Computers and Chemical Engineering 116 (2018) 306–321

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

# A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point

Ishan Bajaj, Shachit S. Iyer, M.M. Faruque Hasan*

*Artie McFerrin Department of Chemical Engineering, Texas A&M University, 3122 TAMU, College Station, TX 77843, USA*

ARTICLE INFO

*Article history:*
Received 17 September 2017
Revised 1 December 2017
Accepted 22 December 2017
Available online 24 December 2017

*Keywords:*
Constrained derivative-free optimization
Black-box optimization
Grey-box optimization
Simulation-based optimization
Surrogate model
Data-driven optimization

ABSTRACT

This paper presents an algorithm for constrained black-box and grey-box optimization. It is based on surrogate models developed using input-output data in a trust-region framework. Unlike many current methods, the proposed approach does not require feasible initial point and can handle hard constraints via a novel optimization-based constrained sampling scheme. A two-phase strategy is employed, where the first phase involves finding feasible point through minimizing a smooth constraint violation function (feasibility phase). The second phase improves the objective in the feasible region using the solution of the feasibility phase as starting point (optimization phase). The method is applied to solve 92 test problems and the performance is compared with established derivative-free solvers. The two-phase algorithm outperforms these solvers in terms of number of problems solved and number of samples used. We also apply the algorithm to solve a chemical process design problem involving highly-coupled, nonlinear algebraic and partial differential equations.

Published by Elsevier Ltd.

## 1. Introduction

In this work, we present an optimization algorithm for problems for which the objective and one or more constraints are evaluated using simulations or proprietary codes. The optimization problems where the analytical form of the objective function and all the constraints are not available are referred as black-box problems. When the analytical form of the objective function is unknown but the analytical form of one or more constraints is known, then such problems are called grey-box problems. In the most general form, these problems can be represented as follows:

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & g_i(x) \le 0 \qquad \forall i \in \{1, \cdots, p\} \\ & g_j(x) \le 0 \qquad \forall j \in \{1, \cdots, q\} \\ & x \in \left[x^L, x^U\right] \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ are the decision variables, $f(x) : \mathbb{R}^n \to \mathbb{R}$ and $g_i(x) : \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, \ldots, p\}$ are $C^1$ functions with hidden expressions in a black-box simulator. However, $g_j(x) : \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, \ldots, q\}$ are functions with known algebraic expressions.

Many engineering problems require to optimize the decision variables such that a certain objective, such as cost, is minimized while satisfying some system requirements. If the system is modeled using a set of algebraic equations, optimization can be performed using existing local and global solvers. However, in many cases, an algebraic model may not be available or is not adequate to allow for detailed optimization. A more realistic model that accounts for the spatial and temporal variation of the system properties could be given by a set of partial differential equations (PDEs).

Although PDE models have higher predictive ability, rigorous optimization using PDE models remains a challenge. One of the approaches investigated in the literature is to discretize the PDEs and convert into algebraic equations. The resulting optimization model then becomes a large-scale nonlinear programming problem. For instance, Nilchan and Pantelides (1998) optimized a periodic adsorption process by applying a sequential quadratic programming algorithm to the discretized equations. Biegler and coworkers (Kawajiri and Biegler (2006), Agarwal et al. (2010)) applied IPOPT (Wächter and Biegler, 2006) to optimize the discretized model of a adsorption based process. Although promising results were obtained, the level of discretization used in these works were moderate to keep the nonlinear model tractable and by doing so, a certain degree of accuracy was lost. An alternative and promising optimization strategy is to use simulation data to optimize while maintaining the high level of accuracy of the model. Of course,

* Corresponding author.
*E-mail address:* hasan@tamu.edu (M.M. Faruque Hasan).

https://doi.org/10.1016/j.compchemeng.2017.12.011
0098-1354/Published by Elsevier Ltd.

using high number of discretization will increase the computational time of the simulation. Therefore, the challenge is to obtain the optima using few number of evaluations. Such class of optimization problems where only simulation/experimental data are available are black-box problems and the algorithms to solve such problems are referred as derivative-free optimization (DFO) methods (Conn et al. (2009b), Lewis et al. (2000), Koziel et al. (2011), Abramson et al. (2009), Nesterov et al. (2011), Yang et al. (2012), Regis and Shoemaker (2013), Sergeyev and Kvasov (2017)).

DFO methods is also increasingly being applied to problems in physics, chemistry, finance, computer science and operations research. In process systems engineering, the use of data-driven models and DFO techniques is steadily increasing (Audet et al. (2008), Fahmi and Cremaschi (2012), Graciano and Le Roux (2013), Caballero and Grossmann (2008), Henao and Maravelias (2011), First et al. (2014), Eason and Biegler (2016), Won and Ray (2005), Meert and Rijckaert (1998), Fernandes (2006), Palmer and Realff (2002), Jia et al. (2009), Rogers and Ierapetritou (2015)). Rios and Sahinidis (2013) provided an extensive comparison of the performance of different DFO solvers. Recent review papers by Garud et al. (2017) and Bhosekar and Ierapetritou (2017) provide excellent overview of different sampling strategies and surrogate models, respectively.

Derivative-free optimization algorithms can be classified into direct-search, model-based or hybrid methods. In direct-search algorithms, the decision for the samples to be evaluated at next steps is based on the previously evaluated points and the best solution so far. Model-based algorithms attempt to capture the curvature of the original model using data-driven surrogate models and the search process is guided by optimizing the surrogate model. Hybrid approaches use models in a direct-search framework. Generally, direct-search methods are more successful for non-smooth problems while model-based methods are preferred for smooth problems.

Although there are many available algorithms and solvers for unconstrained and box-constrained black-box problems (Abramson et al. (2011), Custódio and Vicente (2007), Wild et al. (2008), Huyer and Neumaier (2008), Powell (2009), Vaz and Vicente (2007), Hansen (2016), Dakota (2009), Gilmore and Kelley (1995)), there has been limited development to handle black-box constrained problems.

One of the approaches to handle constraints is to transform the original constrained problem into an unconstrained problem using exact penalty method (Liuzzi and Lucidi, 2009). A line-search method is applied to optimize the resulting problem. However, the penalty parameter needs to be chosen carefully such that it is smaller than a certain threshold that is not known *a priori*. Liuzzi et al. (2010) defined a merit function and applied a sequential penalty approach to handle constraints. The penalty parameter update rule was connected to the sampling technique to ensure convergence. An alternative approach proposed to handle constraints is by employing Augmented Lagrangian approach (Diniz-Ehrhardt et al., 2011). This method formulates a shifted penalty function that includes black-box constraints in the objective function. This results in a black-box bound constrained problem for which both model-based and direct-search approaches were applied. Mesh adaptive direct-search (MADS) was used to treat constraints directly using an extreme barrier (Audet and Dennis Jr, 2006) or a progressive barrier approach (Audet and Dennis Jr, 2009). The extreme barrier based method rejects all infeasible points, while the progressive barrier method reduces the threshold on constraint violation iteratively such that the iterates move towards feasible region. Filter method was applied along with direct-search to solve constrained black-box problems (Audet and Dennis Jr, 2004). Filter method consider a constrained problem as bi-objective and compares the trade-offs between objective function and constraint violation (Fletcher and Leyffer, 2002) by building a pareto front. A similar approach along with linear models was applied in Brekelmans et al. (2005). Eason and Biegler (2016) developed an algorithm for black box/glass box problems by approximating the black box model a using kriging surrogate model and applied it to solve process optimization problems.

COBYLA (Powell, 1994a) was one of the first model-based approaches for solving constrained black-box problems. In COBYLA, linear models for the objective function and constraints are developed using interpolation points on the vertices of a simplex and the optimization is performed in a trust-region framework. Audet et al. (2016) employed quadratic polynomials to approximate the objective function and constraints and progressive barrier approach is applied to handle constraints. Sampaio and Toint (2015) extended the trust-funnel method presented in Gould and Toint (2010) to constrained black-box problems. Each iteration in trust-funnel algorithm is composed of two steps. The first step reduces the infeasibility and the second step aims to improve the optimality. The bound on allowed infeasibility is decreased monotonically with iterations so that the final point is feasible. Augustin and Marzouk (2014) developed NOWPAC based on a trust-region framework. An offset function is added to the constraints to convexify the local feasible region and facilitate the generation of feasible trial points. However, NOWPAC requires the initial point to be feasible. Radial basis functions are used as surrogate models to approximate the objective function and constraints in CONORBIT (Regis and Wild, 2017). Boukouvala and Floudas (2017) developed ARGONAUT that applies bound tightening and variable selection strategies and use global optimization solver while optimizing the subproblems. Arouxét et al. (2015) proposed an inexact restoration method employing polynomial models and used merit function to measure the progress of the algorithm.

In spite of the advances made in derivative-free optimization, there are a lot of challenges that still need to be addressed. Many of the current available algorithms require a feasible initial point. Some of them also assume that all the constraints are black-box. However, in many practical applications, some of the constraints are known and critical problem insights are neglected by treating them as black-box. They also do not offer an effective strategy to handle hard constraints. Violation of the hard constraints lead to simulation failure. In this work, we propose an optimization algorithm for constrained black-box and grey-box systems. Our proposed algorithm has two phases: (i) feasibility phase, and (ii) optimization phase. The feasibility phase finds a feasible point while the optimization phase attempts to decrease the objective function. Both phases of the algorithm are based on developing and optimizing surrogate models in a trust region framework. The two phase algorithm does not require a feasible initial point. A novel optimization-based sampling strategy is also proposed which can handle hard constraints effectively so that the samples generated are feasible with respect to these constraints. The efficiency of the algorithm is demonstrated by applying on a set of 92 nonlinear test problems from GlobalLib and a chemical engineering case study. The algorithm is also compared to NOMAD (Abramson et al. (2011), Le Digabel (2011)) and COBLYA (Powell (1994a)), two widely used solvers for constrained black-box problems.

The paper is organized as follows. Notations and pertinent fundamental concepts are given in Section 2. Section 3 provides the details of the algorithm while Section 4 describes the numerical test problems and the chemical engineering case study. Some conclusions are made in Section 5.

## 2. Preliminaries

### 2.1. Notation

If a vector $s \in \mathbb{R}^n$ is given, a subscript will be used to represent an element of sequence of vectors ($s_k$). The $i$th component of a vector $s$ is represented by ($s_i$). Note that, in order to avoid any ambiguity in representing component of a vector and sequence of vectors, we will use $i$ as the subscript whenever we refer to the component of a vector. The superscript is used ($s^j$) to denote the $j$th element of a finite set of vectors. The norm ||.|| is generally $L^1$-norm unless otherwise specified.

### 2.2. Trust region framework

In a trust-region method (Conn et al., 2000), a surrogate model is typically constructed in the neighborhood of a point $x_k$ such that the model is believed to be an adequate representation of the original function in the region around $x_k$. The trust-region is defined to be the set of points around a point $x_k$ such that:

$$B(x_k, \Delta_k) = \{x \in \mathbb{R}^n : ||x - x_k|| \leq \Delta_k\}$$

When the analytical form of the objective is known, the model $f^{dr}$ is considered quadratic such that it incorporates the first and second order derivative information. The form of the model is as follows:

$$f^{dr}(x) = f(x_k) + \nabla^T f(x_k)x + x^T \nabla^2 f(x_k)x$$

A new point, $\bar{x}$ is obtained by solving a subproblem of the form:

$$\begin{aligned} &\min_x \quad f^{dr}(x) \\ &s.t. \quad x \in B(x_k, \Delta_k) \end{aligned} \tag{2}$$

It is not even necessary to solve Problem (2) exactly but a point $\bar{x}$ providing sufficient decrease in the model is admissible. For unconstrained optimization problems with known derivative information of the objective function, the decisions about moving trust-region center and changing size are based on the ratio of actual reduction in the objective function to the predicted reduction, i.e.

$$\rho_k = \frac{f(x_k) - f(\bar{x})}{f^{dr}(x_k) - f^{dr}(\bar{x})} \tag{3}$$

Given the parameters $0 \leq \eta_0 \leq \eta_1 < 1$, $0 < \gamma_{de} < 1$ and $\gamma_{in} > 1$, the trust-region size is decided by the following rule:

$$\Delta_{k+1} = \begin{cases} \gamma_{in}\Delta_k, & \text{if } \rho_k \geq \eta_1 \\ \Delta_k, & \text{if } \eta_0 \leq \rho_k < \eta_1 \\ \gamma_{de}\Delta_k, & \text{if } \rho_k < \eta_0 \end{cases} \tag{4}$$

The trust-region center is updated as follows:

$$x_{k+1} = \begin{cases} \bar{x} & \text{if } \rho_k \geq \eta_0 \\ x_k, & \text{if } \rho_k < \eta_0 \end{cases} \tag{5}$$

The size of the trust-region is increased if $\rho_k$ is appreciable and it is decreased otherwise. The algorithm converges to a second-order critical point ($x^*$) when at each iteration, (2) is solved approximately and the trust-region center and radius are updated as per the rules given by Eqs. (4) and (5). It is always possible in derivative-based methods to decrease the trust-region sufficiently and obtain a better objective function value unless the trust-region center is an optimal point. However, critical information is lost when the derivatives are unavailable. For optimization problems with the absence of first-order and second-order derivative information, input-output data are used to develop surrogate models. In DFO, the trust-region not only restricts the step length to the region where model is considered appropriate but also forms the basis for choosing samples to build data-driven models. In this case, the trust-region size is not the only factor that prevents us to obtain a better objective function. The models need to be sufficiently accurate before the decision to decrease the trust-region is taken. Conn et al. (2008) formalized the concept of accuracy by defining fully linear and fully quadratic models. Ensuring fully quadratic models require the original function to be twice continuously differentiable function. In this work, we assume the function to be once continuously differentiable and therefore focus on fully linear class of models.

