A Study on the Effectiveness of Constraint Handling Schemes within Efficient Global Optimization Framework

Ahsanul Habib, Hemant Kumar Singh and Tapabrata Ray
School of Engineering and Information Technology, University of New South Wales, Australia.
Email: a.habib@student.adfa.edu.au, {h.singh, t.ray}@adfa.edu.au. Web: www.mdolab.net

***Abstract*—Efficient Global Optimization (EGO) is a well established iterative approach originally introduced to solve computationally expensive unconstrained optimization problems. EGO relies on an underlying Gaussian Process (GP) model and identifies an *infill* location for sampling that maximizes the expected improvement (EI) function. The infill point is evaluated which in turn is used to update the GP model, and this cycle continues until the termination condition is satisfied. In order to deal with constrained optimization problems, several modifications have been suggested in the literature over the years. While the approaches are novel and often complex, the performance is assessed using a small set of test cases (typically two or three). It is thus difficult to judge if they indeed offer significant benefits over simple constrained EGO formulations. In this paper we introduce a simple constrained EGO formulation, where the algorithm attempts to locate infill locations that maximize the probability of feasibility (until a feasible solution is identified) and then switches to maximize the penalized EI function (EI is penalized using the probability of feasibility). The performance of the proposed approach is compared with others using a suite of 10 well studied problems. The results obtained using the proposed approach are competitive and often better than previously reported results for the problems. We hope this study will prompt more interest in the development of efficient constraint handling schemes that can be used within EGO.**

***Index Terms*—Kriging, Efficient Global Optimization, Single Infill Sampling Criterion**

I. Introduction

Over the years, there has been a trend to use increasingly accurate numerical simulations/analysis to design better products and services. Computationally expensive black-box simulations are typically used to compute the constraint and objective functions of the optimization problem. While population based stochastic optimization algorithms are a preferred choice to deal with such non-linear optimization problems, their direct use is impractical since they require evaluation of numerous solutions (designs) prior to convergence. Even with today's computing power, an optimization exercise involving such computationally expensive simulations is a formidable task. A common practice is to employ computationally cheap surrogate/approximate models instead of the true simulations (which are costly) within an optimization algorithm. This category of approaches is commonly referred to as surrogate assisted optimization (SAO). Widely used SAO approaches include Gaussian process (GP models) [1], Multi Layer Perceptrons (MLP) [2], Response Surface Methods (RSM) [3], Radial Basis Functions (RBF) [4], Support Vector Regression (SVR) [5] as the underlying approximation models. Use of such approximation models within a population based stochastic algorithm requires a number of additional considerations which are studied under the broad framework of model management [6].

Jones et. al. [7] proposed Efficient Global Optimization (EGO) for unconstrained optimization which utilizes a GP model and considers the predicted response along with its uncertainty to identify a new infill location. For any potential location, the expected improvement (EI) can be computed based on the predicted response and its variance. This function can then be maximized to obtain the best infill location. After evaluating the solution at the new infill location, the underlying GP model can be updated and the cycle continues until the termination condition is satisfied. The EI value will be large for points where the predicted value is likely to be lower than the best function value and/or where there is high uncertainty in the value of the prediction itself. This EGO approach resembles a serial process. However, the approach can still be expedited for computationally expensive optimization problems if the underlying black-box analysis itself can be parallelized. To achieve greater control on the search behavior, modifications to the expected improvement function have been suggested. Generalized expected improvement function (GEI) [8] is one such modification, where an additional parameter g is used to control the balance between the global and local search. A larger value of g favors exploration, while a smaller value leads to exploitation. Strategies to control the parameter g appear in [9], where the value was progressively reduced over iterations.

Since most real life optimization problems involve constraints, modifications to EGOs have been suggested in the literature to deal with them. The first approach was proposed in [8], where instead of maximizing the GEI function, the product of GEI and the probability of feasibility (PF) was maximized. The probability of feasibility was computed using a GP model for each constraint and PF was calculated as a product of these probabilities. A similar form was also proposed in [10] and referred to as constrained expected improvement. Use of product forms in the computation of PF is known to suffer from single term dominance and is strictly valid when all the constraints are independent [11]. Audet et.

al. [12] introduced the notion of Expected Violation (EV), where a threshold on violation was used instead of the actual violation, thereby allowing greater possibility of sampling around constraint boundaries. The metric was used to assess expected violations of potential sampled solutions using a GP model. Solutions with EV less than a threshold would be further assessed using unconstrained form of EI and a few good solutions would undergo true evaluations. The first screening based on EV attempts to select solutions that are likely to be feasible. In order to allow sampling near constraint boundaries, an user prescribed allowable violation g_{limit} was used instead of 0 in [13] which may not be practical.

