

Efficient Strategies for Constrained Black-box Optimization by Intrinsically Linear Approximation (CBOILA)

Chengyang Liu · Zhiqiang Wan* · Xuewu Li · Dianzi Liu*

Abstract In this paper, a novel trust-region based surrogate-assisted optimization method, called CBOILA (Constrained Black-box Optimization by Intrinsically Linear Approximation), has been proposed to reduce the number of black-box function evaluations and enhance the efficient performance for solving complex optimization problems. This developed optimization approach utilizes an assembly of intrinsically linear approximations to seek the optimum with incorporation of three strategies: (1) extended-box selection strategy (EBS), (2) global intelligence selection strategy (GIS) and (3) balanced trust-region strategy. EBS aims at reducing the number of function evaluations in current iteration by selecting points close to the given trust region boundary. Whilst, GIS is designed to improve the exploration performance by adaptively choosing points outside the trust region. The balanced trust-region strategy works with four indicators, which will be triggered by the quality of the approximation, the movement direction of the search, the location of the sub-optimum, and the condition of the termination, respectively. By modifying the move limit of each dimension accordingly, CBOILA is capable of attaining a balanced search between exploitation and exploration for the optimal solutions. To demonstrate the potentials of the proposed optimization method, three widely used benchmark problems have been examined and the results have also been compared with solutions by other metamodel-based algorithms in published works. Results show that the proposed method can efficiently and robustly solve constrained black-box optimization problems within an acceptable computational time.

Keywords metamodel · trust region strategy · intrinsically linear approximation · constrained black-box optimization

* Corresponding authors: wzq@buaa.edu.cn and dianzi.liu@uea.ac.uk

Chengyang Liu · Dianzi Liu
School of Engineering, Faculty of Science, University of East Anglia, Norwich, UK
E-mail: dianzi.liu@uea.ac.uk

Zhiqiang Wan
School of Aeronautic Science and Engineering, Beihang University, Beijing, China

Xuewu Li
School of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an, China

1 Introduction

Due to the tremendous advances in computing power by supercomputers, numerical simulations performed by Finite Element (FE) analysis or coupled multidisciplinary analysis prove to be the most helpful, valuable and general technique to take the place of expensive experiments across engineering subjects, such as civil, structural, aerospace and automotive engineering. Meanwhile, to maximize the benefits brought from products, design optimization has also been applied in engineering to seek optimal solutions. However, there are three major challenges for optimization techniques applied to high-fidelity simulation models. First of all, numerical simulations in practical engineering are typically computationally expensive. For example, the single run of a whole-vehicle impact simulation that might involve one million finite elements usually takes more than 10 hours, even using a high-performance quad-core computer [1]. As a result, evolutionary methods such as particle swarm optimization (PSO) [2] and genetic algorithms (GAs) [3, 4] used to seek optimal designs are not suitable optimization techniques because they need lots of evaluations which are unaffordable and unrealistic in terms of computational time for solving real-world engineering design problems. Secondly, to the users the analysis of such numerical models by commercial software is like a black box problem. The derivative information is unavailable, unreliable, or impractical to obtain because little or no use of such computations is required in most finite differences-based methods. Last but not the least, real-world optimization problems usually have various complex properties including nonlinearity, non-convexity and multi-modality. As well documented in No Free Lunch theorem [5], without a knowable priori, it is difficult to choose an appropriate optimization algorithm with well configured parameters. In literature, substantial efforts have been devoted into tackling these challenges.

To alleviate the computational overhead, a metamodel-assisted technique proves to be a promising approach by replacing the expensive simulation calls with several simple forms of approximations. Generally, methods for metamodel building include polynomial response surface (PRS) [6], radial basis function (RBF) [7], kriging [8] and support vector regression (SVR) [9]. Based on each metamodel's predominant characteristics, Kriging and RBF belong to interpolation methods that are more appropriate for solving non-convex or multi-modal problems. Specifically, with the increased sampling points, kriging will be superior to RBF with regard to the accuracy of the approximation. PRS and SVR are regression methods that have advantages in dealing with convex problems [10]. PRS is much easier to implement than SVR but SVR outperforms PRS in fitting high dimensional responses.

A popular approach for solving black-box optimization problems is the surrogate-assisted evolutionary algorithm. Initially, surrogates were widely used for solving unconstrained problems [11–13]. However, this technique was still rarely used to approximate constraint functions [14]. Araujo [15] et al. implemented quadratic models in GAs to approximate the objective and constraint functions. Regis [16] used a trust-region-like RBF metamodel to refine the best solution for each generation in the Evolutionary Programming (EP) algorithm.

As demanded from various subjects to solve constrained black-box optimization (CBO) problems, metamodel-based iterative algorithms have gradually been developed. Brekelmans [17] applied linear approximations within sequential trust regions and used a filter method to select current iterates. Basudhar et al. [18] enhanced the ability of Efficient Global Optimization (EGO) for solving constrained problems by using SVR to approximate the boundary of the feasible domain. Koch et al. [19] proposed a new mechanism for Constrained Optimization by Radial Basis Function Approximation (COBRA) to repair violated designs and this technique was further developed by Bagheri [20].

The present work is motivated from the intrinsically linear interpolation [21–25], which are employed for metamodel-based optimization problems. However, little work has been studied to enhance the efficiency and stability of intrinsically linear approximation with the focus on the development of optimization framework to solve a constrained black-box optimization problem. Hence, a new optimization method called constrained black-box optimization by intrinsically linear approximation (CBOILA) has been proposed in this paper. With metamodeling and three effective strategies, this developed approach replaces the original complex optimization problem by a succession of simpler mathematical sub-problems. The solution of an individual sub-problem becomes the starting point for the next iteration and then, the move limits are changed and the optimization loop is repeated until the optimum is reached. In order to achieve the balance of exploitation and exploration during the entire optimization process, there are several important aspects addressed in CBOILA, namely:

- The plan of numerical experiments including the sampling and selection strategies (Section 3.2).
- The specific structure of intrinsically linear expressions for metamodel building (Section 3.3).
- The trust region strategy to balance the search between exploitation and exploration for the optimal solution (Section 3.4).

To demonstrate the capability of CBOILA for constrained black-box design optimization, the brief introduction of constrained black-box problem is described in Section 2 and the overview of the method is presented in Section 3.1. Comparisons with other state-of-the-art algorithms for solving three benchmark problems is shown in Section 4, and a final discussion is given in Section 5.

2 Constrained black-box optimization (CBO) problem

The main characteristic of a CBO problem is that there are no algebraic expressions for both the objective and the constraint functions. In other words, the functional relationships between design variables and responses are implicit. Generally, solving the black-box problem is computationally expensive. Moreover, the derivative information is also unavailable or impractical to obtain when the function evaluation is time-consuming. It is quite common in engineering to use simulation models for prediction of responses of the interests. In Fig 1, a set of design variables $\boldsymbol{x} \in Q \subseteq \mathbb{R}^d$ are inputted to the black-box, e.g.,

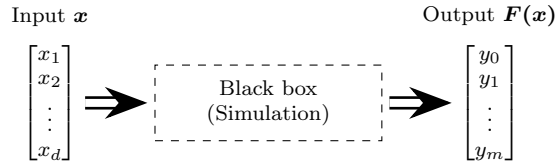


Fig. 1: Black-box optimization.

a simulation tool, and a certain set of responses $\mathbf{F}(\mathbf{x}) \subseteq \mathbb{R}^{m+1}$ are the outputs predicted by the unknown relationships between the variables and responses.