**Definition 2.1.** Assume that the function, $f(x)$ and its data-driven surrogate model, $f^r(x)$ are continuously differentiable and $\nabla f$ and $\nabla f^r$ are Lipschitz continuous. $f^r(x)$ is said to be fully linear if the error between the gradient of the model and the function is bounded such that

$$||\nabla f(x) - \nabla f^r(x)|| \leq \kappa_{df}\Delta \quad \forall x \in B(x_k, \Delta) \tag{6}$$

And also, the error between the function and model satisfy the following relationship:

$$|f(x) - f^r(x)| \leq \kappa_f \Delta^2 \quad \forall x \in B(x_k, \Delta) \tag{7}$$

where the constants $\kappa_{df}$, $\kappa_f > 0$ are dependent on the Lipschitz constants of the surrogate model and the function, and a constant $\Lambda$ that measures the poisedness of the interpolation points (Conn et al., 2008). Note that the parameters that can be controlled are, $\Lambda$ and the trust-region size, $\Delta$. The relations (6) and (7) indicate that the gradient of the function and the function itself can be approximated well if the trust-region size is small and the samples are well poised. If the sample set is well poised, the model is said to be fully linear. In other words, if the surrogate model is not able to obtain a better point, consideration should be given on constructing accurate models to estimate the direction of decrease. If the model is accurate and the trust-region is sufficiently small, a better objective function will be obtained. These bounds are similar to those of Taylor models and are essential to ensure convergence. The update rule for trust-region center remains the same but the criteria for updating the trust-region size is given as follows:

$$\Delta_{k+1} = \begin{cases} \gamma_{in}\Delta_k, & \text{if } \rho_k \geq \eta_1 \\ \Delta_k, & \text{if } \rho_k < \eta_1 \text{ and } f^r \text{ is not fully linear} \\ \gamma_{de}\Delta_k, & \text{if } \rho_k < \eta_1 \text{ and } f^r \text{ is fully linear} \end{cases} \tag{8}$$

For constrained black-box problems with black-box constraints, in addition to developing data-driven models for objective function, the models are also constructed for constraints. We make sure that the surrogate models for the constraints also satisfy the fully linear property, i.e.

$$\begin{aligned} &||\nabla g_i(x) - \nabla g_i^r(x)|| \leq \kappa_{dg_i}\Delta \quad \forall x \in B(x_k, \Delta) \\ &|g_i(x) - g_i^r(x)| \leq \kappa_{g_i}\Delta^2 \quad \forall x \in B(x_k, \Delta) \end{aligned} \tag{9}$$

## 3. Two phase algorithm

We will now give an outline of our algorithm for solving Problem (1) and provide detailed description of each of the components.

### 3.1. Outline of the algorithm

The overall scheme is shown in Fig. 1. The algorithm is designed such that the goal of finding a feasible point and the goal of decreasing the objective function are considered independently. The overall algorithm consists of two phases: the first is the feasibility phase that focuses on finding a feasible point and the second phase is termed as optimization phase that focuses on decreasing the objective function while maintaining feasibility at most of the iterations. One may argue that it might be more efficient to consider

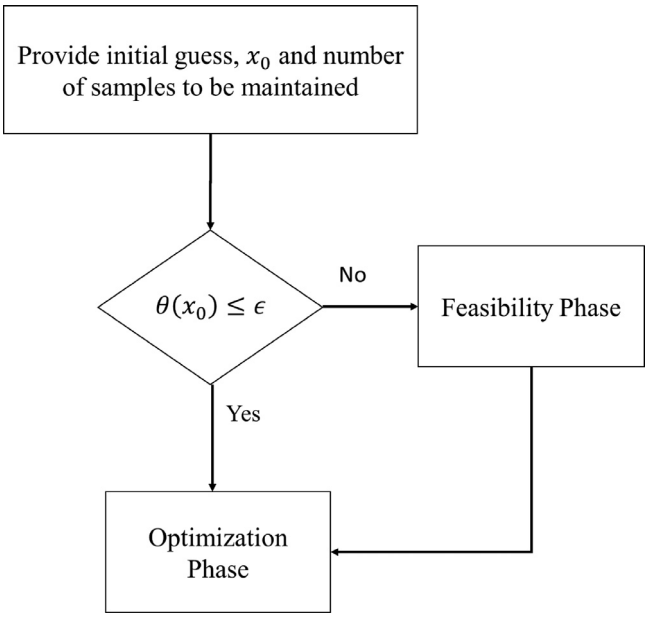


**Fig. 1.** Schematic of the overall algorithm.

decreasing the objective function and infeasibility simultaneously as is done in penalty-based and filter methods. However, from a practitioner's point of view, it is preferred that when the optimization code stops, at least a feasible point is obtained. This is especially important when the problem is black-box and simulations are computationally expensive.

The basic idea of the optimization phase is to construct surrogate models of the objective function and the black-box constraints and solve the optimization subproblems. The updating rule for the objective function and the constraints is based on feasibility of the candidate point as well as the decrease in objective function. We do not require to construct fully linear models at each iteration but only when criticality measure is below a certain threshold so that Eqs. (6), (7) and (9) provide meaningful bounds. Another instance when we require to construct fully linear models is when the algorithm could not decrease the objective function or the constraint violation.

### 3.2. Feasibility phase

The feasibility phase is initiated when the initial point provided by the user is infeasible. We measure the degree of infeasibility at a point by defining a smooth constraint violation function for unknown constraints as follows:

$$\theta(x) = \sum_{i=1}^{p} (max(0, g_i(x)))^2 \tag{10}$$

Note that $\theta = 0$ when a point is feasible and is positive otherwise. We define the optimization problem to be solved during feasibility phase as follows:

$$\begin{aligned} \min_x \quad & \theta(x) \\ s.t. \quad & g_j(x) \leq 0 \quad \forall j \in \{1, \ldots, q\} \\ & x \in [x^L, x^U] \end{aligned} \tag{11}$$

Problem (11) is a black-box problem with known constraints.

The feasibility problem is a box-constrained problem with black-box objective when there are no known constraints. It becomes a black-box problem subject to known constraints when some of the constraints are known. Both of these problems are relatively easy to solve. We adopt a trust-region framework to solve feasibility problem. The decision regarding updating of trust-region center and size are similar to that of unconstrained minimization of black-box problems. For the case when the closed form of all the constraints is known, the sampling technique used in the paper always ensure the feasibility of samples. The overall algorithm for the Feasibility phase (FPA) is given in Algorithm 1. The algorithm initiates by considering the entire space as the initial trust-region and obtaining $m+1$ feasible samples with respect to known constraints. The reason for considering the entire space initially is explore the problem globally and thereby increasing the chances of obtaining global minima. We also allow for exploiting the system knowledge by allowing the user to provide an initial point. The function evaluation is then performed on the samples and the constraint violation values are collected. A surrogate model for the constraint violation, $\theta^r$ is then developed using the samples data and optimized using ANTIGONE (Misener and Floudas, 2014) to obtain a candidate point, $x^{fop}$ by solving:

$$\begin{aligned} \min_x \quad & \theta^r(x) \\ s.t. \quad & g_j(x) \leq 0 \quad \forall j \in \{1, \ldots, q\} \\ & ||x - x_k|| \in \Delta_k \end{aligned} \tag{12}$$

The constraint violation is calculated at $x^{fop}$ and is defined as $\theta(x^{fop})$. If $\theta(x^{fop})$ is less than a pre-specified tolerance, $\theta_{al}$, a feasible point is obtained and the algorithm goes to the optimization phase. If the constraint violation is not below the tolerance, it is then checked if a decrease in the constraint violation is obtained compared to the current iterate. If this is the case, the new trust-region center is updated to $x^{fop}$. The interpolation point which is distant from $x^{fop}$ and damaging the geometry of the sample set is replaced. The first term on the right hand side of Eq. (14) measures the distance and the second measures poisedness. The interpolation set is updated by removing a point that maximizes the combined criteria of distance and poisedness. It is possible that Problem (11) has a local minima at an infeasible point and in order to give an indication to the user to try with a new point, it is important to check if a point $x^{fop}$ is locally optimal. This is done by the comparing criticality measure with the size of the trust-region. The criticality measure for the feasibility problem is given by:

$$\begin{aligned} \psi^{\theta}(x^{fop}) = \min_d \quad & \nabla\theta^r(x^{fop})d \\ s.t. \quad & g_j(x^{fop}) + \nabla g_j(x^{fop})^T d \leq 0 \quad \forall j \in \{1, \ldots, q\} \\ & ||d|| \leq 1 \end{aligned} \tag{13}$$

If both trust-region size and criticality measure is small, $x^{fop}$ is an optimal point for problem (11). If the criticality measure, $\psi^{\theta}$ at $x^{fop}$ is smaller than some tolerance ($\epsilon_{\psi^{\theta}}$), it is possible that $x^{fop}$ is an optimal point of Problem (11). However, bounds given by Eq. (9) is ineffective unless the trust-region is small. Therefore, the size of the trust-region is checked if it is less than some threshold value ($\epsilon_{\Delta}$). In that case, a local minima is achieved but is an infeasible point and the algorithm should be started with a different initial point. If the trust-region size is not below a threshold, the criticality measure gives an indication of proximity to the optima and it is beneficial to decrease the trust-region. Before decreasing the trust-region, it is checked if the surrogate models are fully linear.

If the criticality measure is above $\epsilon_{\psi^{\theta}}$, a parameter defining the ratio of reduction in the constraint violation to the predicted reduction by the surrogate model, $\rho_k^{\theta}$ is calculated. If $\rho_k^{\theta}$ is above a certain threshold value ($\eta_0$), this indicates that the model is sufficiently accurate within the current trust-region size and that the trust-region size can be further increased to allow for larger step length. The trust-region is only decreased if a decrease in the constraint violation is not obtained and the model is fully linear. An-

other possibility of not being able to obtain a decrease in constraint violation could be due to the inaccuracy of the surrogate model. When this happens, a model improvement step is called to construct a fully linear model. The model improvement algorithm (MIA) is discussed in Section 3.4 in detail.

**Algorithm 1** Feasibility Phase Algorithm (FPA).

**STEP 0**: Choose initial $m$ interpolation points, initial guess ($x_0$), initial trust-region radius $\Delta_0 = \Delta_{\max} = ||x^U - x^L||$, $\eta_0$, $\gamma_{de}$, $\gamma_{in}$, $k = 0$, $\epsilon_\Delta$, $\epsilon_{\psi^\theta}$ and $\theta_{al}$.

**STEP 1**: Solve Eq. (28) to construct the surrogate model, $\theta^r$ (Eq. 26) for constraint violation, $\theta$ and solve (12) to obtain $x^{fop}$.

**STEP 2**: Update the next iterate, trust-region and interpolation set. Define

$$y_k^o = \underset{y^j \in Y_k}{argmax} ||y^j - x^{fop}||^2 |l_j(x^{fop})| \tag{14}$$

**if** $(\theta(x^{fop}) \leq \theta_{al})$ **then**, STOP and return $x^{fop}$
**end if**
**if** $(\theta(x^{fop}) < \theta(x_k))$ **then**, $x_{k+1} = x^{fop}$ and $Y_{k+1} = Y_k \backslash y_k^o \cup x^{fop}$
**end if**
**if** $(\psi^\theta(x^{fop}) \leq \epsilon_{\psi^\theta})$ **then**,
  **if** $(\Delta_k \leq \epsilon_\Delta)$ **then**, STOP, no feasible point found. Restart from a different initial point.
  **else**
    **if** (model is fully linear) **then**, $\Delta_{k+1} = \gamma_{de}\Delta_k$
    **else** $\Delta_{k+1} = \Delta_k$ and initiate MIA
    **end if**
  **end if**
**else** Calculate $\rho_k^\theta = \frac{\theta(x^{fop}) - \theta(x_k)}{\theta^r(x^{fop}) - \theta^r(x_k)}$
  **if** $(\rho_k^\theta \geq \eta_0)$ **then**, $\Delta_{k+1} = \gamma_{in}\Delta_k$
  **else if** $(\rho_k^\theta < \eta_0$ and $\theta(x^{fop}) < \theta(x_k))$ **then**, $\Delta_{k+1} = \Delta_k$
  **else if** $(\theta(x^{fop}) \geq \theta(x_k)$ and the model is not fully linear) **then**, $\Delta_{k+1} = \Delta_k$ and initiate MIA and replace at most 2 sampling points
  **else if** $(\theta(x^{fop}) \geq \theta(x_k)$ and the model is fully linear) **then**, $\Delta_{k+1} = \gamma_{de}\Delta_k$
  **end if**
**end if**