The methodology to skip evaluation of solutions that are infeasible inherently limits updates to the constraint approximators. In [14], a threshold violation value (feasibility criterion) was used to allow updates to the constraint functions. An augmented Lagrangian (AL) formulation with a penalty parameter was used to transform the constrained optimization problem to an unconstrained form in [14]. Such a formulation is known to introduce discontinuities and the use of DIRECT algorithm [15] to solve the AL formulation requires the function to be continuous at least in the vicinity of the optimum. Since a product form of PF requires an assumption of constraint independence and also suffers from single term dominance, there have been suggestions to treat EI and each PF corresponding to each constraint as separate objectives in [16]. For a problem with four of more constraints, such a formulation would result in a *many-objective* optimization problem. It has been well established in evolutionary multi-objective optimization (EMO) domain that many-objective problems are significantly harder to solve compared to two or three objective optimization problems [17].

Several variants of sampling strategies have also been proposed in the works of Watson and Barnes [18]. These include means to locate threshold bounded extremal solution, regional extremal solution and one that minimized surprises. The performance of these sampling strategies was reported in [19]. The study revealed that there is no single sampling strategy that is the best for all classes of problems. To inherit the benefits of various sampling strategies, an elaborate approach was introduced in [20], where different infill sampling criterion was used in different stages of the search process. The approach relied on maximizing PF only until a feasible solution was detected. Once a feasible solution was detected, the search attempted to locate other distant feasible points by penalizing the PF with a distance measure (from the identified feasible location). Once several feasible solutions have been located using these strategies, the algorithm would switch to a conventional SAO form attempting to locate solutions with the best objective function value subject to constraints (modeled using a GP model). The final phase would involve constrained optimization attempting to locate distant feasible solutions with performance better than the best feasible solution found so far. The issue of sampling in different regions of the search space in different stages of optimization has been an active area of interest. Recently, a stochastic probability based sampling scheme was introduced [21], where in early phases, infill sampling is preferred, while in later stages boundary sampling is preferred. While the switching is based on the quality of the approximations based on mean squared error (MSE), it is often difficult to provide such MSE thresholds a priori.

To address some of the above concerns, a novel approach was introduced in [11]. A support vector machine classifier (SVM) was used to pre-screen solutions that are likely to be feasible. Among such solutions, one with the highest expected improvement would undergo evaluation. The method delivered promising results on problems with discontinuity and binary constraints. However, the training of SVM classifiers would need large number of training samples and the dataset could soon become unbalanced, thereby requiring a single class classifier as opposed to a two-class classifier used in the study. The approach also had provisions to sample solutions in less explored regions of the search space.

As indicated earlier, this work revisits constrained EGO methods. While the enhancements proposed in the literature are novel and address various drawbacks, this study objectively compares their performance with a simple single objective constrained EGO formulation. While most studies have focused predominantly on the limitations of the probability of feasibility function, we highlight the effect of *reference objective function value* on the EI function. In our proposed approach, the algorithm attempts to locate infill locations that maximize the probability of feasibility (until a feasible solution is identified) and then switches to penalized EI function (EI is penalized using the probability of feasibility) [10] in the next stage. Since PF is used until a feasible solution is identified, the best feasible objective function value is always used as a reference objective function value in the computation of EI.

A brief overview of Gaussian Process (GP) models, expected improvement (EI) and probability of feasibility (PF) is presented in Section II for completeness. The numerical examples are detailed in Section III followed by the description of the proposed algorithm in Section IV. The numerical experiments and analysis are presented in Section V. The conclusions and future research directions are summarized in Section VI.

II. Overview of Gaussian Process, Expected Improvement and Probability of Feasibility

This section provides a brief theoretical overview of Gaussian Process (GP), expected improvement (EI) and probability of feasibility (PF). For the sake of brevity, the key equations are provided without derivation. For detailed explanation and derivation of the equations, readers are recommended to read [22], [23], [24] for Gaussian Process and [10] for EI and constrained EI.