The CBO problem addressed in this paper can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{x} \in Q} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_j(\mathbf{x}) \leq 1 \quad (j = 1, \dots, m) \\ & A_i \leq x_i \leq B_i \quad (i = 1, \dots, d) \end{aligned} \quad (1)$$

95 where \mathbf{x} refers to the vector of design variables; Q is the design space bounded by $[\mathbf{A}, \mathbf{B}]$, A_i and B_i are the given lower and upper bounds for the design variable x_i ; d is the total number of design variables; $f(\mathbf{x})$ is the objective function; $g_j(\mathbf{x})$ is the constraint function and m is the total number of constraint functions.

100 In this paper, only minimization problems are considered because it is straightforward to convert maximization to minimization problems. It is worth noting that the constraints in Eq. 1 should be well normalized by multiplying or dividing by a certain factor based on the response values at the initial stage. Thus variations of constraint violations near the optimum can be ranged at the same level.

105 3 Constrained black-box optimization by intrinsically linear approximation (CBOILA)

3.1 Overview

CBOILA is a trust-region based metamodel-assisted optimization method, which attempts to solve a sequence of constrained optimization sub-problems in a series of trust regions using intrinsically linear approximations for evaluations of the objective function and constraints. The solution of an individual sub-problem becomes the starting point for the next iteration, then the trust region is adaptively panned or zoomed. This procedure is repeated until the search space is shrunk into a sufficiently small trust region that contains the current iterate as its interior point under the condition that the approximations are reasonably good. Each approximation is defined as a function of design variables as well as a number of tuning parameters. Tuning parameters are determined by the weighted least squares surface fitting, in which metamodels are built using the sampling points. Some of the sampling points are generated in the current trust region, and the others are taken from the pool of points utilized in the previous iterations.

110
115
120

In order to reduce the number of calls for evaluations required in the black-box optimization problem, CBOILA replaces the optimization problem (Eq. 1) by a series of approximate optimization problems like Eq. 2:

$$\begin{aligned}
 \min_{\mathbf{x} \in Q_k} \quad & \tilde{f}^k(\mathbf{x}) \\
 \text{s.t.} \quad & \tilde{g}_j^k(\mathbf{x}) \leq 1 \quad (j = 1, \dots, m) \\
 & A_i^k \leq x_i \leq B_i^k, A_i^k \geq A_i, B_i^k \leq B_i \quad (i = 1, \dots, d)
 \end{aligned} \tag{2}$$

where k is the iteration number, $\tilde{f}^k(\mathbf{x})$ and $\tilde{g}_j^k(j = 1, \dots, M)$ are approximations for objective and constraint functions, which are adequately constructed in a current search sub-domain, i.e., a trust region defined by the side constraints A_i^k and B_i^k . Fig. 2 displays graphically how the trust regions are adaptively resized and moved towards the optimum in a 2D optimization problem.

The framework of CBOILA can be described in Algorithms 1-4. Details of the main steps in CBOILA for solving CBO problems are introduced in the following sections.

3.2 Design of Experiments (DOE)

In order to build a metamodel, a dataset of inputs (a vector set of design variables) and corresponding outputs (response values) are required. Design of Experiments (DOE) is applied to determine the distribution of sampling points across the design space in order to predict the best possible information from metamodels built with these experimental data.

As described in Section 3.1, CBOILA constructs metamodels within sequential trust regions, each of which is centered on the successive sub-optimum. During this process, the trust region of the interest adaptively moves or zooms and evaluations of new numerical experiments are performed at each iteration. Moreover, previous information of objectives and constraints is valuable and would not be abandoned during the optimization progression. In this paper, three efficient strategies including Maxmin Stochastic Sampling (MSS) for generating new experimental designs, Extended-box Selection (EBS) for taking full advantage of previous information, and Global Intelligence Selection (GIS) for attaining a balance of the search capability between exploitation and exploration, have been developed to enhance the accuracy of metamodels used in CBOILA. Details of these strategies are discussed respectively in the following subsections.

3.2.1 Maxmin Stochastic Sampling (MSS)

MSS is a variant of stochastic sampling technique and applies an additional constraint on the distance between any two points. Analytical tests [23] have shown that the MSS could improve the quality of approximations and as a result, the solution of the defined subproblem will be much accurate. The pseudocode of MSS is described in Algorithm 2.

Algorithm 1: CBOILA Framework

Input:

- (1) Starting point $\mathbf{x}_0 \in Q$
- (2) The default number of required sampling points n_p ($d+5$)
- (3) The relative size of initial trust region Δ_0 (0.25)
- (4) The relative size of the extended box δ (1.5)
- (5) The range of adaptive enlargement or reduction ϵ_e, ϵ_r ([1.1,1.75])
- (6) The maximum iteration number K ($30 \cdot d$)
- (7) The minimum relative size of trust region Δ_{min} (0.0005)

Output:
The best point \mathbf{x}_{opt} encountered through the optimization process.

for k **in** $range(0, K)$ **do**

Step 1. Design of Experiments (DOE)

if $k = 0$ **then**

$X_k = \text{MSS}(\mathbf{x}_0, n_p, \mathbf{A}_k, \mathbf{B}_k)$

else

$n_\delta, X_\delta = \text{EBS}(\mathbf{x}_0, \delta, n_p, \mathbf{A}_k, \mathbf{B}_k)$

$X_s = \text{MSS}(\mathbf{x}_0, n_p - 0.5 \cdot n_\delta, \mathbf{A}_k, \mathbf{B}_k)$

$X_g = \text{GIS}(\mathbf{x}_0, n_g, X_{all}, Q_\delta)$

$X_k = X_\delta \cup X_s \cup X_g$

Step 2. Black-box Evaluations

Objective: $f(X_k) \subseteq \mathbb{R}^{n_k}$.

Constraints: $g_j(X_k) (j = 1, \dots, m) \subseteq \mathbb{R}^{n_k}$.

Step 3. Metamodel Building

Build approximations using intrinsically linear function.

Approximate objective function: $\hat{f}^k(x)$.

Approximate constraint function: $\hat{g}_j^k(x) (j = 1, \dots, m)$.

Step 4. Solve the subproblem

Use Sequential Quadratic Programming (SQP) method to solve the approximate problem.

Find the optimal point x_{k+1} within current trust region.

Step 5. Trust Region Strategy

if $\Delta_k > \Delta_{min}$ **then**

Define the next trust region Q_{k+1} using the trust region strategy.

else

Traverse all the possible points for the global optimum.

Output \mathbf{x}_{opt} .

Break

Algorithm 2: Maxmin Stochastic Sampling (MSS)

Function $\text{MSS}(\mathbf{x}_0, n_s, \mathbf{x}_l, \mathbf{x}_u)$:

Input:

- (1) \mathbf{x}_0 : Starting point.
- (2) n_s : The number of sampling points needed.
- (3) \mathbf{x}_l : Lower bounds on x .
- (4) \mathbf{x}_u : Upper bounds on x .

Output:
The set of sampling points X_s .

Put \mathbf{x}_0 in the sampling pool X_s .

$Diag = \|\mathbf{x}_u - \mathbf{x}_l\|_2$.

for i **in** $range(0, n_s - 1)$ **do**

Generate a point \mathbf{x} satisfied:

$\frac{Dist}{Diag} \geq r$. // The initial value of r is 0.9

/* $Dist$ is the minimum distance between \mathbf{x} and points in X_s . */

/* r will be reduced after a certain number of generations. */

Put \mathbf{x} in the sampling pool X_s .