**Step 3**: Increment $k$ by one and go to Step 1.

### 3.3. Optimization phase

Once a feasible point is obtained, optimization phase is triggered. The feasible point obtained in the feasibility phase is included in the initial set of samples for the optimization phase. This allows the surrogate models in the optimization phase to have the information about at least one point in the feasible region. With this information, the optimization phase then focuses on finding points with better objective function values in the feasible region. This phase attempts to find a non-increasing sequence of feasible iterates which converges to at least a local minima. Similar to the feasibility phase, the optimization phase considers $m$ interpolation points in the entire space to explore the global features of the problem. A feasible point $x_0^f$ is also provided as one of the initial samples to give an indication to the model about the feasible region. The pseudocode of the optimization phase is given in Algorithm 2. It is similar in spirit to the feasibility phase except for the rule to update the trust-region center. After performing the function evaluation at the sample set, $Y_k$, surrogate models are constructed for the objective function ($f^r(x)$) and the constraints ($g_i^r(x)$) with unknown closed forms. The following problem is then solved to obtain a candidate point, $x^{sop}$:

$$\begin{aligned} \min\ & f^r(x) \\ s.t.\ & g_i^r(x) \leq 0 \qquad \forall i \in \{1, \cdots, p\} \\ & g_j(x) \leq 0 \qquad \forall j \in \{1, \cdots, q\} \\ & \| x - x_k \| \in \Delta_k \end{aligned} \tag{15}$$

where $g_i^r$ is the surrogate model for the $i$th constraint whose form is unknown. The constraint violation at $x^{sop}$ is calculated and checked if it is less than a pre-specified tolerance ($\theta_{tol}$). If it is less, it is examined if a decrease in the objective function is obtained. If that is the case, then the trust-region center is updated to $x^{sop}$. The interpolation set is also updated by including $x^{sop}$ and discarding the worst point. To determine if $x^{sop}$ is optimal, the criticality measure ($\psi^f(x^{sop})$) is calculated and examined if it is less than some tolerance ($\epsilon_{\psi^f}$). The criticality measure corresponding to the optimization phase is calculated by:

$$\begin{aligned} \psi^f(x^{sop}) = \min_d\ & \nabla f^r(x^{sop})d \\ s.t.\ & g_i^r(x^{sop}) + \nabla g_i^r(x^{sop})^T d \leq 0 \quad \forall i \in \{1, \ldots, p\} \\ & g_j(x^{sop}) + \nabla g_j(x^{sop})^T d \leq 0 \quad \forall i \in \{1, \ldots, q\} \\ & ||d|| \leq 1 \end{aligned} \tag{16}$$

The criteria about the trust-region size is also verified before giving the certificate of optimality. The criticality measure can be small due to inaccuracy of the surrogate models. Therefore, before the trust-region is decreased, the fully linear property of the model is ensured to hold true. If the criticality measure is above the threshold ($\epsilon_{\psi^f}$), the ratio of actual reduction in the objective function to the predicted reduction given by the surrogate model is calculated. Note that it is not reasonable to assume the predicted reduction to be always positive as it happens in classic derivative-basedproblems. The reason for predicted reduction being negative can be due to inaccuracy of the surrogate models. Therefore, it is ensured that there is an actual reduction in the objective function before the trust-region size is increased. And it is kept constant if the ratio, $\rho_k^f$ is less. The trust-region size is decreased when the model is fully linear and there is no improvement in the objective function.

In many problem instances, it may happen that the objective function can be decreased while maintaining feasibility only along a thin domain. In extreme cases, such a thin domain can be an equality constraint. It is not trivial to obtain a search direction when the derivatives are not available. NOWPAC and CONORBIT addressed this issue by adding a small offset in problem (15). Once it is ensured that the model is fully linear, the trust-region size is decreased. The filter approach e.g., Eason and Biegler (2016) accepts infeasible point if the new point reduces the objective function sufficiently compared to the constraint violation. We compare the criticality measure to the trust-region size to verify if it is indeed the case of thin domain. If the criticality measure is more than a fraction of the trust-region size, the tolerance on the allowable constraint violation is increased so that slightly infeasible solution is admissible. Once a better objective function is obtained, the tolerance on the constraint violation is reset to the original value. It is possible that problem (15) becomes infeasible at certain iterations. This is due to finite tolerance on constraint violation, all the samples may be slightly infeasible. When this happens, the following subproblem is solved:

$$\begin{aligned} \min_x\ & f^r(x) \\ s.t.\ & g_i^r(x) \leq g_i(x_k) \qquad \forall i \in \{1, \cdots, p\} \\ & g_j(x) \leq 0 \qquad \forall j \in \{1, \cdots, q\} \\ & \| x - x_k \| \in \Delta_k \end{aligned} \tag{17}$$

At this point, we should point out that the tolerance on constraint violation can be selected as 0 but that may slow the progression of the algorithm.

### 3.4. Model improvement

It is known that to ensure the surrogate models to satisfy fully linear property, the set of interpolating points need to be well-poised and is very critical to ensure convergence of DFO algorithms (Conn et al., 2009a). In broad terms, the meaning of well-poised samples is that their geometry is adequate in the sense that it should span all the directions of the space. It is possible to choose an initial set of samples which are well-poised. The models need to be ensured to be fully linear in the course of iterations of the algorithm. A very simple but crude way for doing so is to replace all the samples at each iteration such that all of them lie in the trust-region, $B(x_k, \Delta_k)$. However, this approach is not efficient and most likely will result in large number of evaluations. Therefore, assuming that the initial set of interpolating samples are well-poised, a mechanism is needed that replaces one sample at each iteration while improving the geometry of the samples. Also, note that we do not require the models to be fully-linear at all iterations but only when the objective function is not reduced or the criticality measure is below a certain threshold.

To this end, we employ the self-correcting geometry approach proposed by Scheinberg and Toint (2010). The idea of the method is to replace an existing point in the interpolation set, $Y_k$ by either $x^{fop}$ or $x^{sop}$ if the algorithm is in the feasibility phase or the optimization phase, respectively.

In the literature, one of the metrics of well-poisedness is given by Lagrange polynomials (Powell, 1994b). If a set of interpolation points $Y_k = \{y_0, y_1, \ldots, y_m\}$ is defined, correspondingly a basis of polynomials $l_i(x), i = 0, 1, \ldots, m$ is basis for Lagrange polynomials such that:

$$l_i(y_j) = \delta_{ij} \tag{18}$$

where $\delta_{ij}$ is the Kronecker delta. Clearly, if the degree of the Lagrange polynomials is 2, the number of interpolation points required to uniquely determine its coefficients is $m^Q = \frac{(n+1)(n+2)}{2}$. But, this is not favorable when the computational budget is limited. In this paper, we maintain the number of interpolation points that is less than $m^Q$. To compute the coefficients of the Lagrange polynomials, a convex optimization problem that minimizes the Hessian of the Lagrange polynomials is solved:

$$\begin{aligned} \min \quad & \| \nabla^2 l_i \| \\ s.t. \quad & l_i\left(y^j\right) = \delta_{ij} \end{aligned} \tag{19}$$

A set of interpolation points is said to be $\Lambda$-poised if the corresponding Lagrange polynomials are bounded, i.e.

$$\max_{i=0,1,\ldots,m} \max_{x \in B} |l_i(x)| \leq \Lambda \tag{20}$$

The criteria used to replace a point is based on combined poisedness and distance measure such that on replacement, the geometry of the samples are improved. The model improvement algorithm is called by either the optimization phase or the feasibility phase whenever the geometry of the samples need to be improved. A pseudocode for MIA is given in Algorithm 3.

Firstly, the set of points lying outside the current trust-region are identified denoted by $\mathbb{F}_k$. The point which maximizes the distance and poisedness measure is replaced. If $\mathbb{F}_k$ is empty, a set of non-poised close points, $\mathbb{C}_k$ is identified. Again, a point that maximizes the combined distance and poisedness criteria is identified. If both the sets $\mathbb{F}_k$ and $\mathbb{C}_k$ are empty, then the current interpolation set is well-poised and the surrogate model is fully linear.

**Algorithm 2** Optimization Phase Algorithm (OPA).

**STEP 0**: Choose initial $m$ interpolation points, feasible point obtained from feasibility phase (${x_0}^f$), initial trust-region radius $\Delta_0 = \Delta_{\max} = ||x^U - x^L||$, $\eta_0$, $\gamma_{de}$, $\gamma_{in}$, $k = 0$, $\epsilon_\Delta$, $\epsilon_{\psi^f}$ and $\theta_{al}$, $\theta_{tol} = \theta_{al}$, $\nu < 1$.
**STEP 1**: Solve Eq. (28) to construct surrogate model, $f^r$ (Eq. 26) for the objective function and set of surrogate models, $g_i^r$ $\forall i \in \{1, \cdots, p\}$ (Eq. 26) to approximate unknown constraints. Solve Problem (15) to obtain $x^{sop}$.
**if** (Problem (15) is infeasible) **then**, solve Problem (17) to obtain $x^{sop}$
  **if** $(\psi^f(x^{sop}) > \nu\Delta_k)$ **then**, Replace all the interpolation point by Eq. (29)
  **end if**
**end if**
**STEP 2**: Update the next iterate, trust-region and interpolation set.
$y_k^o = \underset{y^j \in Y_k}{\operatorname{argmax}} ||y^j - x^{sop}||^2 |l_j(x^{sop})|$
**if** $(\theta(x^{sop}) > \theta_{tol})$ **then**,
  **if** (Model is fully linear) **then**, $\Delta_{k+1} = \gamma_{de}\Delta_k$
    **if** $(\psi^f(x^{sop}) > \nu\Delta_k)$ **then**, $\theta_{tol} = 10\theta_{al}$
    **end if**
  **else** Initiate MIA
  **end if**
**else**
  **if** $(f(x^{sop}) < f(x_k))$ **then**, set $\theta_{tol} = \theta_{al}$, $x_{k+1} = x^{sop}$, $Y_{k+1} = Y_k \backslash y_k^o \cup x^{sop}$
  **end if**
  **if** $(\psi^f(x^{sop}) \leq \epsilon_{\psi^f})$ **then**,
    **if** $(\Delta_k \leq \epsilon_\Delta)$ **then**, STOP!
    **else**
      **if** (Model is fully linear) **then**, $\Delta_{k+1} = \gamma_{de}\Delta_k$
      **else** $\Delta_{k+1} = \Delta_k$ and initiate MIA
      **end if**
    **end if**
  **else** Calculate $\rho_k^f = \frac{f(x^{sop}) - f(x_k)}{f^r(x^{fop}) - f^r(x_k)}$
    **if** $(\rho_k^f \geq \eta_0$ and $f(x^{sop}) < f(x_k))$ **then**, $\Delta_{k+1} = \gamma_{in}\Delta_k$
    **else if** $(\rho_k^f < \eta_0$ and $f(x^{sop}) < f(x_k))$ **then**, $\Delta_{k+1} = \Delta_k$
    **else if** $(f(x^{sop}) \geq f(x_k)$ and the model is not fully linear) **then**, $\Delta_{k+1} = \Delta_k$ and initiate MIA
    **else if** $(f(x^{sop}) \geq f(x_k)$ and the model is fully linear) **then**, $\Delta_{k+1} = \gamma_{de}\Delta_k$
    **end if**
  **end if**
**end if**
**Step 3**: Increment $k$ by one and go to Step 1.

### 3.5. Surrogate modeling

As we have discussed earlier, we develop surrogate models $\theta^r$, $f^r$ and $g_i^r$. Surrogate model is an inexpensive approximation of the unknown equations that enables one to utilize derivative-based algorithms to guide the search towards the true optimum of the problem. The surrogate model serves the purpose of implicitly approximating derivative of the underlying function. The surrogate models considered are interpolation or regression based. Certain surrogate model might be well suited for some problems while another surrogate model might be more appropriate for other problems. We are interested in the surrogate models which satisfy the fully linear property. Quadratic (Powell (2006), Conn et al. (2009a), Sampaio and Toint (2015), Conejo et al. (2015)) and radial basis function (Wild et al. (2008), Regis (2014), Regis and Wild (2017))

**Algorithm 3** Model Improvement Algorithm (MIA).

**STEP 0**: Choose parameters $\beta \geq 1$, $\Lambda > 1$ and given a candidate point $x^{op}$.

**STEP 1**: Define a set consisting of distant interpolation points such that:

$$\mathbb{F}_k = \{y^j \in Y_k | \ ||y^j - x^{op}|| > \beta\Delta_k \text{ and } |l_{k,j}| \neq 0\} \tag{21}$$

**if** ($\mathbb{F}_k \neq \emptyset$) **then**, amongst the farthest points, replace a point $y^r \in \mathbb{F}_k$ such that

$$y^r \in \underset{y^j \in \mathbb{F}_k}{argmax} \ ||y^j - x^{op}||^2 |l_j(x^{op})| \tag{22}$$

Define the new interpolation set, $Y_{k+1} = Y_k \setminus \{y^r\} \cup \{x^{op}\}$

**else if** ($\mathbb{F}_k = \emptyset$) **then** Define a set consisting of close interpolation points such that:

$$\mathbb{C}_k = \{y^j \in Y_k \setminus \{x_k\} | \ ||y^j - x^{op}|| \leq \beta\Delta_k \text{ and } |l_{k,j}(x^{op})| > \Lambda\} \tag{23}$$

**if** ($\mathbb{C}_k \neq \emptyset$) **then**, Amongst the closest point, replace a point $y^r \in \mathbb{C}_k$

$$y^r \in \underset{y^j \in \mathbb{C}_k}{arg\,max} \ ||y^j - x^{op}||^2 |l_j(x^{op})| \tag{24}$$

Define the new interpolation set, $Y_{k+1} = Y_k \setminus \{y^r\} \cup \{x^{op}\}$

**else** The model is fully linear

**end if**

**end if**

are some of the commonly used surrogate models to approximate the original function. We build surrogate models for the constraint violation function during feasibility phase and for the objective function and unknown constraints during optimization phase.

