGONAUT

Model type	Model form	Parameters
Linear	$y(\mathbf{x}) = b_0 + \sum_{i=1}^{C} b_i x_i$	b_0, b_i for $i = 1, \ldots, C$
General quadratic	$y(\mathbf{x}) = b_0 + \sum_{i=1}^{C} bl_i x_i + \sum_{i=1}^{C} \sum_{j=1, j \geq i}^{C} bnl_{i,j} x_i x_j$	b_0, bl_i for $i = 1, \ldots, C$ $bnl_{i,j}$ for *i,j=1,...,C,* $j \geq i$
Signomial	$y(\mathbf{x}) = \sum_{k=1}^{S} \left(c_k \prod_{i}^{C} x_i^{\alpha_{k,i}} \right) \quad x_i > 0,$	c_k for $k = 1, \ldots, S$ $a_{k,\iota}$ for $k = 1, \ldots, S, i = 1, \ldots, C$ S: signomial order
Radial basis functions	$y(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \pi_i (\mathbf{x}) + \sum_{n=1}^{N} b_n \varphi(\mathbf{x} - \mathbf{x}_n)$	a_i for $i = 1, \ldots, m$ b_n for $n = 1, \ldots, N$ *N: number of samples,* *m: polynomial function* π *order* φ: *basis type (linear, Gaussian, cubic,multiquadric, see* [27])
Kriging	$y(\mathbf{x}) = \mu + \sum_{n=1}^{N} c_n exp \left[- \sum_{i=1}^{C} \theta_i (x_i - x_i^{(n)})^2 \right]$	μ, θ_i for $i = 1, \ldots, C$ $c_n = f(\theta_i, \mu, \mathbf{x}, y)$

has obtained closed-form expressions representing all of the equations in CGB-NLP. If during this procedure any of the $M + 1$unknown functions is approximated with near perfect cross-validation error, we fix this surrogate model and move this function to the set of known constraints K. This is followed by the repetition of the bounds tightening procedure, in order to potentially reduce the search space even more.

Next, the formulated CGB-NLP problem is optimized using the global optimization solver ANTIGONE [38]. From this global optimization step, we obtain a set of local solutions and a global solution which are the new sampling points of the iteration. In order to validate the predictions of the CGB-NLP formulation, the actual $M + 1$ values of the new solutions are collected from the simulation. At this point a single iteration has been completed and if the convergence criteria are not met the algorithm proceeds to a new iteration. Subsequently, the algorithm performs a series of iterations of bound refining, sampling, parameter estimation and global optimization until one of the convergence criteria is met. Convergence for ARGONAUT is achieved if one of the following criteria is satisfied: (a) no improvement of the incumbent solution over a consecutive set of iterations, (b) the incumbent solution is feasible and all of the unknown functions are predicted with very low cross-validation mean squared error, (c) the maximum number of function calls has been reached.

Once convergence is met and the maximum number of function calls has not been reached, the algorithm analyzes all of the obtained data and performs clustering in order to refine the search space significantly around the best obtained solutions. Once the new bounds have been attained, the framework is repeated within the smaller region (Fig. 1). The individual sections of ARGONAUT are described in detail in the following subsections.

3.1 Bounds tightening

Tight bounds are important in optimization since reduction of the search space can significantly improve performance and speed-up convergence. Bounds tightening is particularly important in grey-box optimization since it can significantly reduce sampling needs as well as the accuracy and predictive ability of surrogate models. ARGONAUT is directly linked with the deterministic global optimization solver ANTIGONE [38], which contains very efficient strategies for tightening of bounds which we aim to employ within our framework.

Initially, ARGONAUT requires the user to provide finite bounds for all of the input variables in order to design the sampling procedure. However, any known constraints (K) of the model are also provided to the framework as part of the CGB-NLP formulation. The set of known constraints may be augmented when a surrogate function is a perfect fit to an unknown function during the algorithmic iterations. ARGONAUT checks whether finite bounds $\left[{x_i}^L, {x_i}^U\right]$ in CGB-NLP can become tighter based on the set of K constraints through optimality-based-bound tightening (OBBT) [39,40]. This is performed at the beginning of the algorithm (if $K > 0$), and it is repeated every time a function is added to set K.

ARGONAUT also performs a different bound tightening technique using clustering information. This is performed after the first session is completed. Specifically, the search space is reduced based on clustering information from all of the obtained samples based on a procedure described in [37]. The aim of this final stage is to refine the final solution by producing higher accuracy models within a local region around the best obtained solution(s). As shown in Fig. 1, each session is defined as a run of ARGONAUT (from initial sampling to convergence). This refinement can continue for multiple sessions until the search space is sufficiently small, and convergence criterion (c) has not been satisfied. In all of the results shown in this work we have set the total number of sessions to two.

3.2 Constrained sampling

Sampling is a very important component of ARGONAUT, since it is desirable to avoid any unnecessary sampling but also design balanced computer experiments that optimally span the investigated region. In ARGONAUT, sampling must be performed subject to any known information and bounds on the variables. These may be linear or non-linear constraints which are in the set of known constraints of the CGB-NLP formulation. At the same time, sampling designs must retain geometrical properties by filling the search space homogeneously. In order to achieve both of the above objectives we use space filling Latin Hypercube designs coupled with scenario reduction [41].