Return X_s

Algorithm 3: Extended-box Selection (EBS)**Function** EBS($\mathbf{x}_0, \delta, n_p, \mathbf{x}_l, \mathbf{x}_u$):**Input:**

- (1) \mathbf{x}_0 : Starting point
- (2) δ : The relative size of the extended box
- (3) n_p : The default number of required sampling points
- (4) \mathbf{x}_l : Lower bounds on \mathbf{x}
- (5) \mathbf{x}_u : Upper bounds on \mathbf{x}

Output:The set of sampling points X_k .Define the extended box $\implies Q_\delta = \delta \cdot \text{Box}(\mathbf{x}_l, \mathbf{x}_u)$.Choose n_δ points inside $Q_\delta \implies X_\delta \subseteq \mathbb{R}^{n_\delta \times d}$.**Return** n_δ, X_δ **Algorithm 4:** Global Intelligence Selection (GIS)**Function** GIS($\mathbf{x}_0, n_g, X_{all}, Q_\delta$):**Input:**

- (1) \mathbf{x}_0 : Starting point
- (2) n_g : The number of global points needed
- (3) X_{all} : The database of all sampling points

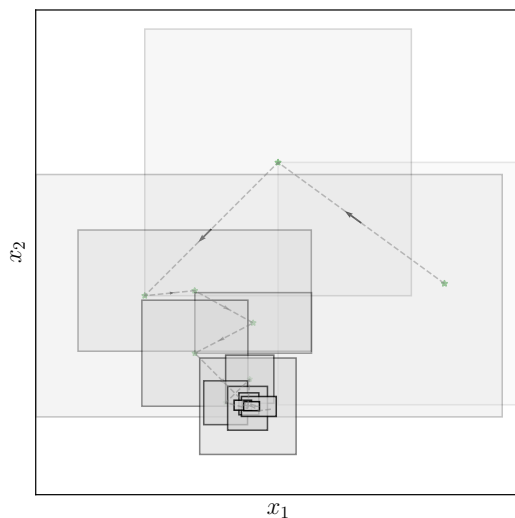
Output:The set of points X_g selected by GIS.**for** \mathbf{x}_i **in** X_{all} **do** **if** $\mathbf{x}_i \notin Q_\delta$ **then** $Dist_i = \|\mathbf{x}_0 - \mathbf{x}_i\|_2$.According to $Dist$, choose n_g points which are close to \mathbf{x}_0 . $X_g \subseteq \mathbb{R}^{n_g \times d}$.**Return** X_g 

Fig. 2: A typical optimization search path by CBOILA

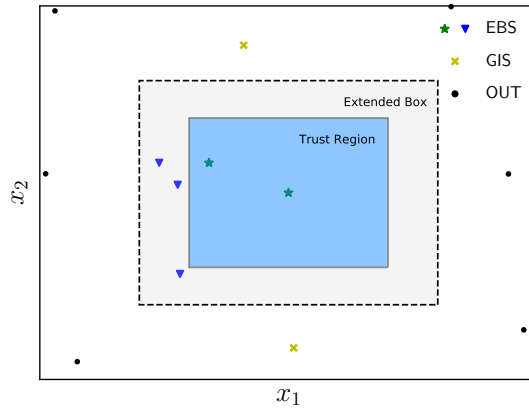


Fig. 3: Three effective strategies for selection of sampling points during the metamodel building process

3.2.2 Extended-box Selection (EBS)

When initial responses become available after the first iteration in CBOILA, a strategy called Extended-box Selection (See Alg. 3) is applied to select some points located in the neighborhood of the current starting point as alternatives to new points required in current iteration. As is shown in Fig. 3, this selection is reasonable because the size of this extended box is just 1.4 to 1.6 times larger than that of current trust region and it is possible that points in the extended box are also useful for building metamodels. Furthermore, to avoid deterioration of the approximation accuracy by overusing previous information, only half of the number of points selected by EBS ($0.5 \cdot n_\delta$) is deemed to be qualified for the alternatives. In other words, the number of new points required in current iteration is $n_k = n_p - 0.5 \cdot n_\delta$. If $n_\delta \geq n_p$, the generation of new points is not required in this iteration because existing points in the extended box are adequate for providing the approximations of responses with an acceptable level of accuracy in current trust region. Summarily, EBS can save the simulation cost without degrading the quality of the metamodel, which is constructed in an economical and practical way, especially when the computational budget is very limited.

3.2.3 Global Intelligence Selection (GIS)

To efficiently seek the global optimal solution, Global Intelligence Selection (GIS) is developed in CBOILA. The strategy for GIS shown in Algorithm 4 is applied to find the points which are close to the starting point and positioned outside of the extended box. These points depicted as the cross-shaped points in Fig. 3 are of global intelligence for two reasons. First, as compared with the current starting point, their locations are outside of the extended box. Second, they are selected from a set of the most nearest points to the current starting point. By achieving a balance in distance, points chosen by GIS can enhance the adaptive search ability of CBOILA. Without global information provided by GIS, CBOILA will possibly

185 trap into the local optima as the local points within the trust region could only support local exploitation. In addition, these global intelligence points could refine the approximation within the current trust region, resulting in efficiently moving the trust region along the most promising direction. It is worthwhile to note here that the number of global points should not be more than the number of points
 190 inside the extended box, otherwise, the quality of metamodel within the trust region can not be guaranteed. In general, the number of global intelligence points n_g is defined by Eq. 3 and will be rounded down during the entire optimization process.

$$n_g = \text{Int}[(0.5 + \frac{0.5}{k}K) \cdot n_p] \quad (3)$$

where k is the current iteration number and K is the maximum iteration number.

195 3.3 Metamodel building using intrinsically linear functions

In the present work, the process of constructing metamodels is based on combining different approximate models $\{\varphi_\ell(\mathbf{x})\}$ into one metamodel using Eq. 4 (note that the indices j and k are suppressed to simplify the notation).

$$\tilde{F}(\mathbf{x}) = \sum_{\ell=1}^{NF} b_\ell \cdot \varphi_\ell(\mathbf{x}) \quad (4)$$

where NF is the number of regressors in the model bank $\{\varphi_\ell(\mathbf{x})\}$ and b_ℓ is
 200 corresponding regression coefficient to each approximate model $\varphi_\ell(\mathbf{x})$.

This proposed assembly approach consists of two subsequent steps. In the first step, a single surrogate φ_ℓ is identified using the weighted least squares method as follows:

$$\sum_{s=1}^n w_s [F(\mathbf{x}_s) - \varphi_\ell(\mathbf{x}_s, \mathbf{a}_j)]^2 \rightarrow \min \quad (5)$$

where the coefficient w_s denotes the weight, in other words, the inequality of
 205 different sampling points, \mathbf{a}_j indicates the tuning parameter associated with the specific surrogate and \mathbf{x}_s is the sampling point. The simplest case of φ_ℓ is the first order polynomial metamodel and more complex ones are intrinsically linear functions (ILF) [26] that have been successfully applied for solving various design optimization problem [23–25]. ILF are nonlinear but they can be led to linear ones
 210 by simple transformations. Currently, five functions are considered in the regressor

pool $\{\varphi_\ell(\mathbf{x})\}$ as

$$\begin{aligned}
\varphi_1(\mathbf{x}) &= a_0 + \sum_{i=1}^d a_i x_i \\
\varphi_2(\mathbf{x}) &= a_0 + \sum_{i=1}^d a_i x_i^2 \\
\varphi_3(\mathbf{x}) &= a_0 + \sum_{i=1}^d a_i / x_i \\
\varphi_4(\mathbf{x}) &= a_0 + \sum_{i=1}^d a_i / x_i^2 \\
\varphi_5(\mathbf{x}) &= a_0 \prod_{i=1}^d x_i^{a_i}
\end{aligned} \tag{6}$$