Due to the ability of radial basis function (RBF) to be more effective in modeling multimodal behavior (Gutmann, 2001), we use RBFs to approximate the unknown functions. Although kriging model has also been used to model multimodal behavior, parameter estimation problem used to determine the parameters of the model is nonlinear and can be computationally expensive (Hasan et al., 2013). The parameter estimation problem for RBF is linear and therefore, it is computationally efficient to use RBF.

RBF is a broad class of functions formed by linear combination of nonlinear basis function:

$$s^r(x) = \sum_{j=0}^{m} \omega_j \phi(||x - y^j||_2) + p(x) \tag{25}$$

The basis function, ($\phi$) can be linear ($\phi = r$), cubic ($\phi = r^3$), multiquadratic ($\phi = \sqrt{\gamma^2 + r^2}$), thin plate spline ($\phi = r^2 log\ r$) and Gaussian ($\phi = e^{-\gamma r^2}$). In general, $p(x)$ is a polynomial defined as linear combination of $x_1^{\xi_1}, \cdots, x_n^{\xi_n}$. Essentially, RBFs encompasses a range of functions that allows to approximate the original function effectively. In this work, we use cubic radial basis function (CRBF) with linear polynomial tail to approximate a function ($f(x)$, $\theta(x)$, $g_i(x)$) and has the following functional form:

$$s_c^r(x) = \sum_{i=1}^{n} b_i x_i + \sum_{j=0}^{m} \omega_j \left( \sqrt{\sum_{i=1}^{n} (x_i - y_i^j)^2} \right)^3 \tag{26}$$

In the surrogate model ($s_c^r(x)$), there are a total of $n + m + 1$ parameters that need to be estimated ($b_i$ and $\omega_j$). The approach suggested by Gutmann (2001) is to use combination of interpolation and positive-definiteness condition, i.e.

$$\begin{aligned} & s_c^r(y^j) = s(y^j) \quad \forall j \in \{0, \ldots, m\} \\ & \sum_{j=0}^{m} \omega_j y_i^j = 0 \quad \forall i \in \{1, \ldots, n\} \end{aligned} \tag{27}$$

The above set of $(n + m + 1)$ linear equations can be solved to uniquely determine the parameters of the model. However, it is required that the geometry of the samples is good so that the linear set of equations are linearly independent. In our proposed method, we do not maintain the geometry of the samples at all iterations. Therefore, we solve a linear optimization problem (LP) by introducing slack variables and thereafter minimizing the slack variables so that the Eqs. (27) are satisfied:

$$\begin{aligned} & \min_{\substack{b_i, \omega_j \\ SP_i^1, SN_i^1 \\ SP_j^2, SN_j^2}} \ \sum_{i=1}^{n} (SP_i^1 + SN_i^1) + \sum_{j=0}^{m} (SP_j^2 + SN_j^2) \\ & s.t. \quad \sum_{i=1}^{n} b_i y_i^j + \sum_{j'=0}^{m} \omega_{j'} \left( \sqrt{\sum_{i=1}^{n} (y_i^{j'} - y_i^j)^2} \right)^3 = s_j + SP_j^2 - SN_j^2 \\ & \forall j = \{0, \cdots, m\} \\ & \sum_{j=0}^{m} \omega_j y_i^j = SP_i^1 - SN_i^1 \quad \forall i = \{1, \cdots, n\} \\ & SP_i^1, SN_i^1, SP_j^2, SN_j^2 \geq 0 \end{aligned} \tag{28}$$

The above LP model is solved to derive the surrogate model parameters for $f^r, \theta^r$ *and* $g_i^r$.

### 3.6. Initial sampling

In many of black-box optimization problems, the simulation is computationally expensive. In this scenario, it is critical that initial set of samples to be selected spans the entire space efficiently. Several sampling schemes are available in the literature and interested reader may refer a recent review paper by Garud et al. (2017) to learn about various static and adaptive sampling schemes. We use two alternate approaches for sampling depending on the constraints of the problem. If all the constraints of the problem are black-box, then we use the approach of Powell (2006) as our first sampling strategy. In this paper, we maintain a set of $2n + 1$ interpolation points i.e. $m = 2n$. The samples are given as follows:

$$\begin{aligned} & y^i = x_{0i} + \Delta_0 e_i, \quad y^{n+i+1} = x_{0i} - \Delta_0 e_i, \quad if \quad x_i^L < x_{0i} < x_i^U \\ & y^i = x_{0i} + \Delta_0 e_i, \quad y^{n+i+1} = x_{0i} + 2\Delta_0 e_i, \quad if \quad x_{0i} = x_i^L \\ & y^i = x_{0i} - \Delta_0 e_i, \quad y^{n+i+1} = x_{0i} - 2\Delta_0 e_i, \quad if \quad x_{0i} = x_i^U \end{aligned} \tag{29}$$

where $x_{0i}$ is the *ith* component of the initial guess, $x_0$ and $\Delta_0$ is the initial trust-region size. In many engineering applications, we do not have completely black-box constraints and may have a subset of known constraints. It is beneficial to take the advantage of known constraints by sampling only in the feasible region (Bajaj and Hasan, 2016). The known constraints may be unrelaxable which means that the simulation fails at certain samples that do not satisfy the unrelaxable constraints. Boukouvala and Floudas (2017) suggested a two step strategy to obtain feasible samples. The first step involved obtaining a large number of samples using Latin Hypercube design (LHD) and filtering out the ones which are infeasible with respect to known constraints. In the next step, distance criteria is used to select the final set of samples. A key issue with this approach is that it is difficult to determine appropriate number of samples to be given by LHD *a priori*.

In our second sampling strategy, we therefore propose an optimization based sampling strategy such that a desired number of samples, which are space filling and feasible with respect to known constraints, are obtained. Optimization formulations based on entropy, integrated mean squared error, minimax and maximin distances and discrepancies can be used as metric to measure space filling characteristics. We minimize the wrap-around $L^2$ discrepancy (Hickernell, 1998) and add known constraints resulting in the

following nonlinear problem:

$$\begin{aligned} \min_{u} \quad & \left(\frac{4}{3}\right)^n + \frac{1}{m^2}\sum_{j=1}^{m}\sum_{j'=1}^{m}\prod_{i=1}^{n}\left[\frac{3}{2} - |u_i^j - u_i^{j'}|(1 - |u_i^j - u_i^{j'}|)\right] \\ s.t. \quad & g_j(u) \le 0 \qquad \forall j \in \{1,\ldots,q\} \\ & ||u - u_k|| \in \hat{\Delta}_k \end{aligned} \tag{30}$$

Here, $u_i^j$ is the scaled $i$th component of $j$th sample. Both of our sampling strategies are deterministic in nature and the only parameter that will change the output of the algorithm is the initial guess. Therefore, the strategy we propose is deterministic in nature as long as the initial guess remains the same.

While the filter approach of Eason and Biegler (2016) considers decreasing objective function and constraint violation simultaneously by building a pareto front, we consider the two aspects independently. It is sometimes not possible to get a point which is feasible and improves the objective function at the same time especially when the feasible domain is thin. The filter approach of Eason and Biegler (2016) deals with this problem by accepting an infeasible point if it decreases the objective function sufficiently. However, in our approach, we increase the tolerance on constraint violation by checking if criticality measure is above a threshold, size of trust-region is below a threshold and the model is fully linear. The algorithm proposed in this paper is similar in spirit to NOWPAC and CONORBIT in the sense that it is neither based on penalty or filter approach. A key difference, however is that we do not assume availability of an initial feasible point which is unreasonable for many practical applications. Secondly, we do not use a margin or offset in the constraints. NOWPAC and CONORBIT also assume that all the constraints are black-box and relaxable. It is possible in many cases that the simulation fails if the samples do not satisfy certain constraints. Our derivative-free optimization framework can handle these constraints efficiently.

## 4. Computational studies

In this section, we provide computational results for our two phase algorithm. After demonstrating the algorithm on a nonlinear 2 dimensional example, we provide comparison of the performance of our method and compare it to COBYLA and NOMAD. The details of the algorithm parameters with their values used is provided in Table 3. We also demonstrate the application of the algorithm on a chemical engineering case study.

### 4.1. An illustrative example

We start the two phase algorithm using a two-dimensional nonlinear smooth problem, `st_e18`. The problem can be stated as follows:

$$\begin{aligned} \min_{x} \quad & x_1 + x_2 \\ s.t. \; & -x_1^2 - x_2^2 \le -1 \\ & x_1^2 + x_2^2 \le 4 \\ & -x_1 + x_2 \le 1 \\ & x_1 - x_2 \le 1 \\ & x_1 \in [-2, 2] \\ & x_2 \in [-2, 2] \end{aligned} \tag{31}$$

In this numerical problem, although the analytical functional form is known, we do not utilize the derivative information and only simulations can be performed. The variable bounds are assumed to be known and only the nonlinear constraints are considered black-box.

**Table 1**
Samples and corresponding constraint violation at iteration 0 of feasibility phase.

| Sample no. | Sample | $\theta$ |
|---|---|---|
| 1 | (−2, −2) | 16 |
| 2 | (2,−2) | 25 |
| 3 | (−2,2) | 25 |
| 4 | (0,−2) | 1 |
| 5 | (−2,0) | 1 |

**Table 2**
Samples and corresponding Lagrange polynomials at iteration 0.

| Sample | Lagrange Polynomials ($l$) |
|---|---|
| (−2, −2) | $-1 - 0.25x_1 - 0.25x_2 + 0.5(0.25x_1^2 + 0.25x_2^2)$ |
| (2,−2) | $0.25x_1 + 0.5(0.25x_1^2)$ |
| (−2,2) | $0.25x_2 + 0.5(0.25x_2^2)$ |
| (0,−2) | $1 + 0.5(-0.5x_1^2)$ |
| (−2,0) | $1 + 0.5(-0.5x_2^2)$ |

**Table 3**
Algorithm parameters.

| Parameters | Values |
|---|---|
| Threshold for changing the trust-region size ($\eta_0$) | 0.1 |
| Factor of decreasing the trust-region size ($\gamma_{de}$) | 0.5 |
| Factor of increasing the trust-region size ($\gamma_{in}$) | 3 |
| Minimum trust-region size ($\epsilon_\Delta$) | $10^{-6}$ |
| Tolerance on criticality measure for feasibility problem ($\epsilon_{\psi^\theta}$) | $10^{-3}$ |
| Allowable constraint violation for the final solution ($\theta_{al}$) | $10^{-8}$ |
| Comparison factor of trust-region size and criticality measure ($\nu$) | 0.1 |
| Factor indicating a particular point to be farther compared to the trust-region size ($\beta$) | 1.2 |
| Threshold on poisedness measure ($\lambda$) | 2 |

The algorithm starts by providing the lower bound, i.e., (−2, −2) as the initial guess. Clearly, the initial guess is infeasible and feasibility phase is therefore initiated. Initial samples generated at iteration 0 using Eq. (29) and the corresponding constraint violation values are listed in Table 1. None of the initial samples are feasible. Fig. 4 shows the behavior of the constraint violation function. As we notice, the feasible region is small and a systematic method is needed to obtain a feasible point. In the next step, surrogate model for the scaled constraint violation is developed and the model parameters are estimated by solving (25). The functional form of the obtained surrogate model is:

$$\begin{aligned} f^r(x) = & -1.185x_1 - 1.185x_2 + 2.278\left(\sqrt{(x_1)^2 + (x_2)^2}\right)^3 \\ & + 0.427\left(\sqrt{(x_1 - 1)^2 + (x_2)^2}\right)^3 \\ & + 0.427\left(\sqrt{(x_1)^2 + (x_2 - 1)^2}\right)^3 \\ & - 0.853\left(\sqrt{(x_1 - 0.5)^2 + (x_2)^2}\right)^3 \\ & - 0.853\left(\sqrt{(x_1)^2 + (x_2 - 0.5)^2}\right)^3 \end{aligned} \tag{32}$$

The above surrogate model is then optimized and the sample (0.4199,0.4199) is obtained. When the function is evaluated at this point, there is improvement in the constraint violation. The interpolation point which maximizes the combined distance and poisedness measure is replaced by the new point. The Lagrange polynomials used to measure the poisedness is listed in Table 2. Note that the Lagrange polynomials do not depend on the objective function values but rather only on set of interpolation points. The criticality measure at this point is $1.426 \times 10^{-9}$ which is less than the pre-specified tolerance, $\epsilon_{\psi^\theta}$ but the size of the trust-