A. Gaussian Process (GP)

In Gaussian Process, the function of interest $y(\boldsymbol{x})$ is expressed as a combination of a global model and localized deviations:

$$y(\boldsymbol{x}) = f(\boldsymbol{x}) + Z(\boldsymbol{x}), \tag{1}$$

where $f(\boldsymbol{x})$ is a polynomial function and $Z(\boldsymbol{x})$ is a Gaussian model with mean 0 and variance σ^2. The co-variance matrix of $Z(\boldsymbol{x})$ is given by:

$$Cov[Z(\boldsymbol{x}^i), Z(\boldsymbol{x}^j)] = \sigma^2 R, \tag{2}$$

where R is a correlation matrix. The correlation between sample i and sample j is denoted by $R(\boldsymbol{x}^i, \boldsymbol{x}^j)$. We assume a Gaussian correlation function with p_k set to 2, as shown in the equation below.

$$R(\boldsymbol{x}^i, \boldsymbol{x}^j) = exp[-\textstyle\sum_{k=1}^{D} \theta_k |x^i{}_k - x^j{}_k|^{p_k}], \tag{3}$$

where D is the dimensionality of $\boldsymbol{x}$, θ_k and p_k are the hyper-parameters and $x^i{}_k$ and $x^j{}_k$ are the k^{th} components of the corresponding $\boldsymbol{x}$.

For a new point $\boldsymbol{x}^{n_s+1}$, the approximated value $\hat{y}(\boldsymbol{x})$ is:

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{r}^T R^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu}) \tag{4}$$

The variance $\hat{\sigma}(\boldsymbol{x})^2$ in prediction at $\boldsymbol{x}$ is given by

$$\hat{\sigma}(\boldsymbol{x})^2 = \sigma^2 \left[1 - r^T R^{-1} r + \tfrac{(1-1^T R^{-1} r)^2}{1^T R^{-1} 1}\right], \tag{5}$$

where $\boldsymbol{y}$ is the column vector of size n_s and $\mathbf{1}$ is a column vector of 1's.

$\hat{\mu}$ is estimated using the following equation:

$$\hat{\mu} = (\mathbf{1}^T R^{-1} \mathbf{1})^{-1} \mathbf{1}^T R^{-1} \boldsymbol{y}, \tag{6}$$

where the variance σ^2 is given by the expression

$$\sigma^2 = \tfrac{1}{n_s}[(\boldsymbol{y} - \mathbf{1}\hat{\mu})^T R^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu})], \tag{7}$$

and $\boldsymbol{r}^T$ is a correlation vector of length n_s between the new point $\boldsymbol{x}^{n_s+1}$ and the n_s sampled data points:

$$\mathbf{r}^T = [R(\boldsymbol{x}^{n_s+1}, \boldsymbol{x}^1), R(\boldsymbol{x}^{n_s+1}, \boldsymbol{x}^2), .., R(\boldsymbol{x}^{n_s+1}, \boldsymbol{x}^{n_s})]^T \tag{8}$$

Finally, the correlation parameters, θ_k can be estimated by maximizing the following likelihood function–

$$\text{Maximize: } -\tfrac{1}{2}(n_s ln(\sigma^2) + ln|R|) \tag{9}$$

The above optimization problem can be solved using a suitable global optimizer.

B. *Expected Improvement (EI)*

Expected improvement at any location $\boldsymbol{x}$ is the expected value by which the predicted objective function value $\hat{y}$ is better than y_{min}. In an unconstrained problem, y_{min} is the best objective function value among all the sampled locations. Thus, EI($\boldsymbol{x}$)=E[$I(\boldsymbol{x})$], where

$$I(\boldsymbol{x}) = max\{y_{min} - \hat{y}(\boldsymbol{x}), 0\}, \tag{10}$$

where y_{min} is the current minimum and $\hat{y}(\boldsymbol{x})$ is the predicted value at location $\boldsymbol{x}$. The expected improvement EI($\boldsymbol{x}$) can be expressed as

$$\text{EI}(\boldsymbol{x}) = \begin{cases} (y_{min} - \hat{y}(\boldsymbol{x}))\ \boldsymbol{\Phi} \times \dots & \\ \dots \left(\frac{(y_{min} - \hat{y}(\boldsymbol{x}))}{\hat{\sigma}(\boldsymbol{x})}\right) + \hat{\sigma}(\boldsymbol{x})\ \boldsymbol{\phi}\left(\frac{(y_{min} - \hat{y}(\boldsymbol{x}))}{\hat{\sigma}(\boldsymbol{x})}\right) & \text{if } \hat{\sigma}(\boldsymbol{x}) > 0 \\ 0 & \text{if } \hat{\sigma}(\boldsymbol{x}) = 0 \end{cases} \tag{11}$$

here, $\boldsymbol{\Phi}$ and $\boldsymbol{\phi}$ are the cumulative density function and the standard normal probability density functions.

