

Article

A Novel Gaussian Process Surrogate Model with Expected Prediction Error for Optimization under Constraints

Hongri Cong ¹, Bo Wang ^{1,*} and Zhe Wang ²

¹ School of Computing and Mathematical Sciences, University of Leicester, Leicester LE1 7RH, UK; hc304@leicester.ac.uk

² Covestro (Netherlands), 6167 RD Geleen, The Netherlands; zhewang2015@outlook.com

* Correspondence: bo.wang@le.ac.uk; Tel.: +44-116-252-2162

Abstract: Optimization, particularly constrained optimization problems (COPs), is fundamental in engineering, influencing various sectors with its critical role in enhancing design efficiency, reducing experimental costs, and shortening testing cycles. This study explores the challenges inherent in COPs, with a focus on developing efficient solution methodologies under stringent constraints. Surrogate models, especially Gaussian Process Regression (GPR), are pivotal in our approach, enabling the approximation of complex systems with reduced computational demand. We evaluate the efficacy of the Efficient Global Optimization (EGO) algorithm, which synergizes GPR with the Expected Improvement (EI) function, and further extend this framework to Constrained Expected Improvement (CEI) and our novel methodology Constrained Expected Prediction Error (CEPE). We demonstrate the effectiveness of these methodologies by numerical benchmark simulations and the real-world application of optimizing a Three-Bar Truss Design. In essence, the innovative CEPE approach promises a potent balance between solution accuracy and computational prowess, offering significant potential in the broader engineering field.

Keywords: constrained optimization problems; Gaussian process regression; constrained expected improvement; constrained expected prediction error; Three-Bar Truss Design

MSC: 62P30



Citation: Cong, H.; Wang, B.; Wang, Z.

A Novel Gaussian Process Surrogate Model with Expected Prediction Error for Optimization under Constraints.

Mathematics **2024**, *12*, 1115. <https://doi.org/10.3390/math12071115>

Academic Editors: Simeon Reich and Ioannis G. Tsoulos

Received: 6 February 2024

Revised: 1 April 2024

Accepted: 6 April 2024

Published: 8 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the fields of science and engineering, optimization problems, especially those with constraints, play a crucial role. Constrained optimization problems (COPs) are vital in a wide range of applications, including process control [1,2], reactor design [3], and production scheduling [4]. Among these, specific challenges such as tension/compression coil spring design [5] highlight the intricate nature of these problems. The aim of addressing COPs is to enhance design efficiency, minimize experimental costs, and reduce testing cycles. These problems hold intrinsic value across various engineering disciplines, including chemical, mechanical, and electrical engineering [6–8], as they seek optimal solutions within the boundaries set by constraints, be they equalities or inequalities [9]. To solve these complex issues, a range of methodologies have been developed, from gradient-based methods [10], known for their efficiency in differentiable problems, to population-based strategies like Genetic Algorithms [11], which are capable of addressing non-differentiable and multi-modal challenges, albeit often at a higher computational cost [12]. COPs are typically represented through mathematical formulations, underscoring their quantitative nature and the systematic approach required for their resolution. We define the formulation

$$\begin{aligned}
\min \quad & y = f(\mathbf{x}), \\
\text{st :} \quad & \mathbf{l} \leq \mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_m(\mathbf{x})) \leq \mathbf{u}, \\
\text{where} \quad & \mathbf{l} = (l_1, l_2, \dots, l_m), \mathbf{u} = (u_1, u_2, \dots, u_m), \\
& \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbf{X}, \\
& \mathbf{X} = \{\mathbf{x} | \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u\}, \\
& \mathbf{x}_l = (x_{l_1}, x_{l_2}, \dots, x_{l_n}), \mathbf{x}_u = (x_{u_1}, x_{u_2}, \dots, x_{u_n}),
\end{aligned} \tag{1}$$

where $\mathbf{x}$ is a solution vector within the solution space $\mathbf{X}$, $\mathbf{g}(\mathbf{x})$ represents the constraints, and $\mathbf{l}$ and $\mathbf{u}$ are the lower and upper bounds of the constraints, respectively. There are two types of scenarios for the solution to this formulation: feasible and infeasible. A solution is considered feasible if the solution vector $\mathbf{x} \in \mathbf{X}$ satisfies all the constraints $\mathbf{g}(\mathbf{x})$. Conversely, a solution is deemed infeasible if $\mathbf{x} \in \mathbf{X}$ but does not satisfy all the constraints $\mathbf{g}(\mathbf{x})$. At times, the feasibility ratio (the percentage of feasible solutions out of the total number of feasible and infeasible solutions) can be very low, which is equivalent to having very stringent constraints. In such cases, it is imperative to find a solution that satisfies the constraints as efficiently as possible.

For solving complex optimization problems, especially those with constraints, the use of surrogate models has emerged as a highly effective strategy. Surrogate models, pivotal in the realm of optimization, provide simplified yet powerful approximations of complex systems, enabling efficient exploration and optimization of design spaces where direct evaluations are prohibitively expensive. These models, also referred to as metamodels, span a diverse array of methodologies, each tailored to capture the intricacies of different problem domains [13]. To encapsulate these varying surrogate model methodologies and their application in constrained optimization problems, Table 1 provides a comprehensive summary of the key literature, highlighting techniques, tested datasets, and their respective advantages and disadvantages.

Table 1. Summary of surrogate model methodologies in constrained optimization problems.

Methodology	Reference	Approach	Advantages	Limitations
Polynomial Regression	Box and Wilson, 1992 [14]	Utilizes polynomial equations to model relationships between variables	Straightforward approach for problems with smooth relationships	May not capture complex, nonlinear behaviour effectively
Radial Basis Function (RBF) Networks	Buhmann, 2000 [15]	Employs radial basis functions to approximate unknown functions	Effective for scattered multidimensional data	Computationally intensive for large datasets
Kriging	Oliver and Webster, 1990 [16]	A best-unbiased predictor based on geostatistical methods	Provides an estimate of prediction error; well-suited for surrogate modelling	Requires significant computational resources for large problems
Gaussian Process Regression (GPR)	Gramacy, 2020 [17]	Models unknown functions as samples from a Gaussian process	Robust in capturing trends and uncertainty quantification	May be challenging to scale to very high-dimensional problems

Despite these advancements, a comprehensive review of the literature underscores certain limitations in the current surrogate modelling approaches, particularly when applied to constrained optimization problems [18]. A notable challenge is the difficulty in accurately capturing and adapting to the complex constraint boundaries that often define feasible regions in optimization landscapes [19]. Many existing surrogate models excel in unconstrained scenarios or situations with simple constraints but may fall short when

faced with complex, nonlinear, or high-dimensional constraints [20]. Additionally, there is a paucity of methodologies that effectively incorporate constraint handling mechanisms within the surrogate model itself, often relying on external penalty functions or constraint relaxation techniques that may not always yield optimal solutions. This gap in the literature highlights the need for more sophisticated surrogate models that can inherently deal with complex constraints while maintaining the balance between exploration and exploitation in the search space. These insights into the limitations of current surrogate models in handling constrained optimization problems have motivated the development of our proposed methodologies. By addressing these identified gaps, we aim to contribute to the advancement of surrogate-assisted optimization strategies, providing more robust and efficient solutions for complex constrained optimization challenges.

Among the diverse array of surrogate modelling techniques, Gaussian Process Regression (GPR) stands out for its efficiency and effectiveness [17]. GPR is renowned for its robustness in capturing the underlying trends of the data with a quantifiable measure of uncertainty, making it particularly suitable for optimization problems where uncertainty plays a critical role [21]. The historical development of surrogate models in optimization was notably advanced in 1998 when Jones et al. [22] introduced the Efficient Global Optimization (EGO) algorithm. This algorithm integrates GPR with the Expected Improvement (EI) function, a pivotal concept in Bayesian optimization. The EI function is designed to systematically identify the global optimum of computationally expensive black-box functions, which are often subject to inherent uncertainties [23]. This approach prioritizes sampling in regions where the anticipated improvement in the function's value is maximized, thereby enhancing the model's accuracy in critical areas. This methodology exemplifies an active learning strategy, dynamically refining the model with each new data point to efficiently navigate the complex landscape of the optimization problem. However, when dealing with optimization problems that include stringent constraints, the standard EI approach may require modifications to accommodate these limitations. In response to this challenge, recent developments have introduced variants such as the Constrained Expected Improvement (CEI), which adapt the EI principle to handle constraints effectively [24,25].

In addressing the challenges inherent in constrained optimization, this study introduces significant innovations that extend the current state-of-the-art methodologies. Firstly, we propose a novel approach to compute the Expected Prediction Error (EPE) [26] at an untested point by leveraging the cross-validation error from a nearby tested point. This method enhances the prediction accuracy of our surrogate model, particularly in the sparse regions of the design space where traditional interpolation methods might falter. Secondly, we introduce the Constrained Expected Prediction Error (CEPE) criterion, a pioneering metric designed to navigate the intricate landscape of constrained optimization problems efficiently. By integrating CEPE within a Differential Evolution (DE) [27] framework augmented with Gaussian Process (GP) surrogates, our approach not only capitalizes on the strengths of evolutionary algorithms in exploring complex solution spaces but also harnesses the predictive prowess of GP models to make informed decisions during the optimization process. The synergy between these innovations presents a robust framework that promises improved optimization performance, especially in scenarios characterized by stringent constraints and expensive function evaluations.