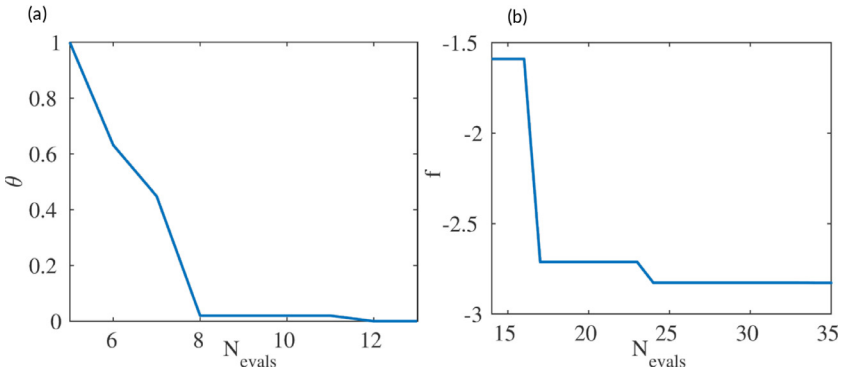


**Fig. 2.** Decreasing (a) constraint violation, and (b) objective function with number of evaluations.

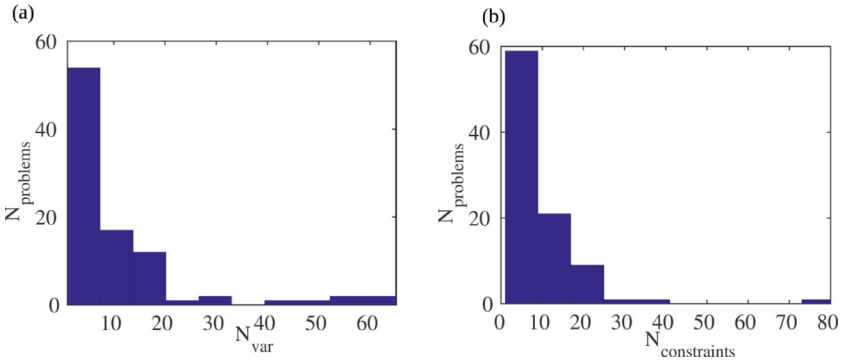


**Fig. 3.** Distribution of problems with (a) number of variables, and (b) number of constraints.

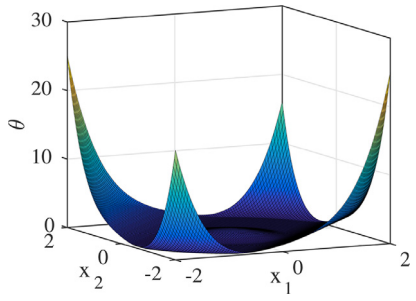


**Fig. 4.** Constraint violation function for `st_e18`.

region is more than the tolerance, $\epsilon_\Delta$. Repeating the steps as listed in Algorithm 1, a feasible point is obtained in 8 iterations and requires a total of 13 function evaluations. Fig. 5(a) shows the movement of iterates during the feasibility phase. The feasibility phase starts with an infeasible point and ultimately gives a feasible solution Fig. 2(a) gives the progress of the iterates with cumulative number of evaluations.

Fig. 5 shows the movement of the iterates during the optimization phase. It starts with the objective value of −1.5897 and finally obtains an objective function value of −2.823. The global minima of the problem is −2.8284. The progress of minimum objective function value with number of evaluations during the optimization phase is given in Fig. 2(b). The algorithm is able to obtain a solution which is within 1% of the global minima in 24 function evaluations.

### 4.2. Globallib test problems

We applied our two-phase approach to a comprehensive list of 92 test problems from GlobalLib (2015) for which the global minima are known. The test problems are the same as those used by Boukouvala and Floudas (2017). All the problems have only inequality constraints. We solve the most difficult problem by as-

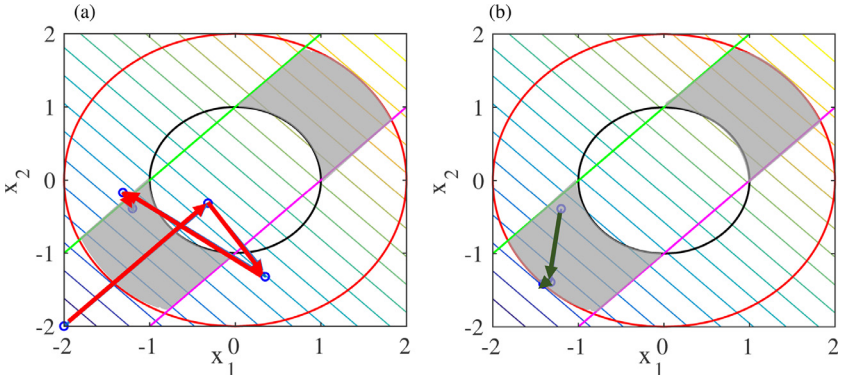


**Fig. 5.** The figure shows the contour plot of the objective function and the feasible region is shown by the grey region. It also illustrates the progression of the iterates in (a) feasibility phase and (b) optimization phase.

suming that all the constraints have unknown analytical functional form. The dimensions of these 92 problems vary from 1 to 65 while the number of constraints range from 1 to 81. The distribution of the problems with dimension and constraints is given in Fig. 3. For the problems which did not have lower and upper bounds for the variables given in the problem formulation, we assigned the lower and upper bounds as follows:

$$\begin{aligned} x_i^L &= x_i^{glob} - 10 \quad \forall i \in \{1, \ldots, n\} \\ x_i^U &= x_i^{glob} + 10 \quad \forall i \in \{1, \ldots, n\} \end{aligned} \tag{33}$$

It is assumed that a single function evaluation gives the values of the objective function and all the constraints. For all the problems, lower bound on the variables is provided as the initial guess. Table 1 in the supplementary material lists each of the problem's dimension, number of constraints and whether the initial point is feasible or infeasible. The algorithm is implemented in MATLAB while GAMS/ANTIGONE is used to solve the surrogate optimization problems. At each iteration, $2n+1$ samples are maintained for developing surrogate models.

In derivative-based solvers, the performance measure is generally the time taken by a solver $s$ to solve a problem $p$. The computational expense required by a derivative-free solver includes the simulation time and the time required to solve all the subproblems. The numerical examples used in this paper require negligible computation cost for simulation and therefore, computation time is not an appropriate metric. The simulation cost can be much higher in other applications and therefore an appropriate metric to measure the performance is the number of samples required to satisfy a convergence test. Two major approaches to compare the performance of derivative-free optimizations have been proposed in the literature (Moré and Wild, 2009). One of the approaches is to use a performance profile. It compares solvers by defining the performance ratio for a solver. Let the set of test problems be $\mathbb{T}$, set of solvers/algorithms be $\mathbb{S}$ and the performance measure be $N_{t,s}$. To get a performance profile, the performance ratio is defined as follows:

$$r_{t,s} = \frac{N_{t,s}}{\min\limits_{s \in \mathbb{S}} N_{t,s}} \tag{34}$$

The ratio, $r_{t,s}$ is 1 for the best solver and greater than 1 for worse solvers. The performance profile for a solver is defined as the fraction of problems for which the performance ratio is at most $\alpha$:

$$\varrho_s(\alpha) = \frac{1}{|\mathbb{T}|} \text{size}\{t \in \mathbb{T} : r_{t,s} \leq \alpha\} \tag{35}$$

where $|\mathbb{T}|$ denotes the cardinality of $\mathbb{T}$. $\varrho_s(\alpha)$ represents the fraction of problems that has performance ratio less than $\alpha$. A solver with higher value of $\varrho_s(\alpha)$ is favorable compared to the lower values.

While the performance profile provides a ranking of the solvers relative to each other, it does not give the complete information about the fraction of problems solved with the number of function evaluations by an individual solver. A data profile considers the absolute performance of a solver and determines its ability to solve an optimization problem. Data profile represents the fraction of problems solved within $\alpha$ function evaluation and defined as follows:

$$d_s(\alpha) = \frac{1}{|\mathbb{T}|} \text{size}\{t \in \mathbb{T} : N_{t,s} \leq \alpha\} \tag{36}$$

We present the comparison using both performance and data profiles. The convergence test used to certify a problem to be solved is also critical for the analysis of different algorithms. We examine two convergence criteria in our computational comparison. The first criterion is to compare the absolute merit of the solution.

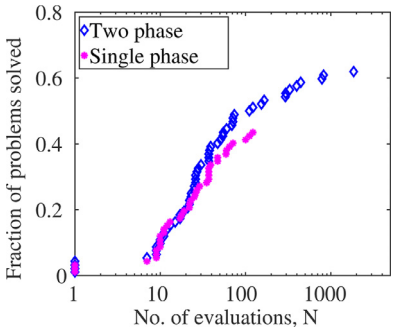


**Fig. 6.** Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that the solution is within 1% of the global minima (Eq. (37)). The figure compares the performance of the two phase algorithm with the algorithm when no feasibility phase is involved.

One of the approaches that can be used is to compare the accuracy with the global solution as done by Rios and Sahinidis (2013). Therefore, one of the convergence test used in this study is that a solver is said to have solved a problem if the constraint violation is less than $10^{-8}$ and the solution is within 1% of the global solution. We say that a solver is able to solve a problem if a point $\hat{x}$ is obtained by the solver that satisfies:

$$\theta(\hat{x}) <= 10^{-8} \quad \& \quad f(\hat{x}) \leq \max(1.01 f(x^*), f(x^*) + 0.01) \tag{37}$$

where $x^*$ is the global solution. The above criterion is often used when a user is interested in the accuracy of the solution. However, when the computational budget of the user is limited, the ability of a solver to improve an initial point can also be of interest. For an unconstrained problem, the criteria is defined as follows:

$$f(\hat{x}) \leq f(x^*) + \tau(f(x^0) - f(x^*)) \tag{38}$$

This criterion can be interpreted as follows. If a solver is able to reduce the objective function by at least $1-\tau$ times the maximum reduction possible, it is considered successful in solving a problem. The smaller value of $\tau$ implies that a solution closer to global minima is desired.For constrained problems, the ability to improve the constraint violation corresponding to the initial point is critical. Therefore, a merit function is defined that incorporates the constraint violation information:

$$\varphi(x) = f(x) + \upsilon\theta(x) \tag{39}$$

In this paper, a fixed penalty parameter value of $\upsilon = 1000$ is chosen. The second criterion is based on the merit function and is defined similar to (38) by replacing $f$ with $\varphi$.

$$\varphi(\hat{x}) \leq \varphi(x^*) + \tau(\varphi(x^0) - f(x^*)) \tag{40}$$

#### 4.2.1. Importance of feasibility phase

CONORBIT, COBYLA, NOWPAC solves Eq. (15) iteratively and do not have any feasibility phase in their algorithm. Eq. (15) embeds the information about the constraints as well as the objective function. Therefore, it is important to elucidate the advantage of using the feasibility phase. To understand the contribution of the feasibility phase in convergence of our algorithm, we tested our algorithm with and without feasibility phase on the entire set of numerical test problems. Fig. 6 illustrates the performance of the algorithm with and without feasibility phase. If the feasibility phase is not used, a total of 40 problems are solved while 56 problems are solved when feasibility phase is used. Therefore, the feasibility phase plays a critical role to provide superior starting point for the optimization phase and is important for convergence.

#### 4.2.2. Computational comparison with other solvers

We compare our algorithm with NOMAD and COBYLA. NOMAD (Le Digabel, 2011) is an implementation of Mesh Adaptive Direct

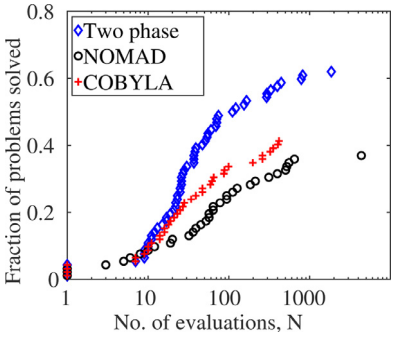


**Fig. 7.** Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that the solution is within 1% of the global minima (Eq. (37)).

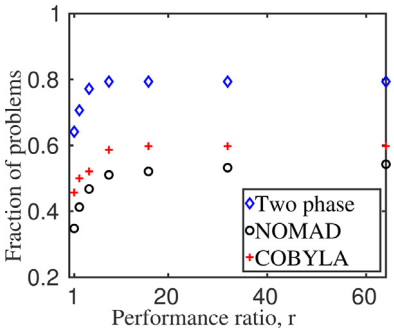


**Fig. 8.** Performance profile indicating the relative performance of the solvers based on the convergence test that the solution is within 1% of the global minima (Eq. (37)).

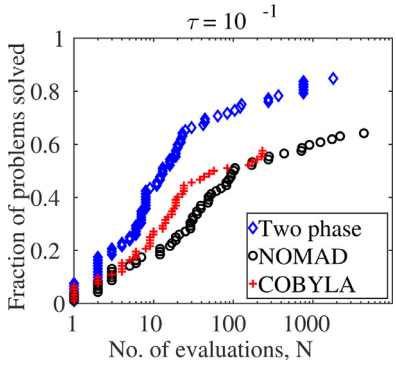


**Fig. 9.** Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that a significant reduction in the final merit function is obtained when compared to its value at the initial point (Eq. (40) with $\tau = 10^{-1}$).

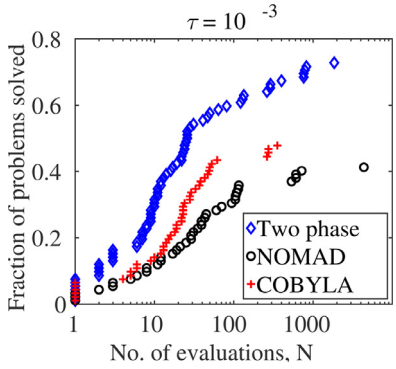


**Fig. 10.** Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that a significant reduction in the final merit function is obtained when compared to its value at the initial point (Eq. (40) with $\tau = 10^{-3}$).

Search algorithm in C++ and has an option to deal with constraints using extreme barrier, filter technique or progressive barrier approach. Progressive barrier option is chosen to handle constraints while comparing on the numerical problems. All other parameters were set to default except the maximum number of evaluations is allowed to be 10000.