C. *Probability of Feasibility (PF)*

While the EI formulation presented above is for unconstrained optimization problems, it is common to use a penalized form of EI to deal with constrained optimization problems. The above EI is multiplied by the probability of feasibility (PF) for constrained optimization problems. Assuming there are q constraints of the form $g_i(\boldsymbol{x}) \leq 0$ and they are independent, the PF is computed as follows:

$$PF = \prod_{i=1}^{q} P(g_i(\boldsymbol{x}) \leq 0) \tag{12}$$

where $P(g_i(\mathbf{x}) \leq 0)$ is evaluated for each constraint as follows

$$P(g_i(\boldsymbol{x}) \leq 0) = \boldsymbol{\Phi}\left(\tfrac{0 - \hat{g}_i(\boldsymbol{x})}{\hat{\sigma}_{g_i}(\boldsymbol{x})}\right) \tag{13}$$

Here, for i^{th} constraint, $\hat{g}_i(\boldsymbol{x})$ and $\hat{\sigma}_{g_i}(\boldsymbol{x})$ are computed using a Gaussian process model.

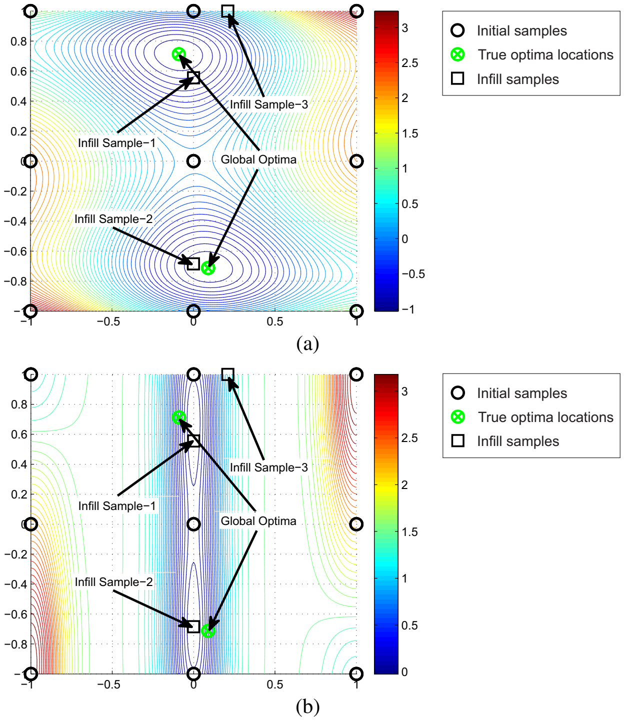
III. Illustrative Example

In this section, we first illustrate the effect of *reference objective function value* on the EI function. We consider the unconstrained Gomez#3 [20] function with two variables.

$$\begin{aligned} &\text{Minimize } f(\boldsymbol{x}) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2 \\ &\qquad \text{where } x_i \in [-1, 1], i = 1, 2 \end{aligned} \tag{14}$$

The unconstrained function has two global optima at $[0.0898, -0.7127]$ and $[-0.0898, 0.7127]$ with an objective function value of -1.0316. A GP model was constructed using 9 initial samples. The best objective function value among these 9 solutions is $y_{min} = 0$. Maximization of EI using the above y_{min} value would result in an infill location of $[0.0010, 0.5553]$. For the sake of discussion, if two different y_{min} values of 1.5 and -1.5 would have been used, the infill locations would turn out to be $[0, -0.6850]$ and $[0.2101, 1]$, as shown in Fig. 1. It is clear that the infill locations can be in completely different regions of the search space depending on the value of reference y_{min} used in EI computation. The EI contours for the three cases are shown in Fig. 2.