More specifically, sampling is performed in one stage, when set K is empty. In this case, we use a Latin Hypercube design given a predefined number of samples. However, if there is a set of known constraints which must be incorporated within sampling, or any other type of reduced-order model which may allow us to collect a large database of prior samples, ARGONAUT uses a two-stage sampling approach. In the first stage, a Latin Hypercube sample is designed with a very large number of samples within the bound constrained space. In the second stage, sampling reduction is performed by first filtering out any samples which do not satisfy the known constraints, and then selecting an optimal subset of samples based on a probabilistic input-output distance criterion using the approach described in [37–41]. Using this approach, the selected subset is optimally chosen to best represent the full distribution of samples in both the input space as well as the output space.

3.3 Surrogate modeling

One of the central components of ARGONAUT is the development of efficient, well-behaved surrogate parametric models in place of any unknown correlations within the CGB-NLP formulation. However, the quality of the parametric models highly depends on (a) the quality and quantity of the data which is available, (b) the selection of the surrogate basis function, and (c) the quality of the optimal parameters obtained by solving nonlinear optimization problems minimizing deviations of the fitted model to the available data. Existing methods typically pre-postulate a single type of surrogate model and develop techniques which aim to identify optimal points through iterative adaptation of the model when new data is collected [25,27,30]. Certain functions may have better characteristics than others for certain types of problems; however, there is no clear answer in terms of the superiority of one type of approach for a general class of problems. Consequently, there have been recent efforts which incorporate information from multiple types of surrogate models [33,34]. However, in both of the above publications the authors locally optimize the surrogate functions which they develop. We have earlier shown that locating the global optimum of CGB-NLP formulations is important towards the performance and consistency of our method, thus our aim is to develop formulations which can be globally optimized [37].

The selection of a type of surrogate function is a complex task, since it must be flexible enough to provide good approximations for any given data set; however, it must have a form which can be then optimized by existing global optimization solvers when incorporated within the CGB-NLP formulation. Non-interpolating basis functions, such as linear, quadratic and polynomial models are often used to approximate unknown correlations for approximation, optimization and integration in the literature [27,28,42]. In grey-box optimization regression models have often been used, however, their rigidness and inability to adapt to highly nonlinear data which do not follow the trends described by these functions is a major disadvantage. On the other hand, all of the aforementioned non-interpolating techniques have the advantage of leading to forms which can be efficiently handled by existing state-of-the-art global optimization solvers, such as ANTIGONE [38].

The second class of surrogate models consists of purely interpolating functions (e.g., radial basis functions [22,27,30,43] and Kriging [29,44]). Interpolating surrogates have increased flexibility since they are built to exactly predict the sampled data, however, they tend to have an increased number of parameters and typically lead to closed-form equations with many nonconvex terms. The natural arising question is: Is it preferable to use more rigid, inaccurate models to represent unknown correlations to create tractable CGB-NLP formulations which can be globally optimized, or to develop more complex interpolating functions which lead to more accurate representations of the data, however also lead to CGB-NLP formulations that cannot be globally optimized at a reasonable computational cost? The advantages of interpolating functions are indisputable, however, as we push the limits of grey-box optimization towards problems with a large number of variables and constraints, we realize that their computational cost becomes a limiting factor. All of the non-interpolating and interpolating functions that exist in ARGONAUT are shown in Table 1. The ARGONAUT framework is developed to accommodate additional new surrogate functions, depending on the knowledge that is available regarding the objective and the constraints of the problem. The selection is made based on the average cross-validation error. Preference is given to simpler non-interpolating functions- if the associated error does not exceed an accuracy threshold- in order to develop tractable CGB-NLP formulations.

In order to showcase the characteristics of different surrogate functions, we calculate the number of parameters as well as the total number of linear and nonlinear terms of the different types of functions with respect to dimensionality and number of samples (Table 2). Based on the values of Table 2, it becomes clear that the flexibility of interpolating functions comes at a cost, since the number of nonlinear terms is not only a function of dimensionality, but also of the number of interpolated points.

Table 2 Specifications of CGB-NLP formulations based on dimensionality C, number of sampled points N and additional parameters specific to different types of surrogate functions: polynomial order m for radial basis functions, signomial order s

	Linear	General quadratic	Signomial (s order)	Radial basis function (m order, N points)	Kriging (order 0, N points)
Parameters	$C+1$	$1+C(C+1)/2$	$C(s+1)$	$1+mC+N$	$C+1$
Linear terms	C	C	0	C	0
Nonlinear terms	–	$C(C+1)/2$ (*bilinear*, *quadratic*)	$C(s+1)$ (*signomial*)	$(m-1)C+CN$ (*polynomial*, *exponential*)	CN (*exponential*)

3.4 Variable selection, model selection and validation, and global optimization of the CGB-NLP formulation

Model sparsity is particularly important as the dimensionality of the grey-box problem increases. It is unlikely that in high dimensional problems with a large number of grey-box constraints, all variables are present in all of the constraints of the problem. As seen in Table 2, dimensionality plays a significant role towards the complexity of the developed CGB-NLP. Dimensionality is also a limiting factor in the parameter estimation of the problem. Thus, if we can determine *a priori* that a subset of variables does not have an effect on a specific constraint or the objective of the problem, the parameter estimation as well as the final surrogate function for this output can be significantly simplified. Traditional sensitivity analysis techniques have been used in derivative-free optimization for ranking the importance of variables as well as decomposing the effects on the variance of an output [45]. However, such methods increase sampling requirements in order to perform the required analysis. In applications of constrained grey-box optimization, we need to employ methods which do not increase sampling requirements and can be used to make informed decisions only based on existing data.