The remainder of this paper is structured as follows. In Section 2, we introduce the theoretical knowledge about Gaussian Process, Expected Improvement and Expected Prediction Error. In Section 3, we discuss methodologies for Constrained Expected Improvement, introduce our novel method Constrained Expected Prediction Error, and provide the GP surrogate-assisted Differential Evolution (DE) algorithm. In Section 4, we evaluate the efficacy of CEI and CEPE using benchmark problems, and in Section 5, we illustrate their application in a real-world problem. Finally, we conclude this paper with discussions and remarks in Section 6.

2. Background and Concepts

In this section, we provide a brief theoretical overview of Gaussian Process Regression (GPR), Expected Improvement (EI), and Expected Prediction Error (EPE).

2.1. Gaussian Process Regression

We aim to use the Gaussian Process Regression (GPR) framework to model an unknown function $f(\mathbf{x})$, by assuming that $f(\mathbf{x})$ is a sample from a Gaussian process denoted as $y(\mathbf{x})$. Within this context, for any input point $\mathbf{x}$, the function value $f(\mathbf{x})$ is viewed as a sample from a Gaussian random variable $y(\mathbf{x})$ and $y(\mathbf{x})$ is distributed as $\mathcal{N}(\mu, \sigma^2)$, where μ and σ^2 are constants independent of $\mathbf{x}$. The covariance of $y(\mathbf{x})$ with another random variable $y(\mathbf{x}')$, where $\mathbf{x}'$ represents another point in the input space, is given by

$$\text{cov}[y(\mathbf{x}), y(\mathbf{x}')] = \sigma^2 \text{Corr}(y(\mathbf{x}), y(\mathbf{x}')). \quad (2)$$

The correlation function $\text{Corr}(y(\mathbf{x}), y(\mathbf{x}'))$ is defined as

$$\text{Corr}(y(\mathbf{x}), y(\mathbf{x}')) = \exp\left(-\sum_{k=1}^p \theta_k |x_k - x'_k|^2\right), \quad (3)$$

where p is the dimension of the input space and θ_k are the hyperparameters.

Considering a set of N input points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$, let $\mathbf{f}^N = (f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_N))^T$ and $\mathbf{y}^N = (y(\mathbf{x}_1), y(\mathbf{x}_2), \dots, y(\mathbf{x}_N))^T$. The parameters $\mu, \sigma^2, \theta_1, \dots, \theta_p$ can be estimated by maximizing the joint Gaussian probability density function of $\mathbf{y}^N$ at $\mathbf{f}^N$

$$p(\mathbf{f}^N | \mu, \sigma^2, \mathbf{R}) = \frac{1}{(2\pi\sigma^2)^{N/2} \sqrt{\det(\mathbf{R})}} \exp\left[-\frac{(\mathbf{f}^N - \mu\mathbf{1})^T \mathbf{R}^{-1} (\mathbf{f}^N - \mu\mathbf{1})}{2\sigma^2}\right], \quad (4)$$

where $\mathbf{R}$ is an $N \times N$ matrix with elements $R_{i,j} = \text{Corr}(y(\mathbf{x}_i), y(\mathbf{x}_j))$, and $\mathbf{1}$ is an N -dimensional column vector of ones.

The log-likelihood function is then

$$\mathcal{L}(\mu, \sigma^2, \theta_1, \dots, \theta_p | \mathbf{f}^N) = -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2} \log(\det(\mathbf{R})) - \frac{(\mathbf{f}^N - \mu\mathbf{1})^T \mathbf{R}^{-1} (\mathbf{f}^N - \mu\mathbf{1})}{2\sigma^2}. \quad (5)$$

Now, let us differentiate the log-likelihood with respect to μ :

$$\frac{\partial \mathcal{L}}{\partial \mu} = \frac{1}{\sigma^2} (\mathbf{1}^T \mathbf{R}^{-1} (\mathbf{f}^N - \mu\mathbf{1})). \quad (6)$$

Setting this derivative to zero and solving for μ gives the maximum likelihood estimate

$$\hat{\mu} = \frac{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{f}^N}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}}. \quad (7)$$

Next, let us differentiate the log-likelihood with respect to σ^2

$$\frac{\partial \mathcal{L}}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{(\mathbf{f}^N - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1} (\mathbf{f}^N - \mathbf{1}\hat{\mu})}{2\sigma^4}. \quad (8)$$

Setting this derivative to zero and solving for σ^2 gives the maximum likelihood estimate

$$\hat{\sigma}^2 = \frac{(\mathbf{f}^N - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1} (\mathbf{f}^N - \mathbf{1}\hat{\mu})}{N}. \quad (9)$$

There is no analytical form for the estimates of $\theta_1, \dots, \theta_p$, so the maximization of the log-likelihood function with respect to $\theta_1, \dots, \theta_p$ is usually completed by conjugate descent.

With the maximum likelihood estimates of the parameters, we can now derive expressions for predicting $f(\mathbf{x})$ at a new point $\mathbf{x}$. Let $\mathbf{r}$ be the vector of correlations between the

new point and the N data points, where the i -th element of $\mathbf{r}$ is $\text{Corr}(y(\mathbf{x}), y(\mathbf{x}_i))$. The best linear unbiased predictor (BLUP) of $f(\mathbf{x})$ can be written as:

$$\hat{f}(\mathbf{x}) = \hat{\mu} + \mathbf{r}^T \mathbf{R}^{-1}(\mathbf{f}^N - \mathbf{1}\hat{\mu}), \quad (10)$$

where $\hat{\mu}$ is the estimate of the mean in (7) and $\mathbf{R}$ is the correlation matrix calculated using the estimates of θ_i .

Similarly, the prediction variance can be derived as:

$$s^2(\mathbf{x}) = \hat{\sigma}^2 \left(1 - \mathbf{r}^T \mathbf{R}^{-1} \mathbf{r} + \frac{(1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r})^2}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right). \quad (11)$$

This forms the basis for Gaussian Process Regression, where we model the unknown function $f(\mathbf{x})$ as a Gaussian process and make predictions at new points by combining information from the observed data.

2.2. Expected Improvement

In this section, we introduce a widely-adopted infill sampling method known as Expected Improvement (EI), developed by Jones et al. [22] for the optimization of expensive black-box functions. Expected Improvement is particularly useful once a surrogate model, typically Gaussian Process Regression, is constructed.

As previously defined in Equation (1), let $f(\mathbf{x})$ denote the objective function with a surrogate model $y(\mathbf{x})$ that follows a Gaussian process. Consider a set of test inputs $\{x_1, x_2, \dots, x_N\}$ with corresponding observed function values $\mathbf{f}^N = (f(x_1), f(x_2), \dots, f(x_N))^T$. We define $f_{\min}$ as the best observed value of the function among the test inputs, i.e., $f_{\min} = \min(\mathbf{f}^N)$. The improvement of $f(\mathbf{x})$ at a new input point $\mathbf{x}$ is defined as

$$I(\mathbf{x}) = \max\{f_{\min} - f(\mathbf{x}), 0\}. \quad (12)$$

Expected Improvement is then defined as the expected value of this improvement, given the observed data $\mathbf{f}^N$

$$E[I(\mathbf{x})|\mathbf{f}^N] = E[\max\{f_{\min} - f(\mathbf{x}), 0\}|\mathbf{f}^N]. \quad (13)$$

Through some non-trivial mathematical manipulations, we can express the Expected Improvement in terms of the predictive mean and standard deviation of the surrogate model as follows:

$$E[I(\mathbf{x})|\mathbf{f}^N] = (f_{\min} - \hat{f}(\mathbf{x}))\Phi\left(\frac{f_{\min} - \hat{f}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x})\phi\left(\frac{f_{\min} - \hat{f}(\mathbf{x})}{s(\mathbf{x})}\right), \quad (14)$$

where $\hat{f}(\mathbf{x})$ and $s(\mathbf{x})$ are the predictive mean and standard deviation of the surrogate model at $\mathbf{x}$, computed according to Equations (10) and (11), and $\phi(\cdot)$ and $\Phi(\cdot)$ denote the probability density function (PDF) and cumulative distribution function (CDF) of the standard normal distribution, respectively.

2.3. Expected Prediction Error

In this section, we introduce the Expected Prediction Error (EPE), an active learning strategy employed to evaluate the accuracy of a predictive model. EPE quantifies the discrepancy between the predicted and actual values. Specifically, it is the expected value of the squared prediction error over a sample of data. EPE is sometimes referred to as the mean squared error (MSE).

Within the context of the GPR methodology, the posterior distribution of the response at a given location $\mathbf{x}$ is denoted as $\hat{y}(\mathbf{x})$, which follows a Gaussian distribution expressed as $\hat{y}(\mathbf{x}) \sim \mathcal{N}(\mu_{\mathbf{x}}, \sigma_{\mathbf{x}}^2)$. In this setting, the mean $\mu_{\mathbf{x}}$ symbolizes the predictive estimate given by $\hat{f}(\mathbf{x})$, while the variance $\sigma_{\mathbf{x}}^2$ reflects the uncertainty associated with the prediction, denoted

as s_x^2 . Given this framework, we can define the prediction error, often referred to as the loss function, $\mathcal{L}$, for a point $\mathbf{x}$ as

$$\mathcal{L}(\mathbf{x}) = (y(\mathbf{x}) - \hat{y}(\mathbf{x}))^2. \quad (15)$$

Here, $y(\mathbf{x})$ embodies the true underlying response, perturbed by inherent observational noise.