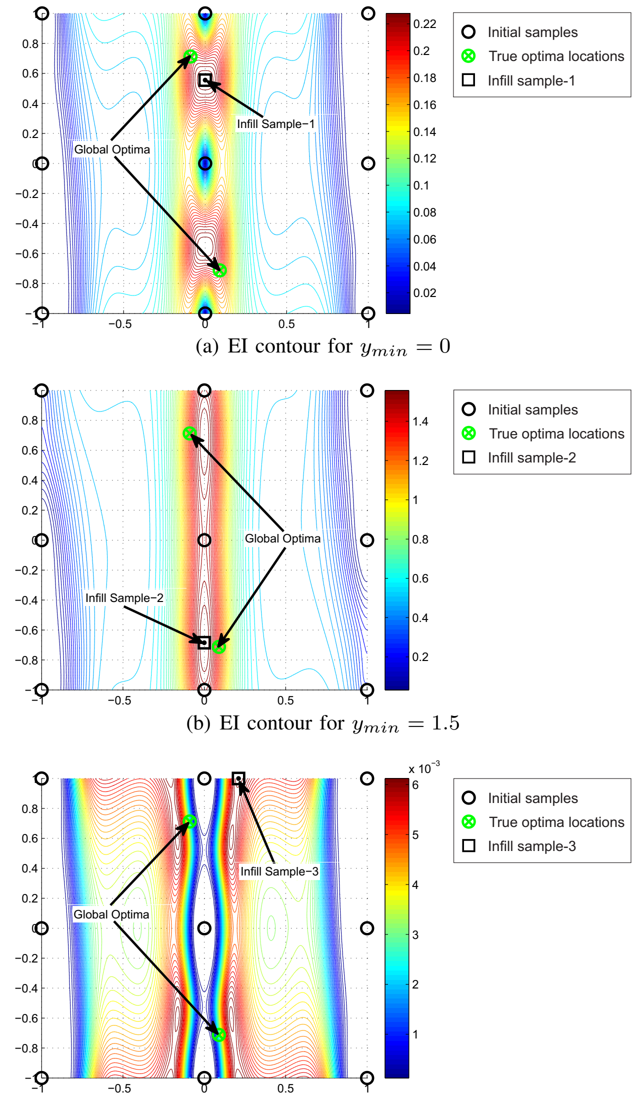
While in the above unconstrained example, we have hypothetically assumed different values of y_{min}, the next example is presented to highlight the differences between two constrained EGO formulations, developed by Forrester et. al. [10] and Schonlau et. al.'s [8]. Consider the following 2 variable constrained optimization problem presented in [9]). There are 3 constraints and two disconnected feasible regions with the global optimum value of -0.7483 at $[0.2017, 0.8332]$.



(a)

(b)

Fig. 1: Location of infill sample points for different values of y_{min}. (a) Infill samples on the contour of f^{true}, (b) Infill samples on the contour of $f^{predicted}$. Infill samples 1, 2, 3 correspond to $f_{min} = 0, 1.5 \; and \; -1.5$ respectively.



(a) EI contour for $y_{min} = 0$

(b) EI contour for $y_{min} = 1.5$

(c) EI contour for $y_{min} = -1.5$

Fig. 2: Contour of the EI function for different values of f_{min} for Gomez#3 function.

$$
\begin{aligned}
\text{Minimize } & f(\boldsymbol{x}) = -(x_1-1)^2 - (x_2-0.5)^2 \\
& g_1(\boldsymbol{x}) \equiv ((x_1-3)^2 + (x_2+2)^2)exp(-x_2^7) - 12 \leq 0 \\
& g_2(\boldsymbol{x}) \equiv 10x_1 + x_2 - 7 \leq 0 \\
& g_3(\boldsymbol{x}) \equiv (x_1-0.5)^2 + (x_2-0.5)^2 - 0.2 \leq 0 \\
& \quad \text{where } x_i \in [0,1], i = 1,2
\end{aligned}
\tag{15}
$$

For this problem, 25 initial samples were used to construct the GP model. The initial samples, contour plot and feasible regions are illustrated in Fig. 3.

From Fig. 3, it can be observed that there is only one initial sample in the feasible region at the location [0.5,0.25]. It has an objective function value -0.3125 and among all the initial samples, the minimum is at [0,0] with an objective value of -1.25, which is an infeasible solution. Thus, $y_{min(all)} = -1.25$ and $y_{min(feas)} = -0.3125$ are two possibilities which can be used in the computation of penalized EI. The effect of employing $y_{min(all)}$ and $y_{min(feas)}$ on the next infill sample is depicted in Fig. 4.