Sure Independence Screening (SIS) is a methodology introduced to perform screening of important variables with respect to the output predictor. SIS is designed to perform well when there is a limited number of samples compared to the high dimensionality of the problem [46,47]. Upon application of SIS all the important variables survive with probability tending to 1. Using this approach, screening is based on correlation learning; specifically SIS uses marginal information of correlations as a screening criterion. The R package SIS [48] is readily available and it is incorporated within ARGONAUT to perform variable selection prior to the parameter estimation. SIS has been developed and tested for ultrahigh dimensional data with a large percentage of sparsity and provides reliable results when the model is no more than 30 % non-sparse. For this reason, prior to performing parameter estimation based on the data and model selection, the SIS algorithm is used to select a subset of important variables for each of the $M+1$ outputs multiple times, each time using a randomly selected subset of all of the available input-output data. If and only if, the same subset of important variables is selected for all of the repetitions of SIS when using different subset of samples, then we discard the non-important variables with certainty. Despite the 70 % sparsity limitation, we have found that this component is extremely valuable as the dimensionality of the problem increases and there is a high probability that a fraction of the constraints depend on very few variables of the problem. Moreover, the SIS method is computationally efficient and does not add a significant computational cost to the overall methodology.

After variable selection is completed, ARGONAUT proceeds to selecting the best surrogate model for representing each of the $M + 1$ unknown functions. For each of the unknown functions, all of the surrogates in Table 1 are fitted by minimizing the least-squares error of prediction between the sampled data and the predictions. This minimization is performed multiple times (cv times), each time using a random subset of existing samples. The average normalized sum-of-squares error of prediction over the cv repetitions is the cross-validation mean square error (CVMSE) which provides a more reliable approximation of the true error of prediction of a surrogate function at unsampled points. Firstly the non-interpolating functions are fitted and validated. If any of the functions have a CVMSE error lower than a very strict tolerance (i.e., 0.01 %), then ARGONAUT considers this function near-perfect and removes the unknown function from the set of unknown functions to the known set (K). If this condition is satisfied but the CVMSE is still under an acceptable threshold (i.e, 1 %), then the non-interpolating function with the minimum CVMSE is used to approximate the unknown function at the specific iteration. Otherwise, the algorithm proceeds to fit the interpolating functions of Table 1 and selects the one with the least CVMSE error. This selection may be different during different iterations, as more data is being collected and bounds may be refined.

Once all of the surrogate functions have been selected and their parameters are globally optimized the formulated CGB-NLP is optimized using ANTIGONE [38,40,49]. Depending on the complexity of the problem (i.e., multimodality, smoothness, feasible region form), the global optimization of CGB-NLP may provide multiple local solutions and a global solution. Since it is preferable to retain a diversity in sampling in order to avoid premature convergence to a local optimum, ARGONAUT collects all of the unique local solutions and the global solution and proceeds

4 Results

4.1 Illustrative example

In order to demonstrate all steps of ARGONAUT, an illustrative example is introduced. The example is test problem ex8_2_1 [50], and it is a problem for design under uncertainty of a batch plant. The problem has 55 variables and 31 constraints. The objective is nonlinear with 3 exponential terms and 50 linear terms; Constraints 1–6 are linear and sparse; and Constraints 7–31 are nonlinear with exponential terms. The full problem formulation is included as Supplementary Material. Initially, it is assumed that all of the 31 constraints and the objective are unknown. The steps of ARGONAUT for solving this problem are outlined below.

Step 1 Sets $C = 55, M = 31$ and $K = 0$. $N = 251$ initial samples using a Latin Hypercube design are collected.

Step 2 Loop through $M + 1$ unknown functions and perform screening using SIS to identify sparse functions. Constraints 1–6 are identified as sparse constraints. Each of them are a function of only 2 variables out of the total set of 55.

Step 3 Loop through $M + 1$ constraints in order to identify the best surrogate function to approximate each one. Constraints 1–6 are only a function of 2 variables, while all of the other functions are assumed to be a function of 55 variables.

Step 4 Linear functions are identified to perfectly fit constraints 1–6, with a CVMSE less than 0.01 %. These are moved to set K. Set $K = 6, M = 25$. The remaining functions are best approximated by kriging models. The CGB-NLP formulation is now complete, containing 6 linear functions and 26 nonlinear kriging-based functions.

Step 5 Optimality-based-bound-tightening is performed for variables participating in constraints 1–6 subject to known constraints 1–6.

Step 6 CGB-NLP formulation is solved to global optimality using ANTIGONE and multiple new local-global optima are identified. The true values of the constraints and the objective are collected for the new design points, and these are added to the sampling set.

Step 7 Steps 3–6 are performed iteratively, until convergence.

Step 8 Convergence is achieved after collection of 298 points because no improvement in the incumbent objective is achieved after 10 consecutive iterations (Fig. 2). Clustering of the 298 samples is performed and the best cluster bounds are used to refine the search space.

Step 9 All of the above steps are repeated for $C = 55, M = 25$ and $K = 6$ until convergence within the refined search space. ARGONAUT converges after 484 samples, where the best objective value found has a relative error of 0.31 % from the true global optimum (Fig. 2).