COBYLA (Powell, 1994b), on the other hand, approximates the objective function and the constraints using linear functions that interpolated at the vertices of a simplex. The algorithm is based on a trust-region framework and the new point obtained at each iteration either improves the geometry of the simplex or improves a merit function. Default parameters are used except the maximum number of evaluations are allowed to be 10,000 and the lower bound on the trust-region size is set to $10^{-6}$.

The data profile based on convergence criteria in (37) is given in Fig. 7. Our proposed two phase approach solves 56 of the 92 numerical problems, while COBYLA and NOMAD is able to attain a solution within 1% of the global minima for 38 and 33 problems, respectively. The two phase method is not only better in terms of obtaining global minima for this set of problems, it is also competitive in terms of number of evaluations. NOMAD in general takes more number of evaluations indicating that a model-based approach is more suited for smooth problems. Note that COBYLA approximates the objective function and the constraints using linear models and is able to perform marginally better than NOMAD.

The performance profile is given in Fig. 8. Note that for $r_{t,s} = 1$, the proposed approach performs better or the same for 60 of the 92 problems. The plot of data and performance profile for convergence test (40) is presented next. These plots are based on three different required levels of reduction in the merit function: $\tau = 10^{-1}, 10^{-3}, 10^{-6}$. The data profiles are given in Fig. 9 based on convergence test (40) with the requirement that a solution is obtained which achieves 90% reduction in the merit function compared to maximum reduction possible. As seen in Fig. 9, the proposed two phase algorithm satisfies this criteria for 78 out of 92 problems while COBYLA and NOMAD satisfy the convergence test for 53 and 59 problems, respectively. This indicates that even though, there are certain problems which we are not able to solve to global optimality, significant reduction in at least constraint violation is achieved. When $\tau$ is decreased further to $10^{-3}$, as expected, the performance deteriorates for all the algorithms as illustrated in Fig. 10. However, our approach still outperforms the other two and COBYLA solves more number of problems compared to NOMAD. Decreasing $\tau$ further to $10^{-6}$ reduces the number of problems that satisfy the convergence test (40) but the rank of performance of the solver still remains the same (Fig. 11).

The performance profile with convergence test (40) is given in Fig. 12(a)–(c) for $\tau = 10^{-1}, 10^{-3}$ and $10^{-6}$, respectively. For $\tau = 10^{-1}$, at small alpha values, proposed algorithm slightly outperforms COBYLA but with increasing $\alpha$, COBYLA performs better. NOMAD performs inferior compared to the other methods on these set of test problems. When the required level of reduction is increased, our method outperforms other methods and this can be observed in Fig. 12(b) and (c).

The proposed approach is efficient in terms of the required number of evaluations. This is due to the use of an efficient model improvement algorithm which replaces no more than 2 interpolating points whenever the geometry of the samples deteriorates. The success of our method can also be attributed to the use of global solver (ANTIGONE in this case) for solving the sub-problems, the algorithm for improving the geometry of the samples, and the initialization of the initial trust-region for the entire space. The

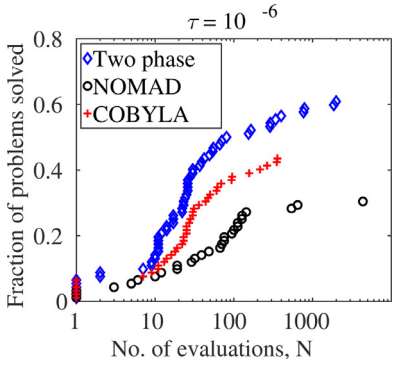


**Fig. 11.** Data profile indicating the fraction of problems solved with number of evaluations based on the convergence test that a significant reduction in the final merit function is obtained when compared to its value at the initial point (Eq. (40) with $\tau = 10^{-6}$).

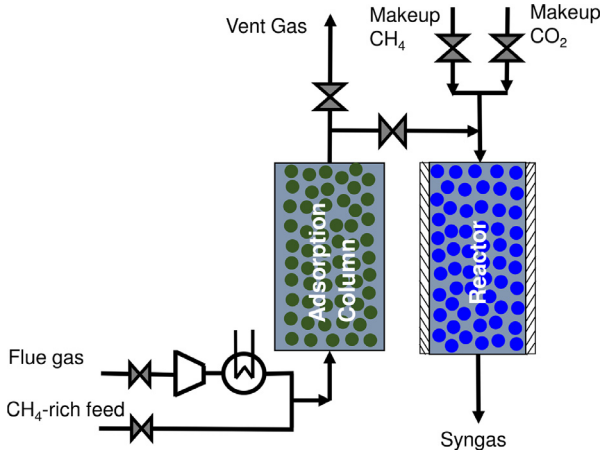


**Fig. 13.** Schematic of the process for integrated carbon capture and conversion.

initial trust-region allows for globally exploring the search space, while using a global solver gives an indication of the global solution if the surrogate model is able to approximate the original model well in the limited number of initial samples. However, using a global solver is computationally expensive. We compared the performance of the algorithm on a few problems using local and global solver to solve the subproblems. We observed that although using local solver is computationally inexpensive, we converged to the global minima for more problems for the cases when the global solver is used. However, this is not conclusive and requires further investigation.

### *4.3. Optimal design of a cyclic and integrated carbon capture and conversion process*

A chemical engineering case study is now presented which has combination of known and unknown analytical form of the constraints with black-box constraints. Due to emergence of process intensification (Stankiewicz et al. (2000), Demirel et al. (2017)) efficient DFO algorithms are needed more than ever. Process intensification merges processes with different aims to lessen number of units in order to reduce cost and/or energy consumption. In these cases, different phenomena interact in a complex manner and obtaining even the appropriate operating conditions and design that satisfies regulatory and performance constraints is not trivial (Iyer et al., 2017). Therefore, an optimization strategy is needed to asses if the intensified process is economically viable and that it meets quality and regulatory constraints. We proposed a multifunctional process that integrates adsorption and reaction to directly utilize $CO_2$ from flue gas to produce syngas in Iyer et al. (2017). We also optimized the operating conditions and design parameters of the process using a preliminary version of the current algorithm. This paper presents an improved optimization algorithm with more effective sampling strategy. Instead of performing multiple evaluations at each iteration, no more than 2 additional function calls are used.

The process configuration is given in Fig. 13. The flue gas contains large amount of $N_2$ (around 86%), while the rest is $CO_2$ and it is therefore important to separate $CO_2$ from $N_2$. The overall process contains two columns. The first is an adsorption column while second is a reactor. The process is configured such that $CO_2$ from flue gas is adsorbed more favorably than $N_2$ using an adsorbent. After adsorption, $CO_2$ is desorbed using methane by utilizing the gradient in concentration rather than pressure. This leads to energy savings since applying pressure swings is energy intensive. The adsorption and desorption steps are run in a cyclic manner until a cyclic steady-state is achieved. The outlet of the adsorption column contains $CO_2$, $N_2$ and $CH_4$. It would be advantageous to remove as much $N_2$ as possible from the gas mixture and keeping the loss of $CO_2$ and $CH_4$ to a minimum. The gas is then sent to the reaction section filled with catalyst that enables dry reforming. $CO_2$ and $CH_4$ need to be mixed in a ratio conducive for dry reforming and is considered to be one of the parameters for optimization.

#### *4.3.1. Process modeling*

The process consists of two sections, namely adsorption and reaction. The adsorption section is dynamic in nature and is therefore, modeled by one-dimensional nonlinear algebraic partial differential equations (NAPDE). The model includes component-wise mass balance, energy balance and other relationships. The gas phase mass balance includes the accumulation, convection, axial dispersion and adsorption terms and is given by the following

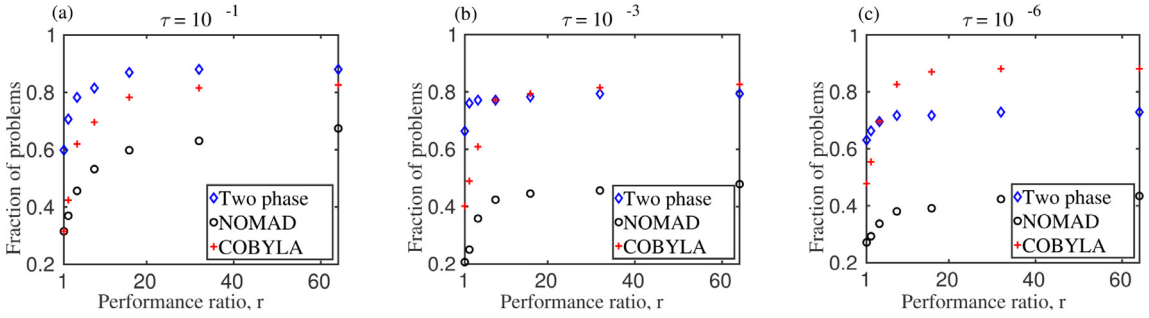


**Fig. 12.** Performance profile indicating the relative performance of the solvers based on the convergence test that a significant reduction in the final merit function is obtained when compared to its value at the initial point (Eq. (40)) for (a) $\tau = 10^{-1}$, (b) $\tau = 10^{-3}$, (c) $\tau = 10^{-6}$.

equation:

$$\frac{\partial c_i}{\partial t} = -\frac{\partial}{\partial z}\left(-cD\frac{\partial y_i}{\partial z} + c_i v\right) - \frac{(1-e)}{e}\frac{\partial q_i}{\partial t} \quad \forall i \in \mathbb{G}_A \tag{41}$$

Here $c_i$ and $y_i$ denotes the bulk concentration and mole fraction of the gaseous species and $q_i$ is the loading on the solid adsorbent. $\mathbb{G}_A$ is the set of species in adsorption column, $\mathbb{G}_A = \{CO_2, CH_4, N_2\}$. The variation of the pressure with time and along the bed is given by:

$$\frac{\partial P}{\partial t} = \frac{P}{T}\frac{\partial T}{\partial t} - T\frac{\partial}{\partial z}\left(\frac{Pv}{T}\right) - RT\frac{(1-e)}{e}\sum_{i\in\mathbb{G}_A}\frac{\partial q_i}{\partial t} \tag{42}$$

where $P$ and $T$ are the column pressure and temperature respectively. Assuming gas and adsorbent to be at the same temperature, energy balance of the streams are given by:

$$\begin{aligned}&(1-e)\left(\rho_a C_{p,ad}\frac{\partial T}{\partial t} + C_{p,g}\frac{\partial\left(\sum_{i\in\mathbb{G}_A} q_i T\right)}{\partial t}\right) + eC_{p,g}\frac{\partial(\rho_g T)}{\partial t}\\ &\quad + eC_{p,g}\frac{\partial(\rho_g v T)}{\partial z}\\ &= K\frac{\partial^2 T}{\partial z^2} + (1-e)\sum_{i\in\mathbb{G}_A}(-\Delta H_i)\frac{\partial q_i}{\partial t} - \frac{2h_{ic}}{r_{ic}}(T - T_c)\end{aligned} \tag{43}$$

The heat will also be transferred across the column wall and is given by Eq. (44) as follows.

$$\rho_c C_{p,c}\frac{\partial T_c}{\partial t} = K_c\frac{\partial^2 T_c}{\partial z^2} + \frac{2r_{ic}h_{ic}}{r_{oc}^2 - r_{ic}^2}(T - T_c) - \frac{2r_{oc}h_{oc}}{r_{oc}^2 - r_{ic}^2}(T_c - T_{am}) \tag{44}$$

where $T_c$ is the wall temperature of the column and $T_{am}$ is the ambient temperature in K respectively.

The linear driving force equation describing the rate of gas adsorption into the adsorbent is given by the following equation.

$$\frac{\partial q_i}{\partial t} = k_i(q_i^* - q_i) \quad \forall i \in \mathbb{G}_A \tag{45}$$

where $k_i$ is the lumped mass transfer coefficient and $q_i^*$ is the equilibrium gas loading. For a complete description of the NAPDE model with other empirical relations and boundary conditions and values of parameters used, the readers are referred to Eqs. 9–19 of Iyer et al. (2017).