Then, the overall generalization error of the GPR-based surrogate model is given by

$$e = \int_{\mathcal{D}} E[\mathcal{L}(\mathbf{x})] d\mathbf{x}, \quad (16)$$

where $E[\mathcal{L}(\mathbf{x})]$ denotes the expected value of the prediction error, and $\mathcal{D}$ is a subset of $\mathbb{R}^p$. This expectation can be decomposed into

$$E[\mathcal{L}(\mathbf{x})] = (E[y(\mathbf{x})] - E[\hat{y}(\mathbf{x})])^2 + E[(\hat{y}(\mathbf{x}) - E[\hat{y}(\mathbf{x})])^2] + E[(y(\mathbf{x}) - E[y(\mathbf{x})])^2], \quad (17)$$

where the first term represents the squared bias, capturing the average difference between the predicted and observed responses; the second term is the prediction variance of the surrogate model; and the third term is the variance of the noise $Var(\epsilon)$, which is often negligible.

The EPE quantifies the discrepancy between the GPR model's predictions and the true function values. In this context, $f(\mathbf{x})$ denotes the true function we aim to approximate, and $\hat{f}(\mathbf{x})$ represents the GPR model's prediction at $\mathbf{x}$. Thus, EPE at a point $\mathbf{x}$ is given by

$$EPE(\mathbf{x}) = \underbrace{(f(\mathbf{x}) - \hat{f}(\mathbf{x}))^2}_{\text{bias}^2} + \underbrace{s^2(\mathbf{x})}_{\text{variance}}. \quad (18)$$

However, the true response $f(\mathbf{x})$ is unknown in the bias term. Following the approach by Liu et al. [26], we employ leave-one-out cross-validation for estimation. Initially, we estimate the cross-validation errors at all training sample locations:

$$e_{CV}^2(x_i) = (f(\mathbf{x}_i) - \hat{f}^{-i}(\mathbf{x}_i))^2, \quad i = 1, 2, \dots, N, \quad (19)$$

where $\mathbf{x}_i$ represents the i -th training sample, and $\hat{f}^{-i}$ denotes the GP model trained using all training samples except $(\mathbf{x}_i, f(\mathbf{x}_i))$. Next, for any point $\mathbf{x}$, we locate the nearest training sample to $\mathbf{x}$, denoted as $\mathbf{x}_i$, and assign its associated cross-validation error to $e_{CV}^2(\mathbf{x})$:

$$e_{CV}^2(\mathbf{x}) = e_{CV}^2(\mathbf{x}_i), \quad \text{where } i = \arg \min_i |\mathbf{x} - \mathbf{x}_i|, \quad i = 1, 2, \dots, N. \quad (20)$$

Finally, incorporating the cross-validation error, we obtain a refined expression for the EPE:

$$EPE(\mathbf{x}) = e_{CV}^2(\mathbf{x}) + s^2(\mathbf{x}). \quad (21)$$

The Expected Prediction Error (EPE) plays a crucial role in assessing the accuracy and reliability of our surrogate model, particularly in constrained optimization problems. By calculating the EPE, we can identify regions in the design space where the model's predictions are less reliable, informing us where additional sampling might be beneficial. Moreover, as EPE comprises both bias and variance, it aids in striking a balance between the two, ensuring that the model is neither too simplistic nor too complex. This information is vital for efficient exploration of the search space and making informed decisions in the optimization process.

3. Methodology

In this section, we first present the CEI method proposed by Jiao et al. [25], then we introduce a novel method CEPE for solving constrained optimization problems (COP). To solve a COP, the objective is to minimize the objective function subject to several constraints.

Before delving into the details of our proposed method, we will define some notations in the following subsection.

In this work, we treat $\mathbf{x}$ as an untested point, while $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ are considered tested points. Let $y(\mathbf{x})$ denote the objective function, and $G_i(\mathbf{x})$ represent the constraint functions for $i = 1, 2, \dots, m$. For a given point $\mathbf{x}$, we define a Gaussian random vector $\mathbf{G}(\mathbf{x})$ as

$$\mathbf{G}(\mathbf{x}) = (G_1(\mathbf{x}), G_2(\mathbf{x}), \dots, G_m(\mathbf{x})). \quad (22)$$

We introduce $\mathbf{G}(\mathbf{x})$ as a Gaussian random vector to represent the constraints' evaluations at the point $\mathbf{x}$. This probabilistic modelling approach allows us to account for and manage the uncertainties associated with constraint evaluations, especially when exact determinations of these constraints are not feasible or when they are subject to variability due to measurement errors, model inaccuracies, or other sources of uncertainty.

For a set of N tested points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$, we define the following Gaussian random vectors:

$$\begin{aligned} \mathbf{y}^N &= (y(\mathbf{x}_1), y(\mathbf{x}_2), \dots, y(\mathbf{x}_N))^T, \\ \mathbf{G}_1^N &= (G_1(\mathbf{x}_1), G_1(\mathbf{x}_2), \dots, G_1(\mathbf{x}_N))^T, \\ \mathbf{G}_2^N &= (G_2(\mathbf{x}_1), G_2(\mathbf{x}_2), \dots, G_2(\mathbf{x}_N))^T, \\ &\vdots \\ \mathbf{G}_m^N &= (G_m(\mathbf{x}_1), G_m(\mathbf{x}_2), \dots, G_m(\mathbf{x}_N))^T, \\ \mathbf{G}^{mN} &= (G_1(\mathbf{x}_1), G_2(\mathbf{x}_1), \dots, G_m(\mathbf{x}_1), \dots, G_1(\mathbf{x}_N), G_2(\mathbf{x}_N), \dots, G_m(\mathbf{x}_N))^T, \end{aligned}$$

where $\mathbf{y}^N, \mathbf{G}_1^N, \mathbf{G}_2^N, \dots, \mathbf{G}_m^N$ and $\mathbf{G}^{mN}$ represent the function values at the tested points. In contrast, $y(\mathbf{x})$ and $\mathbf{G}(\mathbf{x})$ are the quantities to be predicted. The evaluated function values are

$$\begin{aligned} \mathbf{f}^N &= (f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_N))^T, \\ \mathbf{g}_1^N &= (g_1(\mathbf{x}_1), g_1(\mathbf{x}_2), \dots, g_1(\mathbf{x}_N))^T, \\ \mathbf{g}_2^N &= (g_2(\mathbf{x}_1), g_2(\mathbf{x}_2), \dots, g_2(\mathbf{x}_N))^T, \\ &\vdots \\ \mathbf{g}_m^N &= (g_m(\mathbf{x}_1), g_m(\mathbf{x}_2), \dots, g_m(\mathbf{x}_N))^T, \\ \mathbf{g}^{mN} &= (g_1(\mathbf{x}_1), g_2(\mathbf{x}_1), \dots, g_m(\mathbf{x}_1), \dots, g_1(\mathbf{x}_N), g_2(\mathbf{x}_N), \dots, g_m(\mathbf{x}_N))^T, \end{aligned}$$

which are considered as samples of $\mathbf{y}^N, \mathbf{G}_1^N, \mathbf{G}_2^N, \dots, \mathbf{G}_m^N$ and $\mathbf{G}^{mN}$. Essentially, by using the known function values, we can predict $y(\mathbf{x})$ and $G_i(\mathbf{x})$ for $i = 1, 2, \dots, m$ at the tested points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ without additional function evaluations.

3.1. Constrained Expected Improvement

In the exploration of constrained optimization problems, we often encounter a combination of both feasible and infeasible solutions, as categorized by Jiao et al. [25]. The feasibility of a solution $\mathbf{x} \in \mathcal{X}$ is determined by whether it satisfies all constraints $g(\mathbf{x})$. An essential aspect of optimization lies in efficiently transitioning from an infeasible to a feasible situation to reduce computational costs. In our methodology, we initially identify infeasible solutions by evaluating them against all defined constraints $G_i(\mathbf{x})$.

In the following, we introduce the concept of Constrained Expected Improvement to aid this process. For a solution $\mathbf{x}$, we measure the extent of constraint violation by defining

$$G_i^+(\mathbf{x}) = \max\{0, l_i - G_i(\mathbf{x}), G_i(\mathbf{x}) - u_i\}, \quad (23)$$

where l_i and u_i denote the lower and upper constraint boundaries, respectively. We represent the constraint violation of a solution as the maximum violation across all constraints,

$$G^+(\mathbf{x}) = \max_{i=1,2,\dots,m} \{G_i^+(\mathbf{x})\}. \quad (24)$$

This value is zero for feasible solutions and positive for infeasible ones. A solution $\mathbf{x}$ is deemed infeasible if it violates any of these constraints, i.e., if $G_i^+(\mathbf{x}) > 0$ for at least one constraint. This preliminary step is crucial for distinguishing between solutions that require improvement to meet feasibility criteria and those that are already feasible.

Let $g_{\min}^{+N}$ denote the smallest observed constraint violation among all evaluated points up to the current iteration, effectively representing the ‘best’ constraint violation scenario. This value is crucial for assessing the relative improvement of constraint satisfaction for new candidate solutions compared to the existing ones. To quantify the improvement in constraint violation at an untested point $\mathbf{x}$ relative to the current best solution, we define the constrained improvement for infeasible solutions as

$$I_{c,N}(\mathbf{x}) = \begin{cases} g_{\min}^{+N} - G^+(\mathbf{x}), & \text{if } G^+(\mathbf{x}) \leq g_{\min}^{+N}, \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

Here, $I_{c,N}(\mathbf{x})$ quantifies how much the constraint violation at an untested point $\mathbf{x}$ improves relative to the current best solution.