4.2 Test problems

We have selected a comprehensive set of test problems to apply the ARGONAUT framework. The problems are collected from GlobalLib and CUTEr libraries with the

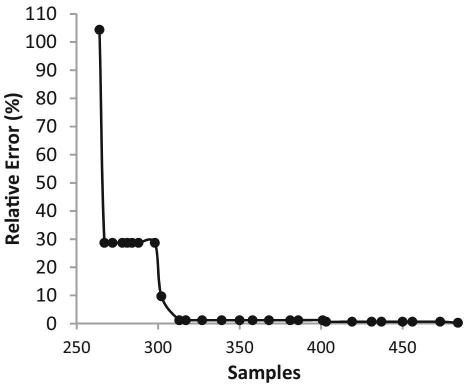

Fig. 2 Relative error of incumbent objective as a function of samples for illustrative example

following characteristics: (a) the dimensionality of the problem must be less or equal to 100 and, (b) the problem must be a NLP problem with no equality constraints. We have collected a total number of 98 problems which fit the above criteria. Currently, ARGONAUT cannot directly handle problems with equality constraints, hence any equality constraints must be transformed to two inequality constraints. The selected upper bound on the dimensionality allows us to solve very challenging problems, which exceeds the typical upper bound of 30 variables for state-of-the-art methods. ARGONAUT can be used to solve problems with even higher number of variables, however, we further want to improve the efficiency of ARGONAUT in terms of sampling and surrogate model approximation in higher dimensions in the future. The statistics of the test problems can be found in the Supplementary Material.

The performance of ARGONAUT is tested for the hardest possible case, namely the case where all of the constraints of the problem are initially assumed to be unknown. The performance of the method, without a doubt, improves if a fraction of these constraints are initially provided because this will reduce the sampling space and improve the grey-box formulation. However, the feature of ARGONAUT to incorporate any known information is tested within our computational results indirectly as follows. During the optimization procedure if the approximation error and cross validation error of any of the unknown constraints is zero, we assume ARGONAUT has identified the form of the unknown constraint(s) and we ‘move’ this constraint to the set of known constraints.

The goal of this study is to investigate the performance of the algorithm, but also the consistency of the method and finally the computational cost of ARGONAUT. Computational cost can be evaluated by both the total CPU time of each run until convergence, but also by the total number of function calls to the input-output model. For all of the test problems the cost of a single function call is negligible, however, in a realistic case study this cost is expected to be significant; imposing constrains on the total number of samples to be obtained in a realistic time frame for optimization.

Finally, we compare the performance of ARGONAUT with three additional methods for constrained non-linear derivative-free optimization. We have restricted our comparative analysis to methods for which an implementation is available and compliant with our computational testing architecture (Linux 64GB/node cluster). The names, references and brief descriptions of these methods are provided in Table 3.

Firstly, it is important to evaluate the average performance of our algorithm, and to analyze the computational cost allocation of ARGONAUT for all of the test prob-

Table 3 Description of constrained derivative-free algorithms used for comparison with ARGONAUT

Name	Description	References
NOMAD	Nonlinear mesh adaptive search: local pattern search (direct or model-based)	[51,52]
ISRES	Improved stochastic ranking evolution strategy: evolutionary search coupled with differential variation	[53,54]
COBYLA	Constrained optimization by linear approximations: local linear approximations of objective and constraints	[54,55]

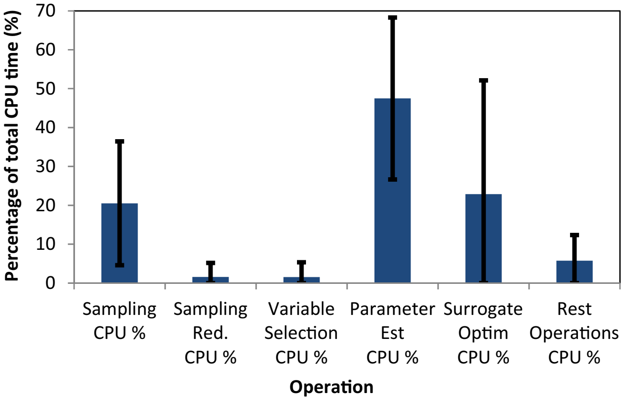

Fig. 3 Breakdown of computational cost of different operations of the ARGONAUT algorithm

lems. The important operations are divided into sampling, sampling reduction, model selection (parameter estimation, cross-validation), global optimization of the CGB-NLP formulation, variable selection and remaining operations (design of experiment, clustering, data exchange e.t.c.). In Fig. 3 the average CPU fraction and the standard deviation of all the operations for all of the test problems is shown. The stage of parameter estimation is the most intensive step of ARGONAUT. This was expected, since this procedure must be performed for all of the unknown functions of the problem. This is a stage that can be parallelized in the future since the parameter estimation problems for each $M+1$ functions are independent. The second most intensive operation is that of global optimization of the CGB-NLP problem, which is a critical step of our algorithm. The large variability of this stage can be attributed to the large range of problems, their complexity, the number of nonconvex terms and dimensionality of the problem. Finally sampling holds the third place in terms of computational cost, however, the fraction of this stage is likely to change significantly based on the computational cost of the application. The sampling stage cost can be reduced by employing parallelization techniques.

There are three aspects of a grey-box algorithm that must be assessed in order to evaluate its applicability. Firstly, the quality of solutions obtained and the variability of the solution when a different sampling set is used. Secondly, the number of function calls which the algorithm requires to recognize that it should converge or stop. This is a critical aspect, especially in expensive simulation applications. Lastly, it is important to compare the total CPU cost of an algorithm until convergence.