The outlet gas mixture from the adsorption column is mixed with additional $CO_2$ and $CH_4$. The primary reactions considered in the conversion of $CO_2$ and $CH_4$ to methane is the dry reforming and the reverse water gas shift reaction:

$$\begin{aligned}&CH_4 + CO_2 \Leftrightarrow 2CO + 2H_2 \quad \Delta H_{298K} = 247 kJ/mol\\ &CO_2 + H_2 \Leftrightarrow CO + H_2O \quad \Delta H_{298K} = 41.7 kJ/mol\end{aligned} \tag{46}$$

The dry reforming reaction is endothermic and produces more number of moles in the product as compared to the reactants. So, according to Le Chatelier's principle, the reaction is favorable at high temperature and low pressure. Since operation of adsorption section is favored at low temperature, the outlet from the adsorption column is heated in the reaction section for favorable conversion. Although, the adsorption process is dynamic, the adsorption column outlet gas over a cycle is mixed with additional $CO_2$ and $CH_4$ and the mixture is then fed to the reactor at constant flow rate and composition. The reactor is thus operated at steady state and is modeled by one dimensional pseudohomogenous model assuming plug flow and isothermal behavior. The component mass balances in the reactor is given by:

$$\frac{dF_i^R}{dz} = \rho_r A_r r_i^g \quad \forall i \in \mathbb{G}_R \tag{47}$$

**Table 4**
Decision variables for Problem (62).

| Decision variables | $x^L$ | $x^U$ |
|---|---|---|
| Pressure set at the outlet of adsorption column ($P_{OA}$)[bar] | 1 | 10 |
| Length of adsorption column ($L_a$)[m] | 0.5 | 2.5 |
| Reactor temperature ($T_c$) [K] | 373 | 1223 |
| Reactor bed length ($L_r$) [m] | 0.5 | 10 |
| Duration of step 1 ($t_{1s}$) [s] | 10 | $t_{ct}$ |
| Total cycle time ($t_{ct}$) [s] | 10 | 200 |
| Venting start time ($t_{v1}$) [s] | 0 | $t_{ct}$ |
| Venting end time ($t_{v2}$) [s] | 0 | $t_{ct}$ |
| Makeup $CO_2$ before reaction ($F_{CO_2}^M$) [mol/s] | 0 | 5 |
| Makeup $CH_4$ before reaction ($F_{CH_4}^M$) [mol/s] | 0 | 5 |

where $F_i^R$ is the species flow rate and $r_i$ is the rate of generation of each species. The heat duty required ($Q_r$) to maintain isothermal operation of the reactor is obtained by integrating the following equation describing the heat consumed by the reactions.

$$\frac{dQ_r}{dz} = \rho_r A_r \sum_m (-\Delta H_m) R_m \tag{48}$$

The pressure variation along the bed length of the reactor ($P_r$) is given by Ergun equation.

$$\frac{dP_r}{dz} = -\frac{\rho_{rg} v_r (1-e_r)}{d_{ce} e_r^2} f \tag{49}$$

Here $\rho_{rg}$ is the density of the gas phase in kg/m$^3$, $v_r$ is the superficial velocity in m/s and $f$ is the friction factor. Other important empirical relations, rate of reactions and boundary conditions are given in Eqs. 22–26, 28–31 and 34–36 in Iyer et al. (2017).

#### 4.3.2. Process simulation and constraints

The performance of the process is evaluated based on several metrics. Two very important metrics are utilization of $CO_2$ and cost of the process. Since the process steps involve venting of gases, it needs to be ensured that the greenhouse gases ($CO_2$ and $CH_4$) are emitted below those allowed by regulatory agencies. Additionally, in order for syngas to be utilized for further processes, it should be of high quality with adequate ratio of CO and $H_2$.

As mentioned earlier, the adsorption process is described by NAPDE model. To solve the model, the partial differential equations are spatially discretized using upwind differencing scheme to form a set of ordinary differential equations (ODEs). These set of ODEs are solved using ode23s in MATLAB. On solving the model, concentration and temperature profiles are obtained along the adsorption column length and over time. The adsorption process is a cyclic two-step process. During the first step, flue gas enters the column and $CO_2$ is selectively adsorbed. In the second step, $CH_4$-rich feed enters the column and $CO_2$ is desorbed due to concentration driving force. The initial condition of the column during second step is the final condition in the first step. The two steps are repeated in a cyclic manner until cyclic steady state (CSS) is attained. The input conditions for the reactor are calculated based on the solution obtained at CSS. The reactor model is given by ordinary differential equations (ODEs) and are solved using ode23s. For obtaining the simulations, the number of spatial discretizations and cycles used are both set to five.

Iyer et al. (2017) utilized simulations to study the performance metrics of the process and identified key optimization parameters. The key decision variables along with the bounds are listed in Table 4. Note that the process needs to occur in a chronological order such that it makes logical sense. The following set of unrelaxable constraints always need to be satisfied to ensure that the simulation does not fail:

$$10 \le t_{1s} \le t_{ct} \tag{50}$$

$$10 \leq t_{ct} - t_{1s} \leq t_{ct} \tag{51}$$

$$0 \leq t_{v1} \leq t_{ct} \tag{52}$$

$$0 \leq t_{v2} \leq t_{ct} \tag{53}$$

$$10 \leq t_{v2} - t_{v1} \leq t_{ct} \tag{54}$$

Eq. (50) states that the first step time does not exceed the cycle time. Eq. (51) puts a constraint that the second step should be less than the cycle time. The lower bound in Eqs. (50) and (51) allows for the first and second step to run for at least 10 s. Similarly, the start and end time of venting should always be less than the cycle time is given in Eqs. (52) and (53). Eq. (54) states that the difference in the start and ending times should be at least 10 seconds. In other words, the venting can not end before starting. The upper bound will always be satisfied due to the relations (52) and (53).

Note that the constraints (50)–(54) are expressed explicitly in terms of the decision variables. In this problem, there are constraints which can not be explicitly expressed in terms of the decision variables unless the model is discretized. Such constraints are referred as black-box constraints and the value of the constraint is obtained as a result of a simulation. Unlike constraints in (50)–(54), violating the black-box constraints will not result in a simulation failure. These constraints are given as follows:

$$\frac{l_{CH_4}^{max} - 10}{100} \leq 0 \tag{55}$$

$$\frac{U_{CO_2}^{min} - 90}{100} \geq 0 \tag{56}$$

$$SG^{min} \geq 0.9 \tag{57}$$

$$SG^{max} \leq 1.1 \tag{58}$$

$$y_{CH_4}^{msg} \leq 0.03 \tag{59}$$

$$y_{CO_2}^{msg} \leq 0.03 \tag{60}$$

$$y_{N_2}^{msg} \leq 0.1 \tag{61}$$

Eq. (55) gives an upper bound on the methane loss from the process since it is a greenhouse gas and will violate regulatory constraints. The constraint on the minimum overall utilization of $CO_2$ is given in Eq. (56). The requirement on minimum and maximum ratio of $H_2$/CO is given by inequalities (57) and Eq. (58). It is important to maintain this ratio so that syngas can be used in further downstream processes. The produced syngas should be of high quality with minimal $CO_2$, $CH_4$ and $N_2$ and it is imposed by Eqs. (59)–(61). Since the process is novel with large number of variables and complex interactions between the variables and constraints it is not trivial to obtain a feasible point using parametric studies. We are interested in two key performance metrics for the process. We want to obtain the design and operating parameters that yields minimum cost per ton of syngas produced and secondly maximize overall $CO_2$ utilization. The two objectives are black-box and the details of calculation procedure can be found in Iyer et al. (2017). The resulting optimization problem is a grey-box problem. Rigorous optimization needs to be performed to balance different trade-offs.

**Table 5**
Initial guess for Problem (62).

| Decision variables | $x^0$ |
|---|---|
| $P_{OA}$ | 1.001 |
| $L_a$ | 0.71 |
| $T_c$ | 478.69 |
| $L_r$ | 4.05 |
| $t_{1s}$ | 135.85 |
| $t_{ct}$ | 149.92 |
| $t_{v1}$ | 29.76 |
| $t_{v2}$ | 122.87 |
| $F_{CO_2}^M$ | 1.26 |
| $F_{CH_4}^M$ | 0 |

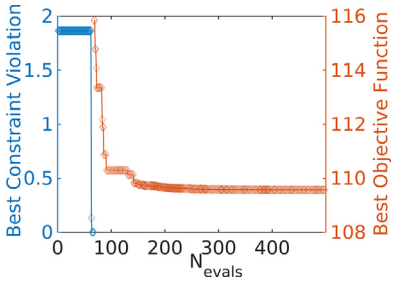


**Fig. 14.** Progress of the best constraint violation and the best objective function value with number of evaluations for case 1.

The optimization problem is formulated as follows:

$$\begin{aligned} \min_{x} \quad & f(x,z) \\ s.t. \quad & g_i(x,z) \leq 0 \quad \forall i \in \{1,\ldots,p\} (Eq.55-61) \\ & g_j(x) \leq 0 \quad \forall j \in \{1,\ldots,q\} (Eq.50-54) \\ & h_u(x,z) = 0 \quad \forall u \in \{1,\ldots,s\} (\text{process model}) \\ & x \in [x^L, x^U] \end{aligned} \tag{62}$$

where $z$ represents intermediate variables in the process model. The variables $z$ can be thought of as the black-box part of the optimization problem (62). The model equations $h(x,z) = 0$ are solved for $z$ for given value of $x$ and thereby, the objective function and constraint values ($g_i(x, z)$) are obtained.

*4.3.3. Optimization results*

The following four case studies are performed:

Case 1: Minimize cost of production of syngas using natural gas as methane source,
Case 2: Maximize overall $CO_2$ utilization using natural gas as methane source,
Case 3: Minimize cost of production of syngas using biogas as methane source,
Case 4: Maximize overall $CO_2$ utilization using biogas as methane source.

The sampling scheme given by Eq. (30) is used to generate initial samples such that they are feasible with respect to the known constraints given by Eqs. (50) – (54). Here we maintain $6n+1$ samples at all the iterations. Same initial guess was provided for all the case studies and is given in Table 5. The two phase algorithm is then applied to the problem to generate optimum results for the four cases. Fig. 14 shows the progress of the best objective function values and the constraint violations obtained with number of evaluations for case 1. The initial point is infeasible and, therefore, the feasibility phase is triggered which finds a feasible point within 68 evaluations. Once a feasible point is obtained, the optimization phase is started that reduces the objective function value. The algorithm ultimately provides the optimal design and operating conditions that gives the production cost to be $109.57. The value of

**Table 6**
Optimum results obtained for the four case studies with constraint violation and objective function at the initial point.

| Case Studies | $f(\hat{x})$ | $\hat{x}(P_{OA}, L_a, T_c, L_r, t_{1s}, t_{ct}, t_{v1}, t_{v2}, F^M_{CO_2}, F^M_{CH_4})$ | $N_{evals}$ | $\theta(x^0)$ | $f(x^0)$ |
|---|---|---|---|---|---|
| Case 1 | \$109.57 | (1, 0.8513, 1209.9, 0.5, 85.31, 95.31, 58.86, 91.3354, 5, 4.8) | 681 | 2.17 | $3.32 \times 10^5$ |
| Case 2 | 99.71% | (1, 2.49, 1223, 8.78, 12.19, 22.19, 9.69, 19.69, 4.48, 4.64) | 332 | 1.38 | 0.88 |
| Case 3 | \$110.27 | (1, 2.33, 1208.1, 0.5, 189.99, 199.99, 189.86, 199.95, 5, 4.90) | 662 | 2.2 | $4.14 \times 10^5$ |
| Case 4 | 99.75% | (1, 2.37, 1223, 4.96, 55.07, 65.07, 47,16, 65.07, 4.6, 5) | 244 | 1.41 | 1.017 |

**Table 7**
Comparison of solution provided by the Two phase algorithm, COBYLA and NOMAD.

| Case Studies | Two phase | | COBYLA | | NOMAD | |
|---|---|---|---|---|---|---|
| | $f(\hat{x})$ | $N_{evals}$ | $f(\hat{x})$ | $N_{evals}$ | $f(\hat{x})$ | $N_{evals}$ |
| Case 1 | \$109.57 | 681 | \$115.75 | 364 | $2.79 \times 10^{5*}$ | 628 |
| Case 2 | 99.71% | 332 | 99.6% | 206 | 97.59% | 1248 |
| Case 3 | \$110.27 | 662 | \$114.94 | 203 | \$109.61 | 1473 |
| Case 4 | 99.75% | 244 | 99.25% | 261 | 98.8% | 1382 |

* A feasible solution could not be obtained.

the objective function, the corresponding decision variables at the optimum are given in Table 6. The constraint violation and the objective function at the initial point and the number of evaluations required for obtaining the final optimal result are also reported for all the cases.

It is observed from Table 6 that the optimal pressure is at the lower bound throughout all the four cases. The algorithm, however, has increased the temperature of the reactor and the amounts of makeup $CO_2$ and $CH_4$ in order to increase the conversion of $CO_2$ and $CH_4$ to syngas while meeting the constraints on process metrics and improving the objective in each case. It is also observed that the temperature has reached the upper bound while maximizing $CO_2$ utilization in both the natural gas and biogas cases. This is because dry reforming is an endothermic reaction, hence, a high temperature is needed to increase the conversion of $CO_2$ and $CH_4$ to syngas. This helps in meeting constraints on loss of $CO_2$ and $CH_4$, while improving overall utilization and reducing the total cost per ton of syngas. Although increase in temperature results in more $CO_2$ conversion and more syngas being produced, there are increased utility costs involved in maintaining the reactor at high temperature. This is observed from the optimal reactor temperature for the cost minimization problem for both the natural gas and biogas cases not reaching the upper bound (1223 K). Instead it is maintained at a high enough value (around 1208 K) while balancing different trade-offs. Similarly it is observed that while moderate to high values of reactor lengths are obtained at optimum for the maximizing $CO_2$ utilization case, the optimal reactor length is at the lower bound in the cost minimization case for both gases. All the decision variables relating to the step and venting times in each case meet the unrelaxable constraints listed in Eqs. (50)–(54). While the flue gas step durations and venting times at optimum are vary between the different cases, methane rich feed step duration however is at the lower bound of 10 s in all cases.

We also compare the performance of the algorithm on the case studies with COBYLA and NOMAD. For both the solvers, it is not possible to avoid simulations at points which violates Eqs. (50) – (54). The simulations fail if the design point do not satisfy these constraints. To avoid simulation failure, a very high arbitrary value ($10^{15}$) is assigned for the objective function and the constraints. All solvers are given the same initial guess listed in Table 5. Table 7 provides a summary of the results. COBYLA is economical in terms of function evaluations compared to the two phase algorithm but does not provide a better objective function value. NOMAD, on the other hand takes a lot of function evaluations. Furthermore, for case 1, a feasible point could not be obtained. However, NOMAD finds a point that gives the least objective function value compared to all other algorithms for case 3.