We can now derive the solution $\mathbf{x}$ as follows:

$$\begin{aligned} E[I_{c,N}(\mathbf{x})|g^{mN}] &= \int_0^{g_{\min}^{+N}} (g_{\min}^{+N} - z) \times p_{G^+(\mathbf{x})|g^{mN}}(z) dz \\ &= \int_0^{g_{\min}^{+N}} (g_{\min}^{+N} - z) \times dP_{G^+(\mathbf{x})|g^{mN}}(z) \\ &= \int_0^{g_{\min}^{+N}} P_{G^+(\mathbf{x})|g^{mN}}(z) dz - g_{\min}^{+N} \times P_{G^+(\mathbf{x})|g^{mN}}(0), \end{aligned} \quad (26)$$

where $p_{G^+(\mathbf{x})|g^{mN}}(\cdot)$ represents the probability density function of the random variable conditional on $G^{mN} = g^{mN}$, and $P_{G^+(\mathbf{x})|g^{mN}}(\cdot)$ denotes the cumulative distribution function under the same condition.

For $z \leq 0$,

$$P(G^+(\mathbf{x}) \leq z|g^{mN}) = 0; \quad (27)$$

and for $z > 0$, the computation is as follows:

$$\begin{aligned} &P_{G^+(\mathbf{x})|g^{mN}}(z) \\ &= P\{G^+(\mathbf{x}) \leq z|g^{mN}\} \\ &= P\{[l_1 - G_1(\mathbf{x}) \leq z|g^{mN}] \cap [G_1(\mathbf{x}) - u_1 \leq z|g^{mN}] \cap \dots \\ &\quad \cap [l_m - G_m(\mathbf{x}) \leq z|g^{mN}] \cap [G_m(\mathbf{x}) - u_m \leq z|g^{mN}]\} \\ &= P\{[l_1 - z \leq G_1(\mathbf{x}) \leq u_1 + z|g^{mN}] \cap \dots \cap [l_m - z \leq G_m(\mathbf{x}) \leq u_m + z|g^{mN}]\} \\ &= \int_{l_1-z}^{u_1+z} dg_1 \dots \int_{l_m-z}^{u_m+z} dg_m \times p_{G(\mathbf{x})|g^{mN}}(g_1, \dots, g_m), \end{aligned} \quad (28)$$

where $p_{G(\mathbf{x})|g^{mN}}(g_1, \dots, g_m)$ is the joint Gaussian probability function of $G(\mathbf{x})$ conditioned on g^{mN} . Since the objective and all constraints are mutually independent, $p_{G(\mathbf{x})|g^{mN}}(g_1, \dots, g_m)$ can be computed by

$$\begin{aligned}
& p_{G(\mathbf{x})|\mathbf{g}^{mN}}(g_1, \dots, g_m) \\
&= \prod_{i=1}^m p_{G_i(\mathbf{x})|\mathbf{g}_i^N}(g_i) \\
&= \prod_{i=1}^m \frac{1}{s_{g_i}(\mathbf{x})} \phi\left(\frac{g_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right),
\end{aligned} \tag{29}$$

where $\hat{g}_i(\mathbf{x})$ and $s_{g_i}(\mathbf{x})$ ($i = 1, \dots, m$) are the expectations and variances of the constraints respectively.

By substituting the result $p_{G(\mathbf{x})|\mathbf{g}^{mN}}(g_1, \dots, g_m)$ obtained from Equation (29) into Equation (28), we obtain the cumulative distribution function for $P_{G^+(\mathbf{x})|\mathbf{g}^{mN}}(\cdot)$ under the condition $\mathbf{G}^{mN} = \mathbf{g}^{mN}$ as follows

$$\begin{aligned}
& P_{G^+(\mathbf{x})|\mathbf{g}^{mN}}(z) \\
&= \int_{l_1-z}^{u_1+z} dg_1 \cdots \int_{l_m-z}^{u_m+z} dg_m \times \prod_{i=1}^m \frac{1}{s_{g_i}(\mathbf{x})} \phi\left(\frac{g_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) \\
&= \prod_{i=1}^m \int_{l_i-z}^{u_i+z} \frac{1}{s_{g_i}(\mathbf{x})} \phi\left(\frac{g_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) dg_i \\
&= \prod_{i=1}^m \left[\Phi\left(\frac{u_i + z - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) - \Phi\left(\frac{l_i - z - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) \right].
\end{aligned} \tag{30}$$

Finally, the constrained expected improvement for a solution $\mathbf{x}$ is given by

$$E[I_{c,N}(\mathbf{x})|\mathbf{g}^{mN}] = \int_0^{g_{min}^{+N}} P_{G^+(\mathbf{x})|\mathbf{g}^{mN}}(z) dz - g_{min}^{+N} \times P_{G^+(\mathbf{x})|\mathbf{g}^{mN}}(0), \tag{31}$$

with $P_{G^+(\mathbf{x})|\mathbf{g}^{mN}}(z) = \prod_{i=1}^m [\Phi(\frac{u_i + z - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}) - \Phi(\frac{l_i - z - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})})]$ calculated as mentioned above.

In situations where a feasible point has already been identified, it is desired to maximize the Expected Improvement (EI) of the objective, ensuring that the feasibility conditions are met. This is essentially a quest for a feasible solution that provides the best possible objective value. This strategy is denoted as the Constrained Expected Improvement (CEI), represented by $E[I_{c,N}(\mathbf{x})|\mathbf{f}^N, \mathbf{g}^{mN}]$. We define the improvement in the objective function subject to the satisfaction of constraints as $I_{c,N}(\mathbf{x})$. Assuming that $y(\mathbf{x})$ and $G_i(\mathbf{x})$ for $i = 1, 2, \dots, m$ are mutually independent, the constrained expected improvement based on $\mathbf{y}^N = \mathbf{f}^N$ and $\mathbf{G}^{mN} = \mathbf{g}^{mN}$ is given by

$$E[I_{c,N}(\mathbf{x})|\mathbf{f}^N, \mathbf{g}^{mN}] = E[I(\mathbf{x})|\mathbf{f}^N] \times \prod_{i=1}^m P\{\{l_i \leq G_i(\mathbf{x}) \leq u_i\}|\mathbf{g}_i^N\}, \tag{32}$$

where $\prod_{i=1}^m P\{l_i \leq G_i(\mathbf{x}) \leq u_i|\mathbf{g}_i^N\}$ represents the probability of feasibility (PF), and $E[I(\mathbf{x})|\mathbf{f}^N]$ denotes the expected improvement at the point $\mathbf{x}$.

In the CEI method, the constrained improvement can be defined as

$$I_{c,N}(\mathbf{x}) = \begin{cases} f_{min} - y(\mathbf{x}), & \text{if } y(\mathbf{x}) \leq f_{min} \text{ and } l_i \leq G_i(\mathbf{x}) \leq u_i, \\ 0, & \text{otherwise.} \end{cases} \tag{33}$$

Here, f_{min} is the best value of $y(\mathbf{x})$ over all the test values $\mathbf{f}^N$, and

$$\begin{aligned}
& \prod_{i=1}^m P\{\{l_i \leq G_i(\mathbf{x}) \leq u_i\} | \mathbf{g}_i^N\} \\
&= P\{\{l_1 \leq G_1(\mathbf{x}) \leq u_1\} | \mathbf{g}_1^N\} \times P\{\{l_2 \leq G_2(\mathbf{x}) \leq u_2\} | \mathbf{g}_2^N\} \times \cdots \times P\{\{l_m \leq G_m(\mathbf{x}) \leq u_m\} | \mathbf{g}_m^N\} \\
&= \int_{l_1}^{u_1} p_{G_1(\mathbf{x}) | \mathbf{g}_1^N}(g_1) dg_1 \int_{l_2}^{u_2} p_{G_2(\mathbf{x}) | \mathbf{g}_2^N}(g_2) dg_2 \cdots \int_{l_m}^{u_m} p_{G_m(\mathbf{x}) | \mathbf{g}_m^N}(g_m) dg_m \\
&= \prod_{i=1}^m \left[\Phi\left(\frac{u_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) - \Phi\left(\frac{l_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) \right].
\end{aligned} \tag{34}$$

Using this, we can now define the Constrained Expected Improvement as

$$E[I_{c,N}(\mathbf{x}) | \mathbf{f}^N, \mathbf{g}^{mN}] = E[I(\mathbf{x}) | \mathbf{f}^N] \times \prod_{i=1}^m \left[\Phi\left(\frac{u_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) - \Phi\left(\frac{l_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) \right]. \tag{35}$$

Here, $E[I(\mathbf{x}) | \mathbf{f}^N]$ is the same with Equation (14), and the CEI of a solution $\mathbf{x}$ is given by

$$\begin{aligned}
E[I_{c,N}(\mathbf{x}) | \mathbf{f}^N, \mathbf{g}^{mN}] &= \left[(f_{\min} - \hat{f}(\mathbf{x})) \Phi\left(\frac{f_{\min} - \hat{f}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x}) \phi\left(\frac{f_{\min} - \hat{f}(\mathbf{x})}{s(\mathbf{x})}\right) \right] \\
&\quad \times \prod_{i=1}^m \left[\Phi\left(\frac{u_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) - \Phi\left(\frac{l_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) \right].
\end{aligned} \tag{36}$$

As previously defined in Equation (1), the special case for COP formula has a lower constraint bound of $\mathbf{l} = -\infty$ and an upper constraint bound of $\mathbf{u} = 0$. Equations (31) and (36) for CEI can be summarized as follows:

For an infeasible situation:

$$E[I_{c,N}(\mathbf{x}) | \mathbf{g}^{mN}] = \int_0^{g_{\min}^{+N}} \prod_{i=1}^m \Phi\left(\frac{z - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) dz - g_{\min}^{+N} \times \prod_{i=1}^m \Phi\left(\frac{-\hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right). \tag{37}$$

For a feasible situation:

$$\begin{aligned}
E[I_{c,N}(\mathbf{x}) | \mathbf{f}^N, \mathbf{g}^{mN}] &= \left[(f_{\min} - \hat{f}(\mathbf{x})) \Phi\left(\frac{f_{\min} - \hat{f}(\mathbf{x})}{s(\mathbf{x})}\right) + s(\mathbf{x}) \phi\left(\frac{f_{\min} - \hat{f}(\mathbf{x})}{s(\mathbf{x})}\right) \right] \\
&\quad \times \prod_{i=1}^m \Phi\left(\frac{-\hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right).
\end{aligned} \tag{38}$$

3.2. Constrained Expected Prediction Error

Building upon the preceding subsection, which discussed the workings of Constrained Expected Improvement (CEI) under both infeasible and feasible situations, we now move our focus towards the Constrained Expected Prediction Error (CEPE). It is our novel method for constrained optimization problems.