We have collected results for all of the test problems and methods for 10 repetitions of each test problem, starting from randomly different sampling designs or initial points. In terms of quality, ARGONAUT is the only method which manages to find feasible solutions for all of the problems over all of the repetitions performed. This is an extremely important feature and it is due to the fact that the entire space is explored instead of local searches around a starting point. The average performance is shown through the mean absolute error between the final obtained optimum and the actual global optimum normalized by the global optimum when it is not equal to zero $(MAE = 100\left|\left(f_{best} - f_{glob}\right)/f_{glob}\right|$. It is important to evaluate the average MAE but also the standard deviation of the MAE over the multiple repetitions of the algorithmic framework. Tables 4 and 5 show the average and the standard deviation of the absolute errors as a function of dimensionality and constraints, respectively. Any values that are not reported in Tables 4 and 5 are due to the fact that feasible solutions where not obtained in the majority of the repetitions- thus the standard deviation of the error could not be calculated. Moreover, an MAE of 100 is assigned to problems for which all runs did not lead to a feasible optimum solution.

The results in both tables show that ARGONAUT has an overall better average performance when compared to other methods, since the best solutions found are on average closest to the true global solutions. In Table 4 we observe 6 cases (2, 3, 4, 7, 13, 40 variables) where all ARGONAUT, COBYLA and NOMAD locate the true global optimum consistently since the average error is almost zero, however COBYLA and NOMAD have a lower average MAE. This increased accuracy can be explained by their detailed local search strategies, which also increase the sampling cost as it will be seen in the analysis below. Based on Table 4, we can observe a trend of an increasing average MAE as the dimensionality of the problem increases, however, the same conclusion is not true when the number of constraints increases. It is also important to point out that the superiority of ARGONAUT becomes more obvious as the dimensionality and the number of constraints of the problems increase. In these cases, other methods often cannot to locate a feasible solution.

It is important to note that there is some variability in the error observed in all of the methods, since in grey-box optimization all methods suffer from the variability introduced by the randomness of the initial design or the initial point. However, ARGONAUT in general suffers the least from this challenge, providing on average good solutions with increased reliability. It should be noted that the average standard deviation should be assessed along with the average best found error, since it is important for both metrics to be low. There are 3 out of 21 and 2 out of 22 cases where NOMAD and COBYLA outperform ARGONAUT in terms of both quality and reliability of solution, however, as we will see from the following analysis this increase in reliability comes at a significant sampling cost. This is the reason why the competing methods perform poorly as the dimensionality of the problems increase and it becomes more difficult to search the space by collecting a large amount of samples (curse of dimensionality).

Figure 4a depicts the average total number of samples required for each algorithm to recognize convergence. Undoubtedly, ARGONAUT reaches convergence with the least number of samples than any other algorithm for the vast majority of problems. NOMAD has more comparable sampling requirements. COBYLA has comparable sampling requirements only for smaller problems, while sampling increases significantly as the number of dimensions and constraints increases. ISRES requires a significant number of samples for all of the test problems. It must be mentioned at this point that the increased sampling requirements for COBYLA and ISRES can be attributed to their implementation which requires the constraint violation output values and the objective to be provided as different functions. This is an advantageous implementation in the chase where feasibility must be evaluated separately from the evaluation of the objective. However, this may cause the simulation to be called multiple times for the same sample, increasing the sampling requirements.

Table 4 Average MAE and standard deviation of MAE as a function of the dimensionality of the problems

C	Average best found MAE (%)				Standard deviation of MAE (%)			
	ARGONAUT	NOMAD	COBYLA	ISRES	ARGONAUT	NOMAD	COBYLA	ISRES
1	0.000	0.000	0.000	0.004	8.001	0.691	31.651	0.203
2	0.002	0.000	0.000	4.034	5.345	1.275	97.745	5.810
3	0.002	9.377	0.000	18.855	6.977	0.308	–	67.347
4	0.021	18.230	0.000	46.901	20.719	0.431	288.264	272.649
5	0.000	0.000	0.000	42.338	13.486	1.267	5.818	7.117
6	0.000	0.001	0.000	28.852	45.356	56.935	76.169	13.807
7	0.030	2.808	0.000	14.073	30.226	266.450	15.527	15.642
8	0.001	40.737	2.244	26.273	1.368	1.706	42.449	15.209
10	0.000	5.373	35.883	84.926	0.243	22.130	–	–
11	0.000	1.257	6.122	5.596	0.000	5.459	179.146	65.749
13	0.018	50.221	0.000	36.215	1.444	0.941	11.690	6.468
20	3.636	49.045	23.876	62.182	9.670	–	–	–
24	2.109	100	23.749	100	8.644	–	51.144	21.152
30	0.000	58.688	0.000	73.886	2.169	–	107.472	37.963
40	0.003	30.606	0.000	47.383	0.352	3.345	19.161	4.721
50	0.000	66.292	0.000	26.677	0.141	4.715	15.296	8.653
55	1.252	99.499	0.895	100	5.579	–	137.612	344.478
59	2.209	100	1.571	0.531	2.441	–	9.310	15.543
65	0.363	4.326	100	3.954	0.430	9.923	3.282	39.462
84	3.400	100	100	100	12.302	–	7.519	–
100	5.600	100	100	100	30.300	–	–	–