Generally, the optimal solution of a generic optimization problem lies on a boundary of the feasible region. In other words, there is at least one constraint to be activated at the optimum. Therefore, it is reasonable to adaptively formulate the weighting coefficient w_s so that the approximation function $\tilde{F}(\mathbf{x})$ could be improved in terms of the accuracy near the promising region. Taking into account the above situation, w_s is defined as follows:

$$\begin{aligned}
w_s &= w_s^o \cdot w_s^j \quad (j = 1, \dots, m) \\
w_s^j &= \begin{cases} g_j^\alpha(\mathbf{x}) & \text{if } g_j(\mathbf{x}) \leq 1 \\ g_j^{-\alpha}(\mathbf{x}) & \text{if } g_j(\mathbf{x}) \geq 1 \end{cases} \\
w_s^o &= \left[\frac{f(\mathbf{x}_1)}{f(\mathbf{x}_s)} \right]^\beta
\end{aligned} \tag{7}$$

where \mathbf{x}_1 is the starting point, $\alpha = 4$ and $\beta = 1.5$ by default. w_s^o is the objective factor and w_s^j is the constraint factor. In the second step, different approximation models are assembled into one metamodel shown in Eq. 8, which is built in the same manner as Eq. 5. It should be noted here that the same DOE is applied when different approximation models φ_ℓ are constructed.

$$\sum_{s=1}^n w_s [F(\mathbf{x}_s) - \tilde{F}(\varphi_\ell(\mathbf{x}_s), \mathbf{b}_\ell)]^2 \rightarrow \min \tag{8}$$

Finally, applying this two-step metamodel building technique, a linear system of NF equations with NF unknowns \mathbf{b}_ℓ is solved.

3.4 Trust region strategy

Once the approximation functions have been constructed, the optimization problems (Eq. 2) can be solved using any mathematical or metaheuristic optimization techniques. During the entire optimization process, a strategy for updating the sub-region of the design space including its current dimensions and locations must be specified in each iteration for efficiently seeking the optimum.

225 To achieve this goal, four indicators have been formulated to help enhance the search capability as follows:

The first indicator is to evaluate the quality of the metamodel and focus on the accuracy of the constraint approximations at the obtained sub-optimal point \mathbf{x}_k . This is based on the following equation:

$$E_k = \text{Max} \left(\left| \frac{\tilde{g}(\mathbf{x}_k) - g(\mathbf{x}_k)}{g(\mathbf{x}_k)} \right| \right) \quad (9)$$

230 where $\tilde{g}(\mathbf{x}_k)$ and $g(\mathbf{x}_k)$ are normalized functions of the approximate and true constraints at the sub-optimal point \mathbf{x}_k , respectively. In this way, a single maximal error quantity between the explicit approximation and the implicit simulation is defined. Then, the quality of the metamodel can be labeled as “Bad”, “Reasonable” or “Good” as shown below.

$$E_k \Rightarrow \begin{cases} \geq 0.25 \cdot S_k & \Rightarrow \text{“Bad”} \\ \leq 0.01 \cdot S_k & \Rightarrow \text{“Good”} \\ \text{Else} & \Rightarrow \text{“Reasonable”} \end{cases} \quad (10)$$

235 where S_k is a measure of the present trust region defined by

$$S_k = \text{Max} \left(\frac{B_i^k - A_i^k}{B_i - A_i} \right) \quad (11)$$

The second indicator indicates the location of the current iterate \mathbf{x}_k in the present search subregion. For each dimension, if none of the current move limits (A_i^k or B_i^k) is active, that dimension is considered as “internal”, otherwise it is denoted as “boundary”. Unless \mathbf{x}_k is actually inside the current trust region, i.e.,
240 none of the move limits ($\mathbf{A}_k, \mathbf{B}_k$) is active, this solution is regarded as “Internal”, otherwise it is viewed as “Boundary”.

The third indicator reflects the movement history for the entire optimization process. For this purpose, the angle between the last two move vectors in each dimension is calculated. The formulation of this measure θ_i^k is given below:

$$\theta_i^k = \frac{x_i^k - x_i^{k-1}}{|x_i^k - x_i^{k-1}|} \cdot \frac{x_i^{k-1} - x_i^{k-2}}{|x_i^{k-1} - x_i^{k-2}|} \quad (i = 1, \dots, d) \quad (12)$$

245 If $\theta_i^k > 0$ holds, the movement in this dimension i will be denoted as “forward”, while $\theta \leq 0$ is denoted as moving “backward”.

The fourth indicator, as a termination criterion, is triggered by the size of the current search subregion. It can be marked as “Small” or “Large” according to the quality of the metamodel determined by the first indicator. When the
250 approximations are “Bad” and $S_k \leq 2 \cdot \Delta_{min}(0.0005)$, the present search subregion is considered as “Small”. When the approximations are “Reasonable” or “Good”, the trust region is denoted as “Small” if $S_k \leq \Delta_{min}(0.0005)$.

A summary of these indicators is shown in Table 1.

Table 1: Indicators for trust region strategy

| The quality of approximation. | | | |
|-------------------------------|---|------------|----------|
| 1st indicator | Good | Reasonable | Bad |
| 2nd indicator | For each dimension, the location of x_i^k with respect to A_i^k, B_i^k . | | |
| | boundary | | internal |
| 3rd indicator | The location of \boldsymbol{x} with respect to $\boldsymbol{A}_k, \boldsymbol{B}_k$ | | |
| | Boundary | | Internal |
| 4th indicator | For each dimension, the angle between the last two move vectors. | | |
| | backward | | forward |
| 5th indicator | Termination criterion: the size of Q_k . | | |
| | Small | | Large |

Based on the indicators above, the next trust region will be a “box” centered at \boldsymbol{x}_k with an adaptive resizing of current subregion by using

$$B_i^{k+1} - A_i^{k+1} = \tau \cdot (B_i^k - A_i^k) \quad (13)$$

$$\text{Reduction: } \tau = \frac{1}{\tau_r} \quad (14)$$

$$\text{Enlargement: } \tau = \tau_e \quad (15)$$

where τ_r and τ_e are the resizing coefficients in the range of $[1.1, 1.75]$ dependent on the improvement of the objective function, which is formulated as

$$I = \frac{f(\boldsymbol{x}_{k-1}) - f(\boldsymbol{x}_k)}{f(\boldsymbol{x}_{k-1})} \cdot 100\% \quad (16)$$

The relationships between the reduction coefficient τ_r and the enlargement coefficient τ_e are defined as

$$\tau_r = \begin{cases} 1.1 & I > 1\% \\ 1.75 & I < 0.5\% \\ 2.1 - 100 \cdot I & 0.5\% \leq I \leq 1\% \end{cases} \quad (17)$$

$$\tau_e = \begin{cases} 1.75 & I > 1\% \\ 1.1 & I < 0.5\% \\ 0.6 + 100 \cdot I & 0.5\% \leq I \leq 1\% \end{cases} \quad (18)$$

The strategy applied for resizing the trust region is described in Fig. 4. Note that some processes will only be executed when the indicators have the same superscript. For example, the process can only be aborted and outputs the final optimum when the approximations are “Good” (with superscript 1) or “Bad” (with superscript 3). If the quality of the metamodel is “Reasonable” with the superscript “2”, the 4th indicator will never be triggered and a reduction of the current trust region is performed directly.