In terms of infeasible situations, our approach with CEPE aligns with the methods used in the preceding CEI subsection. In terms of feasible situations of the CEPE method, we introduce the concept of prediction error, denoted as $PE_{c,N}(\mathbf{x})$, which is defined as follows:

$$PE_{c,N}(\mathbf{x}) = \begin{cases} (f(\mathbf{x}) - f(\mathbf{x}_*))^2, & \text{if } l_i \leq G_i(\mathbf{x}) \leq u_i, \\ 0, & \text{otherwise.} \end{cases} \tag{39}$$

Here, $f(\mathbf{x}_*)$ is the observed output sample at the sample point $\mathbf{x}_*$, which is the nearest evaluated point to the candidate point $\mathbf{x}$. This approach assumes that $f(\mathbf{x})$ exhibits moderate continuity in the vicinity of $\mathbf{x}_*$, an assumption that is generally valid for smooth functions commonly encountered in engineering and optimization applications.

$$E[PE_{c,N}(\mathbf{x})|\mathbf{f}^N] = (\hat{f}(\mathbf{x}_*) - \hat{f}(\mathbf{x}))^2 + s^2(\mathbf{x}). \quad (40)$$

In this formulation, $\hat{f}(\mathbf{x}_*)$ is the surrogate model's prediction at the point $\mathbf{x}_*$, and $\hat{f}(\mathbf{x})$ is the model's prediction at the candidate point $\mathbf{x}$. The term $s^2(\mathbf{x})$ represents the prediction variance at $\mathbf{x}$, reflecting the model's uncertainty.

Then we can obtain the formulation of the CEPE of a solution $\mathbf{x}$ below,

$$\begin{aligned} E[PE_{c,N}(\mathbf{x})|\mathbf{f}^N, \mathbf{g}^{mN}] &= E[PE_{c,N}(\mathbf{x})|\mathbf{f}^N] \times \prod_{i=1}^m [\Phi(\frac{u_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}) - \Phi(\frac{l_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})})] \\ &= [(\hat{f}(\mathbf{x}_*) - \hat{f}(\mathbf{x}))^2 + s^2(\mathbf{x})] \times \prod_{i=1}^m \left[\Phi\left(\frac{u_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) - \Phi\left(\frac{l_i - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) \right]. \end{aligned} \quad (41)$$

This formulation takes into account the constraints by incorporating the product of the cumulative distribution functions for the constraints. It effectively combines the expected prediction error with the feasibility of the solution, guiding the optimization process towards regions of the design space where the model uncertainty is high and potential improvements in the objective function are likely, while still adhering to the constraints.

In summary, the CEPE method is underpinned by the assumption of moderate function continuity near $\mathbf{x}_*$, allowing the use of surrogate model predictions at nearby points to estimate the prediction error at untested points. This approach enables the efficient exploration of the design space, especially in the context of expensive function evaluations and stringent constraints.

Similarly, as previously defined in Equation (1), the special case for COP formula has a lower constraint bound of $\mathbf{l} = -\infty$ and an upper constraint bound of $\mathbf{u} = 0$. Equations (31) and (41) for CEPE can be summarized as follows:

For an infeasible situation:

$$E[PE_{c,N}(\mathbf{x})|\mathbf{g}^{mN}] = \int_0^{g_{min}^{+N}} \prod_{i=1}^m \Phi\left(\frac{z - \hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right) dz - g_{min}^{+N} \times \prod_{i=1}^m \Phi\left(\frac{-\hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right). \quad (42)$$

For a feasible situation:

$$E[PE_{c,N}(\mathbf{x})|\mathbf{f}^N, \mathbf{g}^{mN}] = [(\hat{f}(\mathbf{x}_*) - \hat{f}(\mathbf{x}))^2 + s^2(\mathbf{x})] \prod_{i=1}^m \Phi\left(\frac{-\hat{g}_i(\mathbf{x})}{s_{g_i}(\mathbf{x})}\right). \quad (43)$$

3.3. The GP Surrogate-Assisted DE Algorithm

In this section, we outline and describe the steps of the GP surrogate-assisted Differential Evolution (DE) algorithm. The algorithm applies to problems with any number of dimensions n and any number of constraints m . The steps are as follows:

- **Step 1: Design of Experiments (DOE)**
Initialize the algorithm by assigning samples using the Latin Hypercube Design (LHD). The steps are as follows:
 1. Divide each dimension into *popsiz*e intervals of equal probability.
 2. Randomly assign one value to each interval within each dimension.
 3. For each dimension, permute the vector of assigned values randomly.
 4. Combine the permuted vectors of assigned values across dimensions into a *popsiz*e $\times n$ matrix.
- **Step 2: Identify the Best Point and Assess Feasibility**
Extract the objective function values and constraint violation values from the initial samples. Identify the feasible solutions (those with no constraint violations) and select the one with the optimal objective function value as the best feasible solution. If no feasible solutions exist, select the solution with the least constraint violation as the

best infeasible solution. Update the data structure with information about the best solution discovered thus far, and return this information along with a feasibility flag.

- **Step 3: Clustering—Organize Tested Points into Clusters**

Given N tested points $x_1, x_2, \dots, x_N$, if $N \leq L_1$ (where L_1 is the maximum number of points a local model can contain), use all the points to build a single local model. If $N > L_1$, apply fuzzy clustering to divide the points into c_{size} clusters, where

$$c_{size} = 1 + \left\lceil \frac{N - L_1}{L_2} \right\rceil,$$

and $L_1 > L_2$ (where L_2 is the number of points for adding one more local model). The clustering minimizes the following function J :

$$J = \sum_{i=1}^N \sum_{j=1}^{c_{size}} u_{ij}^{\alpha} \|x_i - v_j\|^2,$$

where α is a constant greater than 1, v_j is the centre of the cluster j , u_{ij} represents the membership degree of x_i in cluster j , and $\|\cdot\|$ denotes the Euclidean norm. v_j can be calculated as

$$v_j = \frac{\sum_{i=1}^N (u_{ij}^t)^{\alpha} x_i}{\sum_{i=1}^N (u_{ij}^t)^{\alpha}}.$$

Initialize u_{ij}^0 for $i = 1, 2, \dots, N$ and $j = 1, 2, \dots, c_{size}$, and set $t = 0$. Compute u_{ij}^{t+1} using

$$u_{ij}^{t+1} = \frac{1}{\sum_{k=1}^{c_{size}} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{\alpha-1}}}.$$

If $\max_{1 \leq i \leq N, 1 \leq j \leq c_{size}} |u_{ij}^{t+1} - u_{ij}^t| < \epsilon$, terminate and output v_j and $u_{ij} = u_{ij}^{t+1}$; otherwise, increment t and recalculate v_j and u_{ij}^{t+1} until the stopping criterion ϵ is satisfied.

- **Step 4: Modelling—Build Local GP Surrogate Models**

For each cluster, construct local Gaussian Process (GP) surrogate models for the objective function and the constraint functions separately.

- **Step 5: Differential Evolution—Generate and Evaluate Candidate Points**

Use Differential Evolution (DE) [27] to generate NP candidate points and evaluate them using the surrogate models. DE is a population-based optimization algorithm that aims to find the global minimum of a given objective function within a high-dimensional search space. The algorithm employs mutation, crossover, and selection strategies to evolve the population towards the optimal solution. During the mutation step, new candidate solutions are generated by combining existing solutions with a weighted difference vector, where the scaling factor is denoted by F . In the crossover step, these candidates are combined with the original population to form a trial population, with a defined crossover probability CR .

Incorporate the Constrained Expected Improvement (CEI) and Constrained Expected Prediction Error (CEPE) criteria, derived from the Gaussian Process (GP) surrogate models, into the evaluation process of candidate points. By computing the CEI and CEPE values for each candidate point, the algorithm leverages these metrics to guide the selection process within DE. Candidate points are prioritized based on higher CEI values, indicating a greater expected improvement, or lower CEPE values, signifying a lower prediction error. This strategic prioritization ensures the exploration of regions in the search space with the highest potential for improvement while adhering to the constraints. This iterative process continues until a termination criterion is met, such

as reaching the maximum number of generations $Maxt$ or achieving a predetermined level of convergence. The flowchart of the DE algorithm, highlighting the integration of CEI and CEPE, is depicted in Figure 1.

- **Step 6: Evaluation—Use Original Objective and Constraint Functions**
Evaluate the NP candidate points using the original objective and constraint functions. This step can also be considered as the final selection step of Differential Evolution.
- **Step 7: Iteration—Update Data and Repeat Steps 3 to 6**
Incorporate new data from the observations into the model, and repeat Steps 3 to 6 until the termination condition is reached. The termination condition is typically based on the number of function evaluations, denoted as FEs .