Table 5 Average MAE and standard deviation of MAE as a function of the number of constraints of the problems

M	Average best found MAE (%)				Standard deviation of MAE (%)			
	ARGONAUT	NOMAD	COBYLA	ISRES	ARGONAUT	NOMAD	COBYLA	ISRES
1	0.006	0.000	0.000	16.580	7.300	0.579	20.884	2.193
2	0.024	0.053	0.000	13.725	7.215	2.064	75.824	209.780
3	0.000	0.000	0.000	3.020	0.000	0.212	27.959	3.731
4	0.002	12.229	0.119	28.879	7.627	32.596	459.383	68.737
5	0.000	2.771	0.641	26.354	3.372	7.726	809.939	30.513
6	0.000	50.000	1.532	11.250	16.668	3.148	88.203	40.803
7	0.000	0.009	0.000	79.971	20.058	1.249	75.768	47.345
8	0.000	0.002	0.000	18.891	0.000	0.008	1339.560	118.912
9	0.000	0.221	0.000	34.403	0.000	0.470	21.243	5.579
10	0.000	25.061	27.927	68.830	9.945	–	–	–
11	0.000	32.021	7.903	47.251	0.000	–	33.117	17.873
12	0.182	2.163	50.000	7.798	0.215	5.212	307.362	30.801
13	2.800	51.523	75.571	100.000	8.932	179.877	–	–
14	0.030	2.808	0.000	14.073	30.226	266.450	15.527	15.642
20	0.422	44.773	4.750	50.414	1.173	4.679	20.891	9.974
21	0.000	100.000	0.000	100.000	0.000	–	202.169	70.232
22	20.000	50.628	53.061	52.798	0.000	–	–	–
24	2.209	100.000	1.571	0.531	2.441	–	9.310	15.543
30	0.035	100.000	0.000	6.950	2.888	–	14.000	7.208
31	0.344	98.998	0.001	100.000	5.631	–	0.521	371.703
42	3.400	100.000	100.000	100.000	47.326	–	7.519	–
81	2.160	100.000	1.789	100.000	5.528	–	274.703	317.254

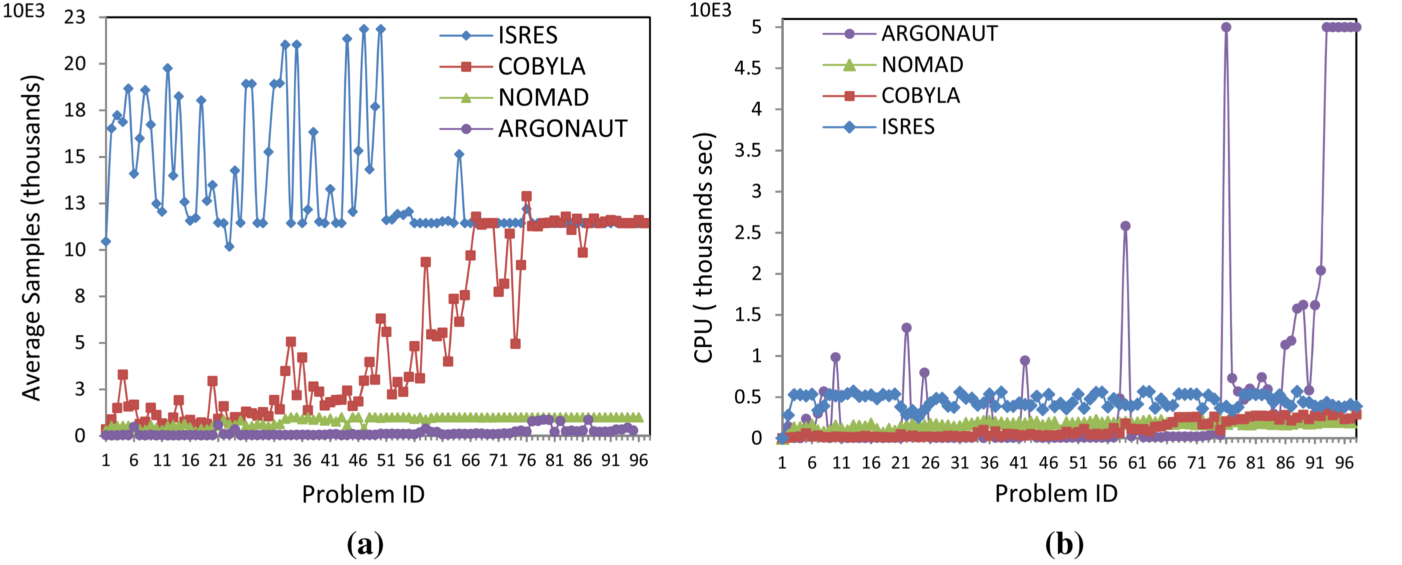

Fig. 4 **a** Average total number of samples collected, **b** average CPU required, for algorithm convergence

On the other hand, Fig. 4b shows the average total CPU time required by each algorithm to reach convergence, where ARGONAUT performs well for the majority of problems, but requires more computational time than other algorithms for certain problems. Looking closely at the results, these problems are the ones with complex nonconvex terms (i.e., exponential, fractional, polynomial, trigonometric) for which interpolating functions are employed leading to very complex CGB-NLP problems. Even though the computational cost is increased, the performance of ARGONAUT on these complex problems is better than that of the competitive methods which often fail to find feasible solutions if a bad initial point is provided.