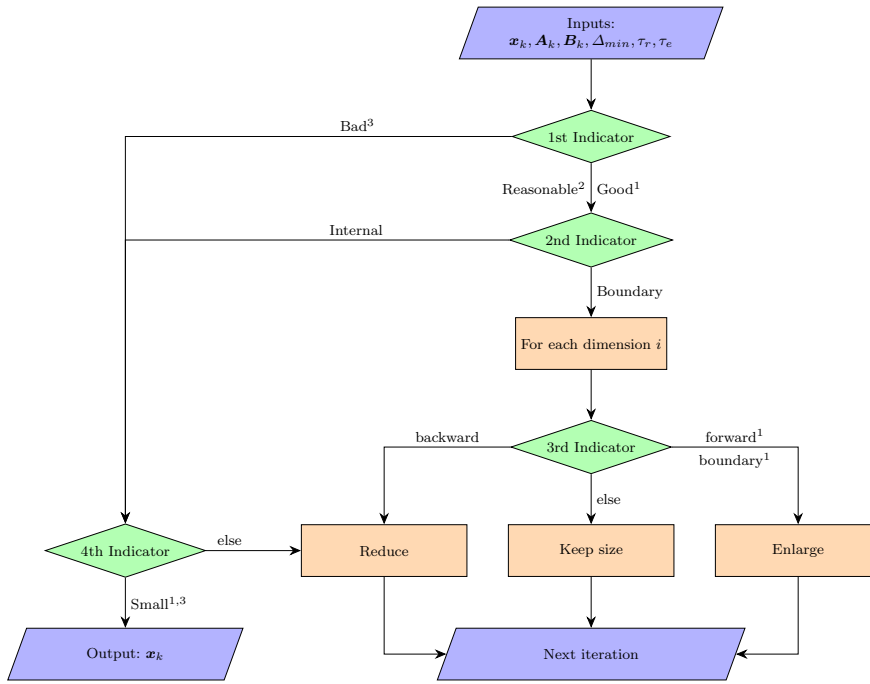


Fig. 4: Schematic description of trust region strategy

4 Benchmark examples

CBOILA was implemented in Synder 3.6.7, which is a scientific python development environment. All numerical runs were executed on a desktop machine with an AMD Ryzen CPU 1800X.

270 To solve constrained black-box optimization problems, various optimization methods and techniques have been proposed by researchers. $V(\mu + 1) - ES$ [27], co-evolutionary particle swarm optimization (CPSO) [28], and genetic algorithm based augmented lagrangian method (GAAL) [29] belong to nature-inspired algorithms, which have been developed to tackle this class of problems.

275 Meanwhile, metamodel-based optimization methods, alternative to evolutionary algorithms, have also been investigated, such as COBYLA [30], EOAS [31], SBPSO [32], COBRA [33], MSSR [34], KCGO [35], and SCGOSR [36]. COBYLA is an iterative method using linear interpolation to replace original objectives and constraints. EOAS is a hybrid evolutionary algorithm that uses the nearest neighborhood regression. SBPSO employs a hybrid surrogate, which is constructed using response surface method, to assist the particle swarm optimization for the global optimum. Meanwhile, the residual error is formulated by radial basis function. COBRA, developed by Regis, uses radial basis function interpolation technique to solve expensive black-box optimization problems.

280

285 MSSR, KCGO and SCGOSR are three kriging-based global optimization methods.

To demonstrate the robustness and efficiency of CBOILA, results have been compared with solutions from the aforementioned state-of-the-art algorithms.

4.1 Case 1: Welded Beam Design

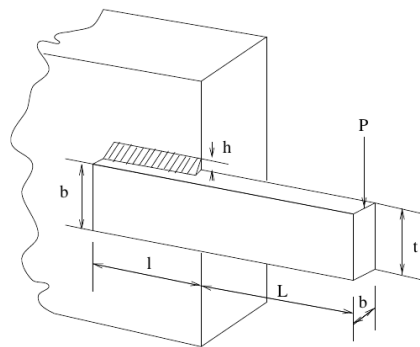


Fig. 5: Schematic of the welded beam structure

290 As shown in Fig. 5, the beam is welded to a rigid support and is designed
 for the minimum cost, considering constraints on shear stress (τ), bending stress
 (σ), buckling load (p_c), and end deflection (δ). The design variables comprise the
 thickness of the weld $h(x_1)$, the length of the welded joint $l(x_2)$, the width of the
 beam $t(x_3)$ and the thickness of the beam $b(x_4)$. The problem can be formulated
 295 mathematically as follows:

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\
\text{s.t.} \quad & g_1(\mathbf{x}) = \tau(\mathbf{x}) - \tau_{max} \leq 0 \\
& g_2(\mathbf{x}) = \sigma(\mathbf{x}) - \sigma_{max} \leq 0 \\
& g_3(\mathbf{x}) = x_1 - x_4 \leq 0 \\
& g_4(\mathbf{x}) = [0.10471x_1^2 + 0.04811x_3x_4(14 + x_2)] - 5 \leq 0 \\
& g_5(\mathbf{x}) = 0.125 - x_1 \leq 0 \\
& g_6(\mathbf{x}) = \delta(\mathbf{x}) - \delta_{max} \leq 0 \\
& g_7(\mathbf{x}) = p - p_c(\mathbf{x}) \leq 0
\end{aligned}$$

where $P = 6000 \text{ lb}$, $L = 14 \text{ in}$, $E = 30 \times 10^6 \text{ psi}$, $G = 12 \times 10^6 \text{ psi}$,
 $\tau_{max} = 13600 \text{ psi}$, $\sigma_{max} = 30000 \text{ psi}$, $\delta_{max} = 0.25 \text{ in}$

$$\begin{aligned}
\tau' &= \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P \left(L + \frac{x_2}{2} \right) \\
R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2} \right)^2} \\
\tau(x) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\
J &= 2 \left\{ \sqrt{2}x_1x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2} \right)^2 \right] \right\} \\
\sigma(x) &= \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{4PL^3}{Ex_3^3x_4} \\
p_c(x) &= \frac{4.013 \sqrt{E \left(\frac{x_3^2x_4^6}{36} \right)}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right)
\end{aligned} \tag{19}$$

$$0.1 \leq x_1 \leq 2, 0.1 \leq x_2 \leq 10, 0.1 \leq x_3 \leq 10, 0.1 \leq x_4 \leq 2$$

To demonstrate the robustness of CBOILA, 10 independent simulations have been conducted to examine its stability and the results have been compared with the published data by seven methods. As shown in Table 2 and Table 3, CBOILA has the ability to find the best result. It is noted that evolutionary algorithms without using surrogates such as CPSO and $V(\mu + 1) - ES$, can obtain a good solution but the number of function evaluations (NFEs, > 20000) is one order larger than the number of simulations by other surrogate-assisted methods. In Table 2, it is observed that both SCGOSR and CBOILA can obtain the similar solutions. From statistic point of view, the average searching quality of CBOILA is the best of all, which has been shown in Table 3. Moreover, the average NFEs required by CBOILA is 94.5 and standard deviation is $2.10e - 7$, which are smaller than results obtained by other algorithms. This confirms that CBOILA is able to solve the problem with high levels of efficiency and robustness.

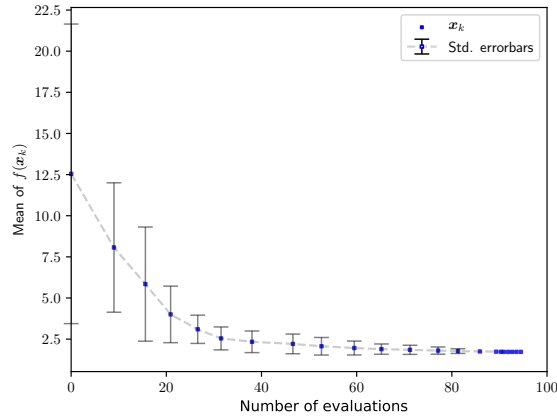
During the entire optimization progress, the average of the objective values of iterates (\mathbf{x}_k) in 10 tests have been plotted in Fig. 6. The error bars describe the standard deviations of the average. According to the plot, the objective value of each run \mathbf{x}_k varies a lot during the beginning but the standard deviation decreases sharply as the optimization proceeds. When the NFEs is over 85, the standard deviation of the objective value is smaller than 0.01, which means most runs have found the optimum and stopped. Summarily, CBOILA presents a robust search ability no matter whether the initial point is feasible or infeasible.