From the figure, the benefit of utilizing the $y_{min(feas)}$ to calculate EI and employing it to find the next infill sample can be observed. Fig. 4(a) depicts the use of $y_{min(all)}$ where, the obtained infill sample is located at $[0.0969, 0.9172]$. The Euclidean distance from this point to the true optimum is 0.1343, which is 9.5% of the solid diagonal based on the variable domain. On the other hand, Fig. 4(b) represents the effect of utilizing $y_{min(feas)}$, where, the next sample point was obtained as $[0.2592, 0.8383]$. It had an Euclidean

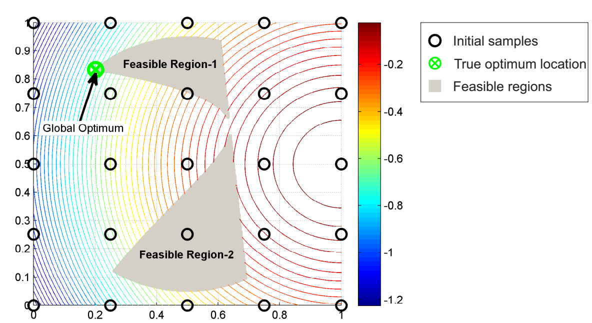


Fig. 3: Contour plot for Sasena function.

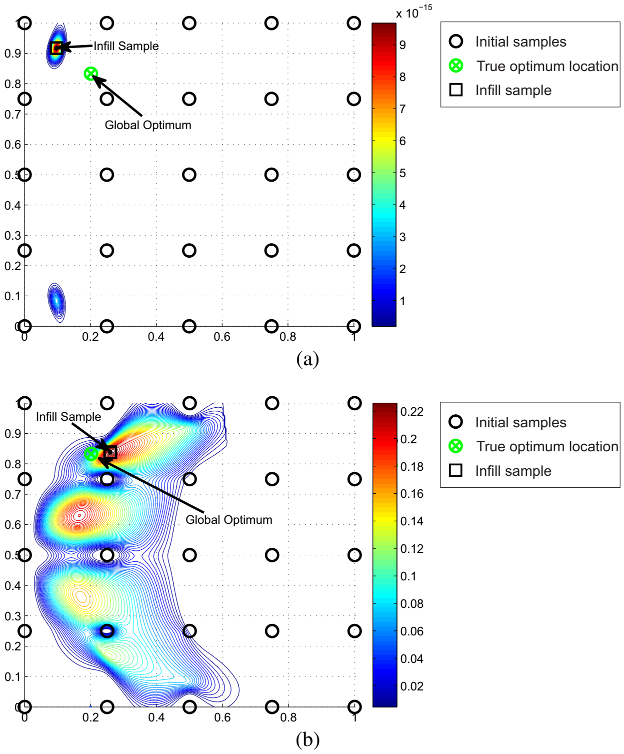


Fig. 4: Location of infill sample points for- (a) Unconstrained formulation of EI penalized with PF, (b) Constrained formulation of EI penalized with PF.

distance of 0.0478 from the true optimum, which corresponds to 4.09% of the solid diagonal distance, which is around 5.5% improvement over the method where calculation of EI was with $y_{min(all)}$.

The above example highlights the fact that the choice of reference objective function values plays an important role in the choice of the next infill location.

IV. Proposed Approach

The motivation of this work stems from the review of constrained EGO literature. The improvements in the literature were targeted towards better control of global and local exploration, infill and boundary sampling strategies and modifications to the probability of feasibility functions. Such modifications come with additional parameters. Besides, some multi-sampling criteria are also studied in the literature employing the conflicting nature of EI and PF [13] or both parts in the EI equation [25]. Although these approaches offer better convergence, they require solving a multiobjective problem in order to obtain the set of infill samples. Other multi-sampling strategies like [26] use expensive multi-dimensional integration methods which can also be deemed as complex and computationally expensive.

In this study, our aim is to develop a single infill sampling criterion based approach which addresses the main shortcoming of the conventional EGO based approaches for constrained optimization problems. In our proposed approach, the algorithm attempts to locate infill locations that maximize the probability of feasibility (until a feasible solution is identified) and then switches to penalized EI function (EI penalized using the probability of feasibility) [10] in the next stage.The pseudo-code of the proposed approach (EGOcons) is presented in Algorithm 1 , while different steps of it are discussed in the following subsections.

Algorithm 1 EGOcons

Input: N = Number of initial samples, q = Number of constraints, **Archive** = repository of all fully evaluated solutions, *Termination Condition* (see detail in Subsection V-B for this study).