5 Conclusions

We have presented a systematic study for coupling surrogate model identification with global optimization through the developed ARGONAUT algorithm. This methodology aims to solve problems with a large size of variables and constraints where a subset of the set of constraints and the objective function are not known in closed-form. Our approach uses concepts from global optimization such as optimality based bounds tightening, variable screening, and a novel strategy for constrained optimized sampling. The performance of ARGONAUT on a large set of benchmark problems for constrained global optimization with up to 100 variables and up to 81 constraints is promising, since it manages to identify the global solution with fewer samples than existing derivative-free optimization solvers for constrained optimization. The higher computational cost of ARGONAUT is attributed to the need to solve multiple global optimization problems for parameter estimation and optimization of the formulated surrogate CGB-NLP problems. However, obtaining global solutions of the CGB-NLP problem is also one of the main reasons for the increased reliability of the method. In fact, as the cost of the function call increases, the relative cost of global optimization will decrease significantly. The performance of ARGONAUT has a reduced variability in its performance since it does not depend on a single initial point; however we have observed that for highly non-linear problems with large ranges in their input variables there is still a small amount of variability on the obtained solutions depending on the initialization of the sampling set.

Acknowledgements The authors acknowledge financial support from the National Science Foundation (CBET-0827907, CBET-1263165).

References

1. Audet, C., Bechard, V., Chaouki, J.: Spent potliner treatment process optimization using a MADS algorithm. Optim. Eng. **9**(2), 143–160 (2008)
2. Bartholomew-Biggs, M.C., Parkhurst, S.C., Wilson, S.P.: Using DIRECT to solve an aircraft routing problem. Comput. Optim. Appl. **21**(3), 311–323 (2002)
3. Boukouvala, F., Ierapetritou, M.G.: Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing. J. Pharm. Innov. **8**(2), 131–145 (2013)
4. 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)
5. Egea, J.A., Rodriguez-Fernandez, M., Banga, J.R., Marti, R.: Scatter search for chemical and bio-process optimization. J. Global Optim. **37**(3), 481–503 (2007)

6. 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)
7. Fowler, K.R., Reese, J.P., Kees, C.E., Dennis Jr., J.E., Kelley, C.T., Miller, C.T., Audet, C., Booker, A.J., Couture, G., Darwin, R.W., Farthing, M.W., Finkel, D.E., Gablonsky, J.M., Gray, G., Kolda, T.G.: Comparison of derivative-free optimization methods for groundwater supply and hydraulic capture community problems. Adv. Water Resour. **31**(5), 743–757 (2008)
8. Graciano, J.E.A., Roux, G.A.C.L.: Improvements in surrogate models for process synthesis. Application to water network system design. Comput. Chem. Eng. **59**, 197–210 (2013)
9. Hemker, T., Fowler, K., Farthing, M., Stryk, O.: A mixed-integer simulation-based optimization approach with surrogate functions in water resources management. Optim. Eng. **9**(4), 341–360 (2008)
10. Henao, C.A., Maravelias, C.T.: Surrogate-based superstructure optimization framework. AIChE J. **57**(5), 1216–1232 (2011)
11. Kleijnen, J.P.C., van Beers, W., van Nieuwenhuyse, I.: Constrained optimization in expensive simulation: novel approach. Eur. J. Oper. Res. **202**(1), 164–174 (2010)
12. Wan, X.T., Pekny, J.F., Reklaitis, G.V.: Simulation-based optimization with surrogate models—application to supply chain management. Comput. Chem. Eng. **29**(6), 1317–1328 (2005)
13. Espinet, A., Shoemaker, C., Doughty, C.: Estimation of plume distribution for carbon sequestration using parameter estimation with limited monitoring data. Water Resour. Res. **49**(7), 4442–4464 (2013)
14. Hasan, M.M.F., Baliban, R.C., Elia, J.A., Floudas, C.A.: Modeling, simulation, and optimization of postcombustion CO_2 capture for variable feed concentration and flow rate. 2. Pressure swing adsorption and vacuum swing adsorption processes. Ind. Eng. Chem. Res. **51**(48), 15665–15682 (2013)
15. Hasan, M.M.F., Boukouvala, F., First, E.L., Floudas, C.A.: Nationwide, regional, and statewide CO_2 capture, utilization, and sequestration supply chain network optimization. Ind. Eng. Chem. Res. **53**(18), 7489–7506 (2014)
16. Li, S., Feng, L., Benner, P., Seidel-Morgenstern, A.: Using surrogate models for efficient optimization of simulated moving bed chromatography. Comput. Chem. Eng. **67**, 121–132 (2014)
17. Forrester, A.I.J., Sóbester, A., Keane, A.J.: Engineering Design via Surrogate Modelling—A Practical Guide. Wiley, Chichester (2008)
18. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MPS-SIAM Series on Optimization, vol. 8. SIAM, Philadelphia (2009)
19. 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)
20. Rios, L.M., Sahinidis, N.V.: Derivative-free optimization: a review of algorithms and comparison of software implementations. J. Global Optim. **56**(3), 1247–1293 (2013)
21. 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**, 701–727(2016) (in press). doi:10.1016/j.ejor.2015.12.018
22. Bjorkman, M., Holmstrom, K.: Global optimization of costly nonconvex functions using radial basis functions. Optim. Eng. **1**(4), 373–397 (2000)
23. Booker, A.J., Dennis, J.E., Frank, P.D., Serafini, D.B., Torczon, V., Trosset, M.W.: A rigorous framework for optimization of expensive functions by surrogates. Struct. Multidiscip. Optim. **17**(1), 1–13 (1999)
24. 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)
25. 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)
26. Forrester, A.I.J., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. **45**(1), 50–79 (2009)
27. Jakobsson, S., Patriksson, M., Rudholm, J., Wojciechowski, A.: A method for simulation based optimization using radial basis functions. Optim. Eng. **11**(4), 501–532 (2010)
28. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. J. Global Optim. **21**(4), 345–383 (2001)
29. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. J. Global Optim. **13**(4), 455–492 (1998)
30. 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)