In summary, the two phase algorithm has been able to obtain a feasible point from a high initial constraint violation and then improve the objective function value in each case for a complex chemical process model involving both explicit and black-box constraints and inter-dependent decision variables. Thus, the use of this algorithm with such novel integrated process designs can greatly help in identifying the feasible and optimal operating regions which may otherwise be difficult to predict based on just parametric studies of individual variables. This can accelerate and pave the way for effective feasibility analyses of novel integrated process technologies.

## 5. Conclusion

We presented a trust-region based two phase algorithm for constrained black-box and grey-box problems. The first phase of the algorithm finds a feasible point if the initial point is infeasible while the second phase decreases the objective function. We also observed that feasibility phase is critical for superior performance of the algorithm. A maximum of two samples are replaced at each iteration using model improvement algorithm to improve the geometry of the samples and make the model fully linear. The samples are economically used and this makes the algorithm suitable for problems with computationally expensive simulations. The efficiency of the algorithm is shown by comparing with two widely used solvers namely, NOMAD and COBYLA. The performance of the solvers are compared by constructing data and performance profile. Two convergence tests were used to plot data and performance profile. The two-phase algorithm not only finds the global minima for more number of problems compared to other solvers but achieves that in economical number of evaluations. The algorithm is also applied to optimize a cyclic and integrated carbon capture and conversion process. Four case studies are solved involving maximization of $CO_2$ utilization and cost minimization with biogas and natural gas used as methane source. Starting from a point with high infeasibility, the algorithm is able to obtain design and operation parameters that enables the process technology to be effective for $CO_2$ utilization. NOMAD and COBYLA were also applied to the integrated carbon capture and conversion case study. On comparing the performance of the three algorithms, it was observed that the proposed approach gives a high quality solution using economical number of evaluations.

## Acknowledgments

We gratefully acknowledge support from U.S. National Science Foundation (award number CBET-1606027). Part of the research was conducted with the computing resources provided by Texas A&M High Performance Research Computing.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.compchemeng.2017.12.011

## References


Abramson, M.A., Audet, C., Couture, G., Dennis Jr, J. E., Le Digabel, S., Tribes, C., 2011. The NOMAD project.

Abramson, M.A., Audet, C., Dennis Jr, J.E., Digabel, S.L., 2009. OrthoMADS: a deterministic mads instance with orthogonal directions. SIAM J. Optim. 20 (2), 948–966.

Agarwal, A., Biegler, L.T., Zitney, S.E., 2010. A superstructure-based optimal synthesis of psa cycles for post-combustion co2 capture. AIChE J. 56 (7), 1813–1828.

Arouxét, M.B., Echebest, N.E., Pilotta, E.A., 2015. Inexact restoration method for nonlinear optimization without derivatives. J. Comput. Appl. Math. 290, 26–43.

Audet, C., Béchard, V., Chaouki, J., 2008. Spent potliner treatment process optimization using a MADS algorithm. Optim. Eng. 9 (2), 143–160.

Audet, C., Conn, A.R., Le Digabel, S., Peyrega, M., 2016. A progressive barrier derivative-free trust-region algorithm for constrained optimization. Technical Report. Technical Report G-2016-49, Les cahiers du GERAD.

Audet, C., Dennis Jr, J.E., 2004. A pattern search filter method for nonlinear programming without derivatives. SIAM J. Optim. 14 (4), 980–1010.

Audet, C., Dennis Jr, J.E., 2006. Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. 17 (1), 188–217.

Audet, C., Dennis Jr, J.E., 2009. A progressive barrier for derivative-free nonlinear programming. SIAM J. Optim. 20 (1), 445–472.

Augustin, F., Marzouk, Y., 2014. NOWPAC: A provably convergent derivative-free nonlinear optimizer with path-augmented constraints. arXiv preprint arXiv:1403.1931.

Bajaj, I., Hasan, M.M.F., 2016. Effective sampling, modeling and optimization of constrained black-box problems. Comput. Aided Process Eng. 38, 553–558.

Bhosekar, A., Ierapetritou, M., 2017. Advances in surrogate based modeling, feasibility analysis and and optimization: a review. Comput. Chem. Eng.

Boukouvala, F., Floudas, C.A., 2017. ARGONAUT: algorithms for global optimization of constrained grey-box computational problems. Optim. Lett. 11 (5), 895–913.

Brekelmans, R., Driessen, L., Hamers, H., Den Hertog, D., 2005. Constrained optimization involving expensive function evaluations: a sequential approach. Eur. J. Oper. Res. 160 (1), 121–138.

Caballero, J.A., Grossmann, I.E., 2008. An algorithm for the use of surrogate models in modular flowsheet optimization. AIChE J. 54 (10), 2633–2650.

Conejo, P., Karas, E.W., Pedroso, L., 2015. A trust-region derivative-free algorithm for constrained optimization. Optim. Methods Softw. 30 (6), 1126–1145.

Conn, A.R., Gould, N.I.M., Toint, P.L., 2000. Trust Region Methods, 1. SIAM.

Conn, A.R., Scheinberg, K., Vicente, L.N., 2008. Geometry of interpolation sets in derivative free optimization. Math. Programm. 111 (1–2), 141–172.

Conn, A.R., Scheinberg, K., Vicente, L.N., 2009. Global convergence of general derivative-free trust-region algorithms to first-and second-order critical points. SIAM J. Optim. 20 (1), 387–415.

Conn, A.R., Scheinberg, K., Vicente, L.N., 2009. Introduction to Derivative-Free Optimization, 8. SIAM.

Custódio, A.L., Vicente, L.N., 2007. Using sampling and simplex derivatives in pattern search methods. SIAM J. Optim. 18 (2), 537–555.

Dakota, A., 2009. Multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis. Sandia National Laboratories, SAND2010-2183.

Demirel, S.E., Li, J., Hasan, M.M.F., 2017. Systematic process intensification using building blocks. Comput. Chem. Eng. 105, 2–38.

Diniz-Ehrhardt, M., Martínez, J., Pedroso, L.G., 2011. Derivative-free methods for nonlinear programming with general lower-level constraints. Comput. Appl. Math. 30 (1), 19–52.

Eason, J.P., Biegler, L.T., 2016. A trust region filter method for glass box/black box optimization. AIChE J. 62 (9), 3124–3136.

Fahmi, I., Cremaschi, S., 2012. Process synthesis of biodiesel production plant using artificial neural networks as the surrogate models. Comput. Chem. Eng. 46, 105–123.

Fernandes, F.A., 2006. Optimization of fischer-tropsch synthesis using neural networks. Chem. Eng. Technol. 29 (4), 449–453.

First, E.L., Hasan, M.M.F., Floudas, C.A., 2014. Discovery of novel zeolites for natural gas purification through combined material screening and process optimization. AIChE J. 60 (5), 1767–1785.

Fletcher, R., Leyffer, S., 2002. Nonlinear programming without a penalty function. Math. Programm. 91 (2), 239–269.

Garud, S.S., Karimi, I.A., Kraft, M., 2017. Design of computer experiments: a review. Comput. Chem. Eng. 106, 71–95.

Gilmore, P., Kelley, C.T., 1995. An implicit filtering algorithm for optimization of functions with many local minima. SIAM J. Optim. 5 (2), 269–285.

GlobalLib, 2015. Global library. http://www.gamsworld.org/global/globallib.htm.

Gould, N.I., Toint, P.L., 2010. Nonlinear programming without a penalty function or a filter. Math. Programm. 122 (1), 155–196.

Graciano, J., Le Roux, G., 2013. Improvements in surrogate models for process synthesis. application to water network system design. Comput. Chem. Eng. 59, 197–210.

Gutmann, H.-M., 2001. A radial basis function method for global optimization. J. Global Optim. 19 (3), 201–227.

Hansen, N., 2016. The CMA evolution strategy: a tutorial. arXiv preprint arXiv:1604.00772.

Hasan, M.M.F., First, E.L., Floudas, C.A., 2013. Cost-effective $CO_2$ capture based on in silico screening of zeolites and process optimization. Phys. Chem. Chem. Phys. 15 (40), 17601–17618.

Henao, C.A., Maravelias, C.T., 2011. Surrogate-based superstructure optimization framework. AIChE J. 57 (5), 1216–1232.

Hickernell, F.J., 1998. Lattice Rules: How Well Do They Measure Up?. Springer.

Huyer, W., Neumaier, A., 2008. SNOBFIT–stable noisy optimization by branch and fit. ACM Trans. Math. Softw. (TOMS) 35 (2), 9.

Iyer, S.S., Bajaj, I., Balasubramanian, P., Hasan, M.M.F., 2017. Integrated carbon capture and conversion to produce syngas: novel process design, intensification and optimization. Ind. Eng. Chem. Res.

Jia, Z., Davis, E., Muzzio, F.J., Ierapetritou, M.G., 2009. Predictive modeling for pharmaceutical processes using kriging and response surface. J. Pharm. Innovation 4 (4), 174–186.

Kawajiri, Y., Biegler, L.T., 2006. Optimization strategies for simulated moving bed and powerfeed processes. AIChE J. 52 (4), 1343–1350.

Koziel, S., Ciaurri, D., Leifsson, L., 2011. Surrogate-based methods. Comput. Optim. Methods Algorithms 33–59.

Le Digabel, S., 2011. Algorithm 909: NOMAD: nonlinear optimization with the mads algorithm. ACM Trans. Math. Softw. 37 (4), 44:1–44:15. doi:10.1145/1916461.1916468.

Lewis, R.M., Torczon, V., Trosset, M.W., 2000. Direct search methods: then and now. J. Comput. Appl. Math. 124 (1), 191–207.

Liuzzi, G., Lucidi, S., 2009. A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an $\ell_\infty$ penalty function. SIAM J. Optim. 20 (1), 1–29.

Liuzzi, G., Lucidi, S., Sciandrone, M., 2010. Sequential penalty derivative-free methods for nonlinear constrained optimization. SIAM J. Optim. 20 (5), 2614–2635.

Meert, K., Rijckaert, M., 1998. Intelligent modelling in the chemical process industry with neural networks: a case study. Comput. Chem. Eng. 22, S587–S593.

Misener, R., Floudas, C.A., 2014. ANTIGONE: algorithms for coNTinuous/integer global optimization of nonlinear equations. J. Global Optim. 59 (2–3), 503–526.

Moré, J.J., Wild, S.M., 2009. Benchmarking derivative-free optimization algorithms. SIAM J. Optim. 20 (1), 172–191.

Nesterov, Y., et al., 2011. Random gradient-free minimization of convex functions. Technical Report. Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).

Nilchan, S., Pantelides, C., 1998. On the optimisation of periodic adsorption processes. Adsorption 4 (2), 113–147.

Palmer, K., Realff, M., 2002. Metamodeling approach to optimization of steady-state flowsheet simulations: model generation. Chem. Eng. Res. Des. 80 (7), 760–772.

Powell, M.J., 1994. A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Advances in optimization and numerical analysis. Springer, pp. 51–67.

Powell, M.J., 2006. The NEWUOA software for unconstrained optimization without derivatives. In: Large-scale nonlinear optimization. Springer, pp. 255–297.

Powell, M.J., 2009. The BOBYQA algorithm for bound constrained optimization without derivatives. Cambridge NA Report NA2009/06, University of Cambridge, Cambridge.

Powell, M.J.D., 1994. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. Springer Netherlands, Dordrecht, pp. 51–67.

Regis, R.G., 2014. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Eng. Optim. 46 (2), 218–243.

Regis, R.G., Shoemaker, C.A., 2013. Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. Eng. Optim. 45 (5), 529–555.

Regis, R.G., Wild, S.M., 2017. CONORBIT: constrained optimization by radial basis function interpolation in trust regions. Optim. Methods Softw. 32 (3), 552–580.

Rios, L.M., Sahinidis, N.V., 2013. Derivative-free optimization: a review of algorithms and comparison of software implementations. J. Global Optim. 56 (3), 1247–1293.

Rogers, A., Ierapetritou, M., 2015. Feasibility and flexibility analysis of black-box processes part 2: surrogate-based flexibility analysis. Chem. Eng. Sci. 137, 1005–1013.

Sampaio, P.R., Toint, P.L., 2015. A derivative-free trust-funnel method for equality–constrained nonlinear optimization. Comput. Optim. Appl. 61 (1), 25–49.

Scheinberg, K., Toint, P.L., 2010. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. SIAM J. Optim. 20 (6), 3512–3532.

Sergeyev, Y.D., Kvasov, D.E., 2017. Lipschitz global optimization. In: Deterministic Global Optimization. Springer, pp. 1–17.

Stankiewicz, A.I., Moulijn, J.A., et al., 2000. Process intensification: transforming chemical engineering. Chem. Eng. Progress 96 (1), 22–34.

Vaz, A.I.F., Vicente, L.N., 2007. A particle swarm pattern search method for bound constrained global optimization. J. Global Optim. 39 (2), 197–219.

Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Programm. 106 (1), 25–57.

Wild, S.M., Regis, R.G., Shoemaker, C.A., 2008. ORBIT: optimization by radial basis function interpolation in trust-regions. SIAM J. Sci. Comput. 30 (6), 3197–3219.

Won, K.S., Ray, T., 2005. A framework for design optimization using surrogates. Eng. Optim. 37 (7), 685–703.

Yang, X.-S., Gandomi, A.H., Talatahari, S., Alavi, A.H., 2012. Metaheuristics in Water, Geotechnical and Transport Engineering. Newnes.