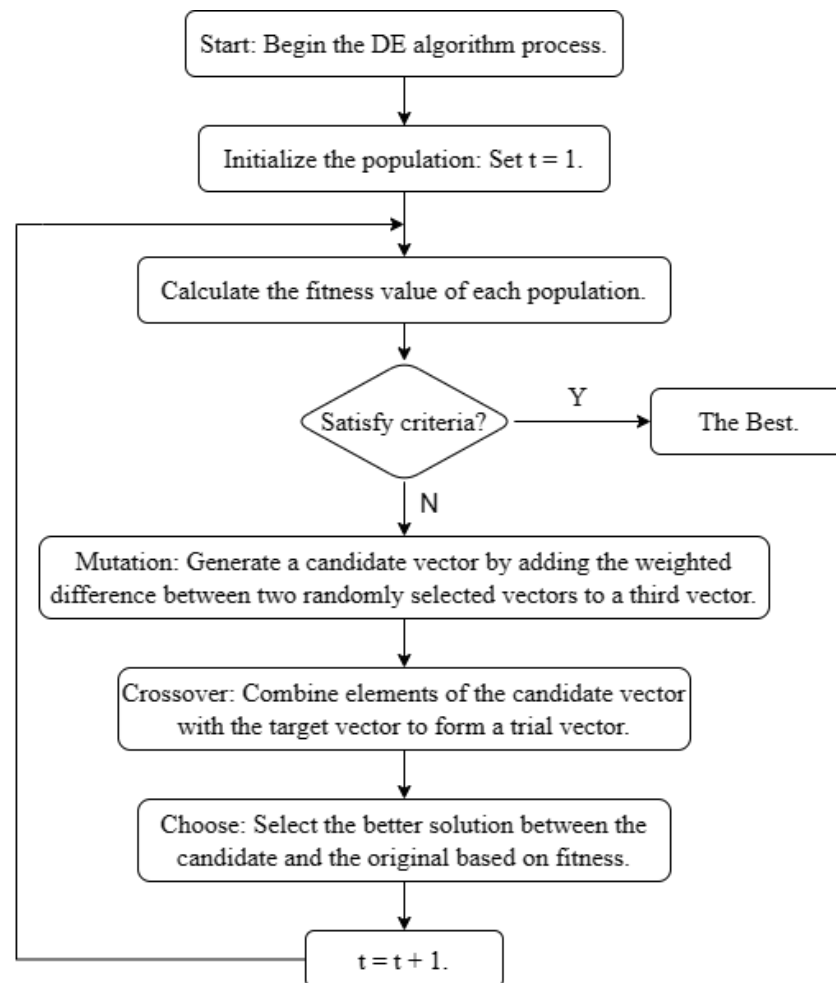


Figure 1. Flowchart of the Differential Evolution Algorithm.

4. Simulation

In this section, we illustrate the efficacy of the proposed CEI (Constrained Expected Improvement) and CEPE (Constrained Expected Prediction Error) methods for addressing constrained optimization problems. We begin by introducing traditional benchmark problems, and then undertake a comparison and evaluation of the two methods by analyzing statistics for each problem.

4.1. Benchmark Problem Description

We employ the classic set of problems presented in CEC 2006 by Liang et al. [28], which was proposed as part of the IEEE Congress on Evolutionary Computation in 2006, and all problems we solve including the modified parts can be found by Chaiyotha and Krityakierne [24]. This suite encompasses 25 real-parameter optimization problems, rep-

representing a diverse array of characteristics. GP (Gaussian Process) is advantageous in low-dimensional settings (i.e., when the number of dimensions $n \leq 10$) [29]. Consequently, we select problems with suitable dimensions n and numbers of constraints m , as higher dimensions would entail significantly extended computation times with our GP surrogate-assisted DE algorithm. Nine problems are chosen, including two that we modify. The characteristics of these problems can be found in Table 2, where *Prob* denotes the name of the problems, *Dim* represents the number of decision variables or dimensions, $f(\mathbf{x}^*)$ signifies the best-known results, and ρ is the approximate ratio of feasible solutions within the search space (inclusive of both infeasible and feasible solutions).

Table 2. Summary of benchmark characteristics.

<i>Prob</i>	<i>Dim</i>	Type of f	$f(\mathbf{x}^*)$	ρ
G02mod	2	Nonlinear	-	-
G03mod	2	Polynomial	-	-
G04	5	Quadratic	−30665.539	52.123%
G06	2	Cubic	−6961.8139	0.0066%
G08	2	Nonlinear	−0.095825	0.8560%
G09	7	Polynomial	680.630	0.5121%
G11	2	Quadratic	0.7499	0.0000%
G12	3	Quadratic	−1.0	4.7713%
G24	2	Linear	−5.508	79.6556%

4.2. Experimental Settings

As discussed in Section 3.3 regarding the GP surrogate-assisted DE algorithm, we will specify certain parameters for solving these numerical simulations, follows closely the methodologies and values reported in existing literature, particularly the work by Jiao et al. [25]. This reference provided a foundational basis for our initial parameter settings, ensuring consistency and comparability with established practices in the field. The problems have n dimensions and m constraints. In the Latin Hypercube Design (LHD), we generate $(11n - 1)$ initial samples, which serve as our population size (*popsiz*). The parameters utilized in the fuzzy clustering are $L_1 = 80$, $L_2 = 20$, $\alpha = 2$, and $\epsilon = 0.5$. In the Differential Evolution (DE) process, we set the population size (NP) to 30, the number of generations to 500, the crossover probability (CR) to 0.9, and the scaling factor (F) to 0.5. In the CEPE method, the number of candidates is set to $50 \times n$. The number of function evaluations (FES) is also set to $50 \times n$. Each benchmark problem is repeated 100 times using both methodologies (CEI and CEPE) with identical parameters to ensure fairness in the comparison.

The experiments were conducted on a personal computer with an Intel(R) Core(TM) i5-8265U CPU @ 1.60 GHz (turbo up to 1.80 GHz). The Intel Corporation is based in Santa Clara, California, USA. The computational analysis and simulations were performed using MATLAB version 2020b. This setup provided a consistent and controlled environment for all computational experiments, ensuring the reliability and reproducibility of our results.

4.3. Results

Figure 2 showcases box plots for nine different benchmark problems. Each box plot displays the distribution of values obtained from 100 independent experiments using two methods: CEI and CEPE. Additionally, Table 3 lists the mean, standard deviation (sd), best, and worst values of the objective function obtained by these methods after 100 iterations. The mean and standard deviation values indicate the quality of the best feasible solutions across the 100 independent experiments. The best and worst entries signify the lowest and highest values of the best feasible solution obtained from the 100 independent experiments for each method, respectively.

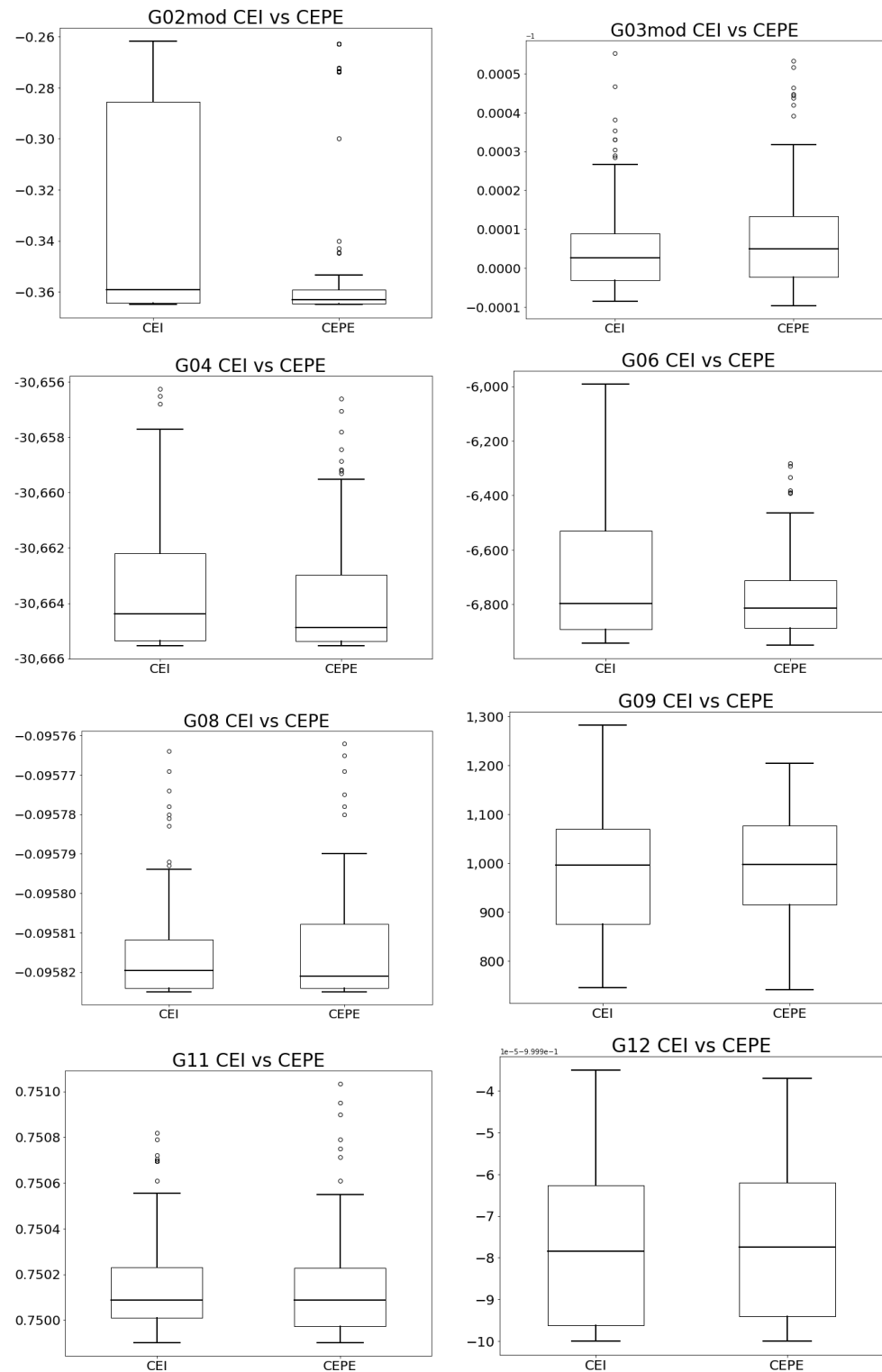


Figure 2. Cont.