Table 2: Comparison of present optimized designs with literature for the welded beam.

| Methods | $x_1 (h)$ | $x_2 (l)$ | $x_3 (t)$ | $x_4 (b)$ | f |
|------------------------|-----------|-----------|-----------|-----------|----------|
| CPSO [28] | 0.202369 | 3.544214 | 9.04821 | 0.205723 | 1.728024 |
| $V(\mu + 1) - ES$ [27] | 0.199742 | 3.612060 | 9.037500 | 0.206082 | 1.737300 |
| SBPSO [32] | 0.2294 | 6.3054 | 8.8475 | 0.2294 | 2.3497 |
| COBRA-Local [33] | N.A. | N.A. | N.A. | N.A. | 1.7250 |
| MSSR [34] | 0.2056902 | 3.4683028 | 9.0445203 | 0.2056904 | 1.7256 |
| KCGO [35] | N.A. | N.A. | N.A. | N.A. | 2.3230 |
| SCGOSR [36] | 0.2057 | 3.4705 | 9.0366 | 0.2057 | 1.7249 |
| CBOILA | 0.205730 | 3.470487 | 9.036618 | 0.205729 | 1.724852 |

Table 3: Statistical results from different optimization methods for the welded beam design problem.

| Methods | Best | Mean | Worst | S.D. | NFEs |
|------------------------|----------|----------|----------|----------|--------|
| CPSO [28] | 1.728024 | 1.748831 | 1.782143 | 0.012926 | 200000 |
| $V(\mu + 1) - ES$ [27] | 1.737300 | 1.813290 | 1.994651 | 0.070500 | 25,000 |
| SBPSO [32] | 2.3497 | N.A. | N.A. | N.A. | 582 |
| COBRA-Local [33] | 1.7250 | N.A. | N.A. | N.A. | 164.57 |
| MSSR [34] | 1.7256 | N.A. | N.A. | N.A. | 156 |
| KCGO [35] | 2.3230 | N.A. | N.A. | N.A. | 115 |
| SCGOSR [36] | 1.7249 | N.A. | 1.7888 | N.A. | 101.9 |
| CBOILA | 1.724852 | 1.724852 | 1.724852 | 2.10e-07 | 94.5 |

Fig. 6: Mean of the objective values of iterates (x_k) in 10 trials obtained by CBOILA for welded beam design problem. Error bars are standard deviations about the mean.

4.2 Case 2: The tension/compression spring design

The spring design example ([37],[38]) has been widely used as a benchmark for solving constrained optimization problems. As shown in Fig. 7, the design variables include the wire diameter $d(x_1)$, the mean coil diameter $D(x_2)$, and

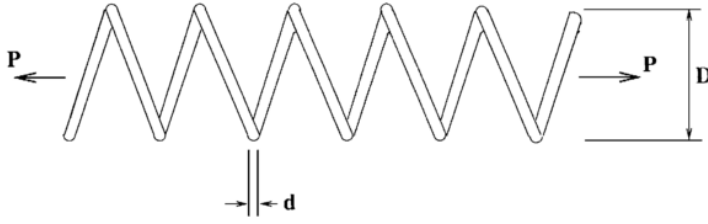


Fig. 7: Schematic of the spring structure

the number of active coils $N(x_3)$. The design objective is to minimize the weight of the spring subject to constraints on the minimum deflection g_1 , shear stress g_2 , surge frequency g_3 and the limits on the outside diameter g_4 . The mathematical description of this problem is given as follows:

$$\begin{aligned}
 \min \quad & f(\mathbf{x}) = x_1^2 x_2 (x_3 + 2) \\
 \text{s.t.} \quad & g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0 \\
 & g_2(x) = \frac{4x_2^2 - x_2 x_1}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0 \\
 & g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \\
 & g_4(x) = \frac{x_2 + x_1}{1.5} - 1 \leq 0 \\
 \text{where} \quad & 0.05 \leq x_1 \leq 1; 0.25 \leq x_2 \leq 1.3; 2 \leq x_3 \leq 15.
 \end{aligned} \tag{20}$$

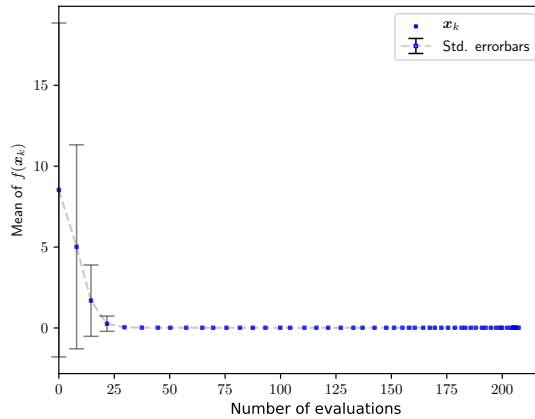
325 As illustrated in Table 4, CBOILA could find the best solution with the
smallest objective value (0.0126652). According to the statistical results obtained
from Table 5, it is evident that CBOILA has strong stability in searching the
global optimal solution because the standard deviation is just $4.83e - 8$, which is
much smaller than results obtained by other methods. Although KCGO used
330 just 38 function evaluations, it can only find a feasible point, but not the optimal
solution. SCGOSR completed the optimization process with average 75.7
function evaluations. As described in Section 3.4, CBOILA keeps on searching
for the optimum until the trust region is relatively small and the solution is just
located in this region. In Fig. 8, it can be observed that CBOILA is able to find
335 the near-optimal solution using approximately one-quarter of the total number of
function evaluations and the rest of simulations is used to perform the search for
the actual global optimum. Again, evaluation based optimization methods, such
as KCGO and SCGOSR, have to use more NFEs than the number of simulations
by metamodel based algorithms for the global optimum. It is evident that
340 CBOILA is demonstrated to be a robust and comparably efficient algorithm for
solving this benchmark problem.

Table 4: Statistical results from different optimization methods for the spring design problem.

| Methods | $x_1 (d)$ | $x_2 (D)$ | $x_3 (N)$ | f |
|------------------------|-----------|-----------|-----------|-----------|
| CPSO [28] | 0.051728 | 0.357644 | 11.244543 | 0.0126747 |
| $V(\mu + 1) - ES$ [27] | 0.355360 | 0.051643 | 11.397926 | 0.012698 |
| SBPSO [32] | 0.0517 | 0.3568 | 11.3027 | 0.012689 |
| KCGO[35] | N.A. | N.A. | N.A. | 0.0135 |
| SCGOSR [36] | 0.0516 | 0.3550 | 11.3904 | 0.0126653 |
| CBOILA | 0.0517108 | 0.357240 | 11.25837 | 0.0126652 |

Table 5: Statistical results from different optimization methods for the spring design problem.

| Methods | Best | Mean | Worst | S.D. | NFEs |
|------------------------|-----------|-----------|-----------|---------|--------|
| CPSO [28] | 0.0126747 | 0.012730 | 0.012924 | 5.20e-5 | 200000 |
| $V(\mu + 1) - ES$ [27] | 0.012698 | 0.013461 | 0.016485 | 9.66e-4 | 25,000 |
| SBPSO [32] | 0.012689 | N.A. | N.A. | N.A. | 558 |
| KCGO[35] | 0.0135 | N.A. | N.A. | N.A. | 38 |
| SCGOSR [36] | 0.01267 | N.A. | 0.01278 | N.A. | 75.7 |
| CBOILA | 0.0126652 | 0.0126653 | 0.0126654 | 4.83e-8 | 207.4 |

Fig. 8: Mean of the objective values of iterates (x_k) in 10 trials obtained by CBOILA for spring design problem. Error bars are standard deviations about the mean.