$FE = 0$
$\mathbf{pop}_{init} \leftarrow initialize()$
$evaluate(\mathbf{pop}_{init})$
$\mathbf{Archive} \leftarrow archive\ update(\mathbf{pop}_{init})$
$(\boldsymbol{x}_{best}, f_{best}) \leftarrow update\ best\ solution(\mathbf{Archive})$
$\mathbf{Models} = construct\ GP\ models(\mathbf{Archive})$
while *Termination Condition* not met **do**
  $\boldsymbol{x}_{infill} \leftarrow identify\ infill\ sample()$
  $evaluate(\boldsymbol{x}_{infill})$
  $\mathbf{Archive} \leftarrow archive\ update(\mathbf{Archive}, \boldsymbol{x}_{infill})$
  $\mathbf{Models} \leftarrow update\ GP\ models(\mathbf{Archive})$
  $(\boldsymbol{x}_{best}, f_{best}) \leftarrow update\ best\ solution(\mathbf{Archive})$
  $FE = FE + 1$
end while
Return: $\boldsymbol{x}_{best}$ and f_{best}.

A. Initialization

Initially, a predefined number of solutions is generated using the variable bounds. Any design of experiment (DOE) method such as, random, full-factorial or Latin hypercube design can be used to generate the set of initial solutions. In this study we adopted Latin Hypercube Sampling (LHS) to observe the performance of the algorithm.

B. Archive update

After generating the initial population, their actual objective and constraint responses are evaluated and stored in Archive. The Archive is periodically updated whenever a new infill sample is evaluated.

C. Constructing/updating GP models

All unique solutions from the Archive are used to build the GP models. Such models are built for each objective and each constraint function.

D. Identifying infill sample

Such an identification is dependent on the history of solutions evaluated so far as listed below.

- If there is no feasible solution in the Archive, only PF is maximized till a feasible solution is obtained. In this process, initially the algorithm attempts to identify a feasible solution and in the process updates the approximation of the constraint functions.
- Whenever a feasible solution is obtained, it's objective response value is used to calculate the EI (for multiple feasible solutions, the best feasible response is used instead). Later, the EI function is penalized by the PF and the penalized EI function is maximized to identify the infill sample.

E. Updating best solution

The best solution is updated whenever the new infill solution is better than the best in the Archive.

V. Numerical Experiments

In this section, we study the performance of the proposed approach and compare it with the approach suggested in [10] and [27], [20] for a range of constrained optimization problems collected from literature. The details of the problems are provided below for completeness.

A. Test problems

In Table I, the properties of the test problems are listed i.e. number of constraints, the optimum location, $\boldsymbol{x}^*$, optimum objective function value, f^* and the percentage of the feasible space, ρ. The computation of ρ is based on $1e^6$ randomly generated solutions.

B. Experimental Settings

It is important to take note that the studies reported in the literature use different termination conditions. To make a fair comparison, we have also used the same termination conditions as used in the corresponding reports. In all our cases, a real coded elitist evolutionary algorithm with simulated binary crossover and polynomial mutation [29] was used as the underlying optimizer for the maximization of the PF and the EI functions since they are known to be highly multimodal functions. For the evolution process, the probability of crossover and mutation were set to 0.9 and 0.1 respectively, whereas the distribution index of crossover and mutation were set to 20 and 30 respectively. A population of 100 individuals was evolved over 100 generations in each of the above listed maximization exercises.

1) Settings for comparison with [10]:

- Initial sample size was set to $11D-1$, where D represents the number of variables of the problem.
- The process was terminated if either $\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq 1e^{-2}$, ($\boldsymbol{x}^*$ is the true optimum solution) or the maximum number of function evaluations exceeded $D \times 100$.
- Number of independent runs was set to 20.

2) Settings for comparison with [20]:

- Initial sample size was set to 10.
- The process was terminated when either the best obtained feasible solution was within a box with a size $\pm 1\%$ *of SD* centered around the true optimum location. This condition was named as $x_{1\%}$ metric or maximum number of function calls exceeded $D \times 50$.
- Number of independent runs was set to 100.

3) Settings for comparison with [27]:

- Initial sample size was set to $11D - 1$.
- The process terminated had either of the following conditions met– $a)$ $x_{1\%}$ metric, $b)$ $f_{1\%}$ metric ($f_{1\%} = 1\%$ of the optimum objective value) or $c)$ maximum number of function calls surpassed $D \times 50$.
- Number of independent runs was set to 100.

C. Experimental Results

1) Comparison with [10]: Table II presents a comparison of results obtained using the proposed algorithm and [10] based on 20 independent runs. The performance is judged based on distance of the solution from the true optimum (Euclidean distance in the variable space), difference between the obtained and the true optimum (in objective space) and the number of function evaluations. Out of 10 problems studied, based on the mean values, the proposed algorithm delivers better results in 5 problems across all metrics, better in 7 problems based on the x error metric, better in 8 problems based on the f error metric and better in 6 problems based on the number of function evaluations.