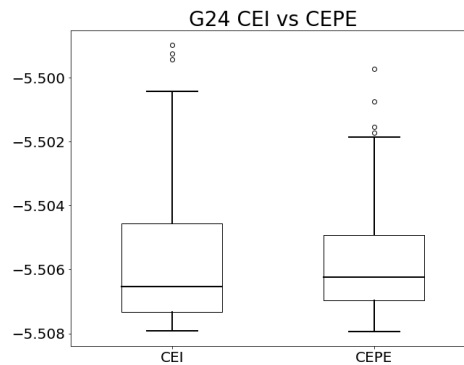


Figure 2. Box plots of the best feasible solution after 100 repetitions by CEI and CEPE in nine benchmark problems.

Table 3. Function values (mean, standard deviation, best, and worst) achieved by CEI and CEPE methods after 100 independent repetitions. Values in bold indicate the superior results across comparisons.

		CEI	CEPE
G02	mean	−0.3360827	−0.3510043
	sd	0.0398702	0.02986116
	best	−0.364923	−0.364946
	worst	−0.261689	−0.262792
G03	mean	−0.999943	−0.9999153
	sd	0.000125786	0.000146517
	best	−1.000086	−1.000097
	worst	−0.999447	−0.999467
G04	mean	−30663.44	−30663.82
	sd	2.397651	2.18125
	best	−30665.54	−30665.54
	worst	−30656.25	−30656.61
G06	mean	−6673.657	−6761.299
	sd	280.4894	172.4326
	best	−6942.869	−6949.9
	worst	−5990.58	−6282.121
G08	mean	−0.0958143	−0.09581347
	sd	1.36×10^{-5}	1.51×10^{-5}
	best	−0.095825	−0.095825
	worst	−0.095764	−0.095762
G09	mean	988.2063	995.5317
	sd	128.8924	110.2954
	best	744.6612	740.9385
	worst	1282.99	1204.271
G11	mean	0.7501684	0.7501476
	sd	0.000233982	0.000241632
	best	0.749901	0.749902
	worst	0.750818	0.751034
G12	mean	−0.9999771	−0.9999772
	sd	1.89×10^{-5}	1.85×10^{-5}
	best	−1	−1
	worst	−0.999935	−0.999937
G24	mean	−5.505601	−5.505723
	sd	0.002281037	0.00178044
	best	−5.50792	−5.507943
	worst	−5.498976	−5.499741

Examining Table 3, we observe that in six of the benchmark problems (G02mod, G04, G06, G11, G12, and G24), the CEPE method yields lower mean values compared to the CEI method, indicating better performance. Conversely, the CEI method performs better in the other three problems. Moreover, the standard deviation values mostly align with the mean values, with seven benchmark problems (G02mod, G03mod, G04, G06, G08, G12, and G24) exhibiting lower standard deviations for the method that has the lower mean. Regarding the best values, CEPE outperforms or matches CEI in eight out of nine problems, with the exception being G11, where CEI has a slight edge. For the worst values, CEPE is superior in seven out of nine problems, with CEI prevailing only in G08 and G11.

To statistically compare the performance differences between CEI and CEPE, a non-parametric Wilcoxon rank-sum test was employed. This test is particularly suitable when the data is not normally distributed, and it assumes equal variances between groups. The null hypothesis posits that the two methods yield the same distribution of values, whereas the alternative hypothesis contends that the distributions differ. A test statistic representing the sum of the ranks of the values is computed. A significant test statistic indicates a notable difference between the two methods, and a p -value below 0.05 is deemed statistically significant. The notation $X < Y$ indicates that method X is significantly better than method Y at the 5% significance level, whereas $X \approx Y$ signifies that there is no significant difference between the two methods. The results are compiled in Table 4. The table reveals that for most benchmark problems, CEI and CEPE exhibit similar performance, with CEPE being significantly better than CEI only for problem G02.

Table 4. Wilcoxon signed-ranks test results at the 5% significance level.

Problem	p -Value	Results
G02mod	0.001996	CEPE < CEI
G03mod	0.1895	CEPE $\approx$ CEI
G04	0.2478	CEPE $\approx$ CEI
G06	0.1301	CEPE $\approx$ CEI
G08	0.7026	CEPE $\approx$ CEI
G09	0.5071	CEPE $\approx$ CEI
G11	0.3431	CEPE $\approx$ CEI
G12	0.9248	CEPE $\approx$ CEI
G14	0.4606	CEPE $\approx$ CEI

5. Application

In this section, we will apply the GP surrogate-assisted DE algorithm to the Three-Bar Truss Design problem. The Three-Bar Truss Design problem [30,31] is a classical optimization problem that is widely studied within the engineering and mathematical communities. The objective is to design a truss structure, composed of three bars, capable of withstanding a specified load while minimizing weight. This problem is particularly significant in the design of lightweight and efficient structures in fields such as aerospace and civil engineering. Moreover, it serves as a quintessential example of how optimization algorithms can address complex societal and civilizational challenges. The problem can be formulated as follows:

$$\begin{aligned}
 &\text{minimize} && f(\mathbf{x}) = (2\sqrt{2}x_1 + x_2) \times l, \\
 &\text{subject to :} && \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0, \\
 &&& \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0, \\
 &&& \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0,
 \end{aligned} \tag{44}$$

where $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$; $l = 100$ cm, $P = 2$ KN/cm², and $\sigma = 2$ KN/cm².

For the experiment, we employ the same parameter settings as in the numerical simulations. The problem has two dimensions and three constraints. Using Latin hypercube design (LHD), we generate 21 initial samples for our population size. The parameters in the fuzzy cluster are set to $L_1 = 80$, $L_2 = 20$, $\alpha = 2$, and $\epsilon = 0.5$. In Differential Evolution (DE), the population size is assumed to be $NP = 30$, the number of generations is set to 500, the crossover probability is $CR = 0.9$, and the scaling factor is $F = 0.5$. In the CEPE method, we set the candidate number to 100. The number of function evaluations (FEs) is 100. The experiment is repeated 100 times using two different methodologies (CEI and CEPE), and the parameters are kept the same to ensure a fair comparison.

Figure 3 depicts box plots that display the distribution of the objective function values obtained using the CEI and CEPE methods across 100 independent experiments. Table 5 lists the mean, standard deviation (sd), best, and worst of the objective function values obtained by both methods over 100 iterations. The best-known result for this problem is approximately 263.8958. From the table, it is evident that CEPE performs better in terms of mean, standard deviation, and worst objective function values, while CEI marginally outperforms CEPE in the best objective function value.

Additionally, we employ the Wilcoxon rank-sum test to statistically compare the performance of the CEI and CEPE methods. With a p -value of 0.0538%, the test indicates that the CEPE method is significantly superior to the CEI method.

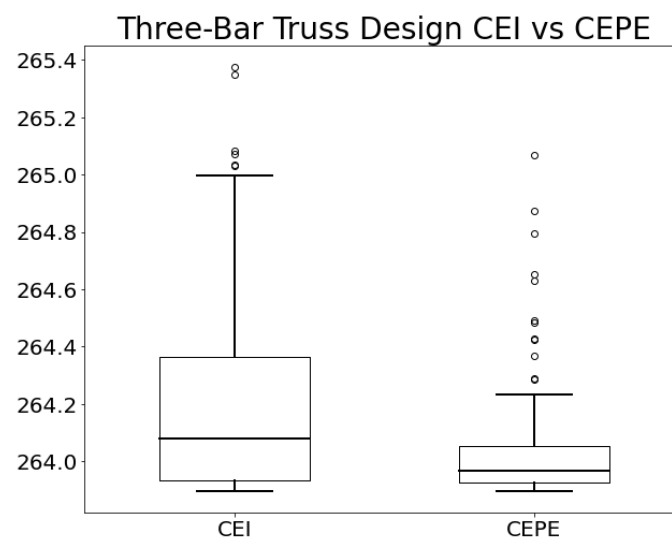


Figure 3. Box plots representing the distribution of the best feasible solutions obtained after 100 repetitions using the CEI and CEPE methods in the Three-Bar Truss Design problem.

Table 5. Function values (mean, standard deviation, best and worst) achieved by CEI and CEPE methods after 100 independent repetitions. Values in bold indicate the superior results across comparisons.

		CEI	CEPE
Three-Bar Truss Design	mean	264.2152	264.0456
	sd	0.3605	0.2208
	best	263.8962	263.8972
	worst	265.3764	265.0683

6. Discussion and Remarks

In this paper, we introduced the formulation of constrained optimization problems, the methodologies of Constrained Expected Improvement (CEI) and Constrained Expected Prediction Error (CEPE) in conjunction with the Gaussian Process (GP) surrogate-assisted Differential Evolution (DE) algorithm, and presented numerical benchmark simulations and applications. Additionally, we would like to emphasize the potential application of these methodologies in the field of engineering, where optimization plays a crucial role.