4.3 Case 3: Mathematical optimization problem—G10

This problem was one of the nonlinear constrained benchmark problems introduced in 2006 IEEE Congress on Evolutionary Computation. The

345 mathematical description is shown below:

$$\begin{aligned}
\min \quad & f(\mathbf{x}) = x_1 + x_2 + x_3 \\
\text{s.t.} \quad & g_1(x) = -1 + 0.0025(x_4 + x_6) \leq 0 \\
& g_2(x) = -1 + 0.0025(x_5 + x_7 - x_4) \leq 0 \\
& g_3(x) = -1 + 0.01(x_8 - x_5) \leq 0 \\
& g_4(x) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leq 0 \\
& g_5(x) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leq 0 \\
& g_6(x) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leq 0 \\
\text{where} \quad & 100 \leq x_1 \leq 10000 \\
& 1000 \leq x_i \leq 10000 \quad (i = 2, 3) \\
& 10 \leq x_i \leq 1000 \quad (i = 4, \dots, 8)
\end{aligned} \tag{21}$$

This has been reported as a challenging example because it belongs to a class of non-uniform scaling problems [29], i.e., highly varied ranges for different constraints. As can be observed from the above mathematical expressions, only the first three constraints are normalized. Regis [33] proposed a logarithmic transformation for the constraints with large values. However, the algorithm could not guarantee the robust optimal solution. To demonstrate the superiority of CBOILA in terms of efficiency and robustness in seeking the optimum, the obtained results have been compared with the solutions by other methods shown in Table 6. In cooperation with three developed strategies described in Section 3.4, CBOILA has the capability of finding the robust optimal solution by simply diving constraints with the factor ($10^3 \sim 10^5$). The values of eight design variables for the best solution are

$$\begin{aligned}
\mathbf{x} = & (579.311703, 1359.97245, 5109.96387, 182.018119 \\
& 295.601445, 217.981881, 286.416674, 395.601445)
\end{aligned}$$

and the optimization objective is $f = 7049.2480$.

As shown in Table 6, the best result of 7049.2480 is the global optimum found by CBOILA and the worst result obtained among 10 independent runs is 7049.2483. This remarkable performance of CBOILA owes to the very small standard deviation ($1.14e - 4$) for 10 independent runs. GAAL could also find the global optimum while the number of function evaluations is approximately 18 times more than what CBOILA requires. COBRA and KCGO can obtain a feasible solution efficiently while the obtained optimal solution is far away from the global optimum. The iterative history of results by CBOILA has been given in Fig. 9. The objective value varies significantly within the first 200 iterations, and then steadily converges to the optimum because the global optimal solution has been achieved with small standard deviations (S.D.) in the remaining optimization process. In summary, CBOILA has ability to alleviate the difficulty imposed by the larges ranges of different design constraints and find the global optimum with high levels of efficiency and robustness.

Table 6: Statistical results from different optimization methods for G10.

| Methods | Best | Mean | Worst | S.D. | NFEs |
|-------------|-----------|-----------|-----------|---------|--------|
| COBYLA [30] | 7050.3 | 8085.5 | 11259.7 | 1.90e+3 | 270840 |
| EOAS [31] | 7049.404 | 7082.227 | 7258.540 | 4.20e+1 | 304066 |
| GAAL [29] | 7049.2480 | 7049.2480 | 7049.2480 | N.A. | 10578 |
| COBRA [33] | 8238.78 | 17498.57 | 25086.88 | 892.28 | 100 |
| KCGO[35] | 7180.35 | N.A. | N.A. | N.A. | 51 |
| CBOILA | 7049.2480 | 7049.2481 | 7049.2483 | 1.14e-4 | 574.4 |

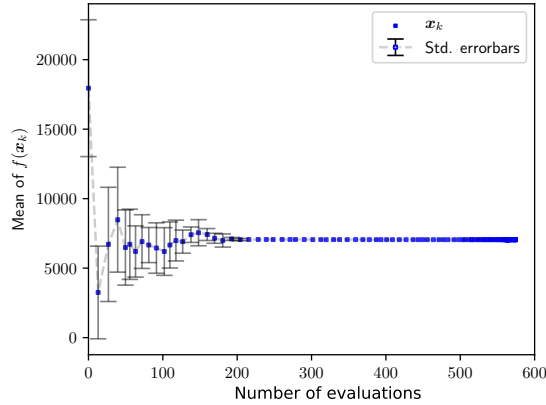


Fig. 9: Mean of the objective values of iterates (\mathbf{x}_k) in 10 trials obtained by CBOILA for G10 problem. Error bars are standard deviations about the mean.

5 Conclusions

In this work, a new framework called CBOILA has been developed for solving real-world optimization problems with expensive black-box objectives and constraints. The enhanced trust-region strategy is the corner stone of the established framework to deal with such class of optimization problems. With the integration of the developed EBS (extended-box selection) and GIS (global intelligence selection) strategies, CBOILA can identify the global optimal solution with far fewer number of function evaluations and more robust performance as compared against other state-of-the-art methods through three well-known benchmark problems. Results show that CBOILA has great potentials to efficiently and robustly solve constrained black-box optimization problems and this framework provides a useful insight to find the global optimum of complex large-scale engineering structural designs within an acceptable computational time.

References

- [1] Jichao Gu, Guangyao Li, and Nianfei Gan. “Hybrid metamodel-based design space management method for expensive problems”. In: *Engineering Optimization* (2016). DOI: 10.1080/0305215X.2016.1261126.

- [2] Chaoli Sun et al. “A two-layer surrogate-assisted particle swarm optimization algorithm”. In: *Soft Computing* 19.6 (2015), pp. 1461–1475. DOI: 10.1007/s00500-014-1283-z.
- [3] John H. Holland. *Adaptation in Natural and Artificial Systems: An introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1975, p. 183. DOI: 10.1137/1018105.
- [4] Qi Zhou et al. “A two-stage adaptive multi-fidelity surrogate model-assisted multi-objective genetic algorithm for computationally expensive problems”. In: *Engineering with Computers* August 2019 (2019). DOI: 10.1007/s00366-019-00844-8.
- [5] David H. Wolpert and William G. Macready. “No-Free-Lunch Theorem”. In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82. DOI: 10.1007/978-0-387-30164-8_592.
- [6] R H Myers et al. “Response Surface Methodology: A Retrospective and Literature Survey”. In: *Journal of Quality Technology* 36.1 (2004), pp. 53–78. DOI: 10.1080/00224065.2004.11980252.
- [7] Yoel Tenne and S. W. Armfield. “A framework for memetic optimization using variable global and local surrogate models”. In: *Soft Computing* 13.8-9 (2009), pp. 781–793. DOI: 10.1007/s00500-008-0348-2.
- [8] Jiachang Qian et al. “A sequential constraints updating approach for Kriging surrogate model-assisted engineering optimization design problem”. In: *Engineering with Computers* 36.3 (July 2020), pp. 993–1009. DOI: 10.1007/s00366-019-00745-w.
- [9] Stella M. Clarke, Jan H. Griebisch, and Timothy W. Simpson. “Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses”. In: *Transactions of ASME, Journal of Mechanical Design* 127.6 (2005), pp. 1077–1087. DOI: 10.1115/1.1897403.
- [10] Timothy W. Simpson et al. “Metamodels for Computer-Based Engineering Design: Survey and Recommendations”. In: *Engineering with Computers* 17.2 (2001), pp. 129–150. DOI: 10.1017/CB09781107415324.004.
- [11] Yew Soon Ong, P.B. Nair, and K.Y. Lum. “Max-min surrogate-assisted evolutionary algorithm for robust design”. In: *IEEE Transactions on Evolutionary Computation* 10.4 (2006), pp. 392–404. DOI: 10.1109/TEVC.2005.859464.
- [12] Zongzhao Zhou et al. “Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization”. In: *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37.1 (Jan. 2007), pp. 66–76. DOI: 10.1109/TSMCC.2005.855506.
- [13] Rommel G. Regis. “Particle swarm with radial basis function surrogates for expensive black-box optimization”. In: *Journal of Computational Science* 5.1 (Jan. 2014), pp. 12–23. DOI: 10.1016/j.jocs.2013.07.004.
- [14] Efrén Mezura-montes and Carlos A. Coello Coello. “Constraint-handling in nature-inspired numerical optimization : Past , present and future”. In: *Swarm and Evolutionary Computation* 1.4 (2011), pp. 173–194. DOI: 10.1016/j.swevo.2011.10.001.
- [15] Marcella C Araujo et al. “Constrained Optimization Based on Quadratic Approximations in Genetic Algorithms”. In: *Constraint-Handling in Evolutionary Optimization*. Ed. by Efrén Mezura-Montes. Berlin,