2) Comparison with [20]: Table III presents a comparison of results obtained using the proposed algorithm and superEGO 1 and 2 from [20], Simulated Annealing (SA), Sequential Quadratic Programming (SQP) and DIRECT methods from [30]. In both the problems, the results obtained by our proposed algorithm is better than the reported results.

3) Comparison with [27]: Table IV presents a comparison of results obtained using the proposed algorithm and variants of algorithms reported in [27]. There were two termination criteria used in the study. For the first study, our proposed algorithm offered better results than all reported algorithm variants in [27] for both the problems. For the next termination criterion, our proposed algorithm was better than 4 variants and marginally worse than 3 variants for the first problems, and better than 2 variants and competitive with other 5 variants of the algorithm for the second problem.

VI. Conclusion and Future Research Directions

In this paper, we have critically evaluated various constrained EGO formulations. While a number of approaches have been proposed to deal with constrained optimization problems, it is difficult to objectively assess the performance, since each study compared performance on a small set of selected problems. The approaches incorporated better control of global and local exploration strategies, infill and boundary sampling strategies and enhancements to the probability of feasibility functions. Such modifications come with additional parameters and the fundamental question remains *are such algorithms significantly better than a simple constrained single objective EGO* for a range of optimization problems?. To answer this question, a simple single objective constrained EGO was designed which attempts to locate infill locations that maximize the probability of feasibility (until a feasible solution is identified) and then switches to penalized EI function (EI penalized using the probability of feasibility) [10] in the next stage. The effect of using different reference objective function values in the EI function is also highlighted in the study.

The results in the paper clearly support the view that a simple constrained single objective EGO offers competitive and often better results than several existing constrained EGO algorithms. We have used the same set of test functions that have been regularly used by other constrained EGO studies.

TABLE I: Number of constraints, optimum location, optimum objective function value and feasibility ratio (ρ) for the test problems.

Problem Name	**Num. of constraints**	$\boldsymbol{x}^*$	f^*	ρ
SHCBc [21]	1	[1.8150,-0.8750]	-2.9501	20.03%
Newbranin (g≤5) [20]	1	[3.2730,0.0489]	-268.7879	8.47%
Newbranin (g≤2) [20]	1	[3.2143,0.9633]	-243.0747	3.09%
G-Pc [9]	2	[0.5955,-0.4045]	5.6692	42.82%
Gomez#3 [20]	1	[0.1093,-0.6234]	-0.9711	18.49%
Sasena [9]	3	[0.2017,0.8332]	-0.7483	17.71%
Mystry [9]	1	[2.7450,2.3523]	-1.1743	48.30%
Reverse Mystry [9]	1	[3.0716,2.0961]	-0.5504	4.25%
Haupt-Schewefel [21]	1	[6.0860,6.0860; -6.0860,6.0860; 6.0860,-6.0860]	-13.6478	50.05%
HS100 [28]	1	[2.3305, 1.9514, -0.4775, ... 4.3657, -0.6245, 1.0381, 1.5942]	680.6283	74.65%
QCP4 [21]	3	[0.5,0,3]	-4	7.97%

TABLE II: Comparative performance analysis table of algorithms with constraint handling mechanism of Forrester et. al. [10] and EGOcons.

Problem	**Algorithms**	**Error_X** ($\|\boldsymbol{x}-\boldsymbol{x}^*\|$)		**Error_F** ($\|f-f^*\|$)		**Num. of FEs**	
		Mean	**Max**	**Mean**	**Max**	**Mean**	**Max**
SHCBc	Forrester et. al. [10]	0.0716	0.1445	0.0489	0.0499	210.6	**220**
	EGOcons	**0.0055**	**0.0113**	**0.0005**	**0.0026**	**45.9**	**220**
Newbranin	Forrester et. al. [10]	4.7939	16.0690	178.0722	181.2121	220.0	**220**
	EGOcons	**0.0332**	**0.1023**	**0.1333**	**0.4082**	**192.45**	**220**
G-Pc	Forrester et. al. [10]	0.2133	0.5613	3.0597	4.4726	220.0	220
	EGOcons	**0.0036**	**0.0090**	**0.0124**	**0.0254**	**56**	**120**
Gomez#3	Forrester et. al. [10]	0.3881	1.2873	0.0539	0.0600	210.3	220
	EGOcons	