31. Regis, R.G., Shoemaker, C.A.: Constrained global optimization of expensive black box functions using radial basis functions. J. Global Optim. **31**(1), 153–171 (2005)
32. Yao, W., Chen, X.Q., Huang, Y.Y., van Tooren, M.: A surrogate-based optimization method with RBF neural network enhanced by linear interpolation and hybrid infill strategy. Optim. Methods Softw. **29**(2), 406–429 (2014)
33. Muller, J., Shoemaker, C.A.: Influence ensemble surrogate models and sampling strategy on the solution quality of algorithms for computationally expensive black-box global optimization methods. J. Glob. Optim. **60**(2), 123–144 (2014)
34. Viana, F.A.C., Haftka, R.T., Watson, L.T.: Efficient global optimization algorithm assisted by multiple surrogate techniques. J. Glob. Optim. **56**(2), 669–689 (2013)
35. Floudas, C.A.: Deterministic Global Optimization, vol. 37. Springer Science & Business Media, New York (1999)
36. Floudas, C.A., Gounaris, C.E.: A review of recent advances in global optimization. J. Global Optim. **45**(1), 3–38 (2009)
37. Boukouvala, F., Hasan, M.M.F., Christodoulos, A.F.: Global optimization of constrained grey-box problems: new method and application to constrained PDEs for pressure swing adsorption. J. Global Optim. (2015). doi:10.1007/s10898-015-0376-2
38. Misener, R., Floudas, C.: ANTIGONE: algorithms for continuous/integer global optimization of nonlinear equations. J. Global Optim. **59**(2–3), 503–526 (2014)
39. Maranas, C.D., Floudas, C.A.: Finding all solutions of nonlinearly constrained systems of equations. J. Global Optim. **7**(2), 143–182 (1995)
40. Misener, R., Floudas, C.A.: GloMIQO: global mixed-integer quadratic optimizer. J. Global Optim. **57**(1), 3–50 (2013)
41. Li, Z., Floudas, C.A.: Optimal scenario reduction framework based on distance of uncertainty distribution and output performance: I. Single reduction via mixed integer linear optimization. Comput. Chem. Eng. **70**, 50–65 (2014)
42. Powell, M.J.D.: Approximation Theory and Methods. Cambridge University Press (1981)
43. Holmstrom, K., Quttineh, N.-H., Edvall, M.M.: An adaptive radial basis algorithm (ARBF) for expensive black-box mixed-integer constrained global optimization. Optim. Eng. **9**(4), 311–339 (2008)
44. Kleijnen, J.P.C.: Kriging metamodeling in simulation: a review. Eur. J. Oper. Res. **192**(3), 707–716 (2009)
45. Adjengue, L., Audet, C., Ben Yahia, I.: A variance-based method to rank input variables of the mesh adaptive direct search algorithm. Optim. Lett. **8**(5), 1599–1610 (2014). doi:10.1007/s11590-013-0688-4
46. Fan, J., Lv, J.: Sure independence screening for ultrahigh dimensional feature space. J. R. Stat. Soc. Ser. B (Statistical Methodology) **70**(5), 849–911 (2008). doi:10.1111/j.1467-9868.2008.00674.x
47. Fan, J., Song, R.: Sure independence screening in generalized linear models with NP-dimensionality. Ann. Stat. **38**(6) 3567–3604 (2010). doi:10.1214/10-AOS798
48. Fan, J., Feng, Y., Saldana, D.F., Samworth, R., Wu, Y.: Sure independence screening version 0.7-5 (2015). https://cran.r-project.org/web/packages/SIS/index.html
49. Misener, R., Floudas, C.A.: Global optimization of mixed-integer models with quadratic and signomial functions: a review. Appl. Comput. Math. **11**, 317–336 (2012)
50. Floudas, C.A., Pardalos, P.M., Adjiman, C.S., Esposito, W.R., Gumus, Z.H., Harding, S.T., Klepeis, J.L., Meyer, C.A., Schweiger, C.A.: Handbook of Test Problems in Local and Global Optimization. Kluwer Academic Publishers, Berlin (1999)
51. Abramson, M.A., Audet, C., Couture, G., Dennis, J.E., S, Tribes, C.: The NOMAD Project (2009). https://www.gerad.ca/nomad
52. Le Digabel, S.: Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm. ACM Trans. Math. Softw. (TOMS) **37**(4), 1–15 (2011)
53. Runarsson, T.P., Xin, Y.: Search biases in constrained evolutionary optimization. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **35**(2), 233–243 (2005). doi:10.1109/TSMCC.2004.841906
54. Johnson, S.G.: The NLopt nonlinear-optimization package (2015). http://ab-initio.mit.edu/nlopt
55. 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. Mathematics and Its Applications, vol. 275, pp. 51–67. Springer, Netherlands (1994)