- Heidelberg: Springer Berlin Heidelberg, 2009, pp. 193–217. DOI: 10.1007/978-3-642-00619-7_9.
- 430 [16] Rommel G. Regis. “Trust Regions in Surrogate-Assisted Evolutionary Programming for Constrained Expensive Black-Box Optimization”. In: *Evolutionary Constrained Optimization*. Ed. by Rituparna Datta and Kalyanmoy Deb. New Delhi: Springer India, 2015, pp. 51–94. DOI: 10.1007/978-81-322-2184-5.
- 435 [17] Ruud Brekelmans et al. “Constrained optimization involving expensive function evaluations: A sequential approach”. In: *European Journal of Operational Research* 160.1 (2005), pp. 121–138. DOI: 10.1016/j.ejor.2003.10.009.
- [18] Anirban Basudhar et al. “Constrained efficient global optimization with support vector machines”. In: *Structural and Multidisciplinary Optimization* 46.2 (2012), pp. 201–221. DOI: 10.1007/s00158-011-0745-5.
- 440 [19] Patrick Koch et al. “A New Repair Method For Constrained Optimization”. In: *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference - GECCO '15*. GECCO '15 September. New York, New York, USA: ACM Press, 2015, pp. 273–280. DOI: 445 10.1145/2739480.2754658.
- [20] Samineh Bagheri et al. “Self-adjusting parameter control for surrogate-assisted constrained optimization under limited budgets”. In: *Applied Soft Computing Journal* 61 (2017), pp. 377–393. DOI: 450 10.1016/j.asoc.2017.07.060.
- [21] NR Draper and H Smith. *Applied regression analysis*. 2nd. New York: John Wiley and Sons, 1981.
- [22] Vassili V. Toropov. “Simulation approach to structural optimization”. In: *Structural Optimization* 1.1 (Mar. 1989), pp. 37–46. DOI: 10.1007/BF01743808.
- 455 [23] Andrey Polynkin and Vassili V. Toropov. “Mid-range metamodel assembly building based on linear regression for large scale optimization problems”. In: *Structural and Multidisciplinary Optimization* 45.4 (Apr. 2012), pp. 515–527. DOI: 10.1007/s00158-011-0692-1.
- [24] Dianzi Liu and Vassili V. Toropov. “Implementation of Discrete Capability into the Enhanced Multipoint Approximation Method for Solving Mixed Integer-Continuous Optimization Problems”. In: *International Journal of Computational Methods in Engineering Science and Mechanics* 17.1 (2016). DOI: 460 10.1080/15502287.2016.1139013.
- [25] Stefano Caloni, Shahrokh Shahpar, and Vassili V. Toropov. “Multi-Disciplinary Design Optimisation of the Cooled Squealer Tip for High Pressure Turbines”. In: *Aerospace* 5.116 (2018). DOI: 465 10.3390/aerospace5040116.
- [26] George E P Box and Norman R Draper. *Empirical Model-Building and Response Surfaces*. 1987, p. 669. DOI: 10.1037/028110.
- 470 [27] Efrén Mezura-Montes and Carlos A. Coello Coello. “An empirical study about the usefulness of evolution strategies to solve constrained optimization problems”. In: *International Journal of General Systems* 37. April 2013 (2008), pp. 443–473. DOI: 10.1080/03081070701303470.
- [28] Qie He and Ling Wang. “An effective co-evolutionary particle swarm 475 optimization for constrained engineering design problems”. In: *Engineering*

Applications of Artificial Intelligence 20.1 (2007), pp. 89–99. DOI: 10.1016/j.engappai.2006.03.003.

- [29] Kalyanmoy Deb and Soumil Srivastava. “A genetic algorithm based augmented Lagrangian method for constrained optimization”. In: *Computational Optimization and Applications* 53.3 (Dec. 2012), pp. 869–902. DOI: 10.1007/s10589-012-9468-9.
- [30] M. J.D. Powell. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*. Ed. by Susana Gomez and JP. Hennart. Dordrecht: Springer, Dordrecht, 1994, pp. 51–67. DOI: 10.1007/978-94-015-8330-5_4.
- [31] Thomas Philip Runarsson. “Constrained Evolutionary Optimization by Approximate Ranking and Surrogate Models”. In: *Parallel Problem Solving from Nature - PPSN VIII*. Ed. by Xin Yao et al. Berlin, Heidelberg: Springer, Berlin, Heidelberg, 2004, pp. 401–410. DOI: 10.1007/978-3-540-30217-9_41.
- [32] Yuanfu Tang, Jianqiao Chen, and Junhong Wei. “A surrogate-based particle swarm optimization algorithm for solving optimization problems with expensive black box functions”. In: *Engineering Optimization* 45.5 (May 2013), pp. 557–576. DOI: 10.1080/0305215X.2012.690759.
- [33] Rommel G. Regis. “Constrained Optimization by Radial Basis Function Interpolation for High-Dimensional Expensive Black-Box Problems with Infeasible Initial Points”. In: *Engineering Optimization* 46.2 (2014), pp. 218–243. DOI: 10.1080/0305215X.2013.765000.
- [34] Huachao Dong et al. “Multi-start Space Reduction (MSSR) surrogate-based global optimization method”. In: *Structural and Multidisciplinary Optimization* 54.4 (2016), pp. 907–926. DOI: 10.1007/s00158-016-1450-1.
- [35] Yaohui Li et al. “A Kriging-based constrained global optimization algorithm for expensive black-box functions with infeasible initial points”. In: *Journal of Global Optimization* 67.1-2 (2017), pp. 343–366. DOI: 10.1007/s10898-016-0455-z.
- [36] Huachao Dong et al. “SCGOSR : Surrogate-based constrained global optimization using space reduction”. In: *Applied Soft Computing Journal* 65 (2018), pp. 462–477. DOI: 10.1016/j.asoc.2018.01.041.
- [37] Ashok D. Belegundu. “A study of mathematical programming methods for structural optimization”. Ph.D. University of Iowa, 1982.
- [38] Jasbir S. Arora. *Introduction to Optimum Design*. 2nd. New York: Elsevier, 2004. DOI: 10.1016/B978-0-12-064155-0.X5000-9.