The implementation of CEI and CEPE has showcased their capacity for the efficient utilization of computational resources, making them highly advantageous in engineering contexts where simulation models can be computationally demanding. Their adeptness in managing multiple constraints, robustness against uncertainties, and flexibility to adapt to various problem domains highlight the methodologies' applicability in real-world engineering problems. These attributes facilitate the optimization of processes, enhancement of product yields, reduction of waste, and improvement in safety and environmental performance.

Despite these advantages, the methodologies do present limitations, such as potential inflexibility in highly specialized engineering challenges and sensitivity to parameter configuration, which could impact their effectiveness. Moreover, the computational intensity required by CEI and CEPE, especially for large-scale problems, warrants consideration. These limitations suggest a cautious approach to their application, emphasizing the need for tailoring or integrating them with other techniques to meet specific engineering requirements.

In summary, the findings from this study affirm the promise of CEI and CEPE in advancing surrogate-assisted optimization strategies for engineering applications. While addressing the identified gaps in constrained optimization, this research also paves the way for future investigations into more sophisticated surrogate models and optimization techniques. Further exploration in this domain could yield even more robust and efficient solutions for complex optimization challenges, contributing to the continuous advancement of engineering optimization practices.

Author Contributions: Conceptualization, H.C. and B.W.; methodology, H.C. and B.W.; software, H.C. and Z.W.; validation, H.C.; formal analysis, H.C. and B.W.; investigation, H.C.; resources, H.C.; writing—original draft preparation, H.C.; writing—review and editing, H.C. and B.W.; visualization, H.C.; supervision, B.W.; project administration, H.C. and B.W.; funding, H.C. and B.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by Chinese Scholarship Council (No.202008060052) and in part by University of Leicester (No. CSE-MA5-WANG).

Data Availability Statement: The data generated for this study are derived from formulae detailed in the referenced literature and further processed using the author's methodologies. They are available from the corresponding author upon reasonable request. The data are not publicly available as they are unique to the study's bespoke computational analysis.

Acknowledgments: The authors thank the editor and the reviewers for their very helpful and constructive comments.

Conflicts of Interest: Author Zhe Wang was employed by the company Covestro (Netherlands). The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The Covestro (Netherlands) had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CEI	Constrained Expected Improvement
CEPE	Constrained Expected Prediction Error
COPs	Constrained Optimization Problems
DE	Differential Evolution
EGO	Efficient Global Optimization
EI	Expected Improvement
EPE	Expected Prediction Error
GPR	Gaussian Process Regression

References

1. Banks, A.; Vincent, J.; Anyakoha, C. A review of particle swarm optimization. Part II: Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Nat. Comput.* **2008**, *7*, 109–124. [\[CrossRef\]](#) [\[CrossRef\]](#)
2. Schueller, G.I.; Jensen, H.A. Computational methods in optimization considering uncertainties—An overview. *Comput. Methods Appl. Mech. Eng.* **2008**, *198*, 2–13. [\[CrossRef\]](#) [\[CrossRef\]](#)
3. Kumar, A.; Wu, G.; Ali, M.Z.; Mallipeddi, R.; Suganthan, P.N.; Das, S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm Evol. Comput.* **2020**, *56*, 100693. [\[CrossRef\]](#) [\[CrossRef\]](#)
4. Akyol, D.E.; Bayhan, G.M. A review on evolution of production scheduling with neural networks. *Comput. Ind. Eng.* **2007**, *53*, 95–122. [\[CrossRef\]](#) [\[CrossRef\]](#)
5. Kim, T.H.; Maruta, I.; Sugie, T. A simple and efficient constrained particle swarm optimization and its application to engineering design problems. *Proc. Inst. Mech. Eng. Part J. Mech. Eng. Sci.* **2010**, *224*, 389–400. [\[CrossRef\]](#) [\[CrossRef\]](#)
6. Martelli, E.; Amaldi, E. PGS-COM: A hybrid method for constrained non-smooth black-box optimization problems: Brief review, novel algorithm and comparative evaluation. *Comput. Chem. Eng.* **2014**, *63*, 108–139. [\[CrossRef\]](#) [\[CrossRef\]](#)
7. Rao, R. Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. *Decis. Sci. Lett.* **2016**, *5*, 1–30. [\[CrossRef\]](#)
8. Guedria, N.B. Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Appl. Soft Comput.* **2016**, *40*, 455–467. [\[CrossRef\]](#) [\[CrossRef\]](#)
9. Agarwal, H.; Renaud, J.E.; Preston, E.L.; Padmanabhan, D. Uncertainty quantification using evidence theory in multidisciplinary design optimization. *Reliab. Eng. Syst. Saf.* **2004**, *85*, 281–294. [\[CrossRef\]](#) [\[CrossRef\]](#)
10. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [\[CrossRef\]](#) [\[CrossRef\]](#)
11. Maier, H.R.; Kapelan, Z.; Kasprzyk, J.; Kollat, J.; Matott, L.S.; Cunha, M.C.; Dandy, G.C.; Gibbs, M.S.; Keedwell, E.; Marchi, A.; et al. Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. *Environ. Model. Softw.* **2014**, *62*, 271–299. [\[CrossRef\]](#) [\[CrossRef\]](#)
12. Kashani, A.R.; Camp, C.V.; Rostamian, M.; Azizi, K.; Gandomi, A.H. Population-based optimization in structural engineering: A review. *Artif. Intell. Rev.* **2022**, *345*–452. [\[CrossRef\]](#)
13. Forrester, A.; Sobester, A.; Keane, A. *Engineering Design via Surrogate Modelling: A Practical Guide*; John Wiley & Sons: Hoboken, NJ, USA, 2008. [\[CrossRef\]](#)
14. Box, G.E.P.; Wilson, K.B. On the Experimental Attainment of Optimum Conditions. In *Breakthroughs in Statistics: Methodology and Distribution*; Springer: New York, NY, USA, 1992; pp. 270–310. [\[CrossRef\]](#)
15. Buhmann, M.D. Radial Basis Functions. *Acta Numer.* **2000**, *9*, 1–38. [\[CrossRef\]](#) [\[CrossRef\]](#)
16. Oliver, M.A.; Webster, R. Kriging: A Method of Interpolation for Geographical Information Systems. *Int. J. Geogr. Inf. Syst.* **1990**, *4*, 313–332. [\[CrossRef\]](#) [\[CrossRef\]](#)
17. Gramacy, R.B. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*; CRC Press: Boca Raton, FL, USA, 2020. [\[CrossRef\]](#)
18. Queipo, N.V.; Haftka, R.T.; Shyy, W.; Goel, T.; Vaidyanathan, R.; Tucker, P.K. Surrogate-based analysis and optimization. *Prog. Aerosp. Sci.* **2005**, *41*, 1–28. [\[CrossRef\]](#) [\[CrossRef\]](#)
19. Tabatabaei, M.; Hakanen, J.; Hartikainen, M.; Miettinen, K.; Sindhya, K. A survey on handling computationally expensive multiobjective optimization problems using surrogates: Non-nature inspired methods. *Struct. Multidiscip. Optim.* **2015**, *52*, 1–25. [\[CrossRef\]](#) [\[CrossRef\]](#)
20. Lataniotis, C.; Marelli, S.; Sudret, B. Extending classical surrogate modeling to high dimensions through supervised dimensionality reduction: A data-driven approach. *Int. J. Uncertain. Quantif.* **2020**, *10*, 55–82. [\[CrossRef\]](#) [\[CrossRef\]](#)
21. Zhou, Z.; Ong, Y. S.; Nguyen, M. H.; Lim, D. A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–5 September 2005; pp. 2832–2839. [\[CrossRef\]](#)
22. Jones, D.R.; Schonlau, M.; Welch, W.J. Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* **1998**, *13*, 455. [\[CrossRef\]](#) [\[CrossRef\]](#)
23. Hawe, G.; Sykulski, J. Considerations of accuracy and uncertainty with kriging surrogate models in single-objective electromagnetic design optimisation. *IET Sci. Meas. Technol.* **2007**, *1*, 37–47. [\[CrossRef\]](#) [\[CrossRef\]](#)
24. Chaiyotha, K.; Krityakierne, T. A comparative study of infill sampling criteria for computationally expensive constrained optimization problems. *Symmetry* **2020**, *12*, 1631. [\[CrossRef\]](#) [\[CrossRef\]](#)
25. Jiao, R.; Zeng, S.; Li, C.; Jiang, Y.; Jin, Y. A complete expected improvement criterion for Gaussian process assisted highly constrained expensive optimization. *Inf. Sci.* **2019**, *471*, 80–96. [\[CrossRef\]](#) [\[CrossRef\]](#)
26. Liu, H.; Cai, J.; Ong, Y.-S. An adaptive sampling approach for Kriging metamodeling by maximizing expected prediction error. *Comput. Chem. Eng.* **2017**, *106*, 171–182. [\[CrossRef\]](#) [\[CrossRef\]](#)
27. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evol. Comput.* **2010**, *15*, 4–31. [\[CrossRef\]](#) [\[CrossRef\]](#)
28. Liang, J.J.; Runarsson, T.P.; Mezura-Montes, E.; Clerc, M.; Suganthan, P.N.; Coello, C.A.C.; Deb, K. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. *J. Appl. Mech.* **2006**, *41*, 8–31.

29. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; The MIT Press: Cambridge, MA, USA, 2006. [[CrossRef](#)]
30. Ray, T.; Liew, K.-M. Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Trans. Evol. Comput.* **2003**, *7*, 386–396. [[CrossRef](#)] [[CrossRef](#)]
31. Yıldırım, A.E.; Karci, A. Application of three bar truss problem among engineering design optimization problems using artificial atom algorithm. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 September 2018; pp. 1–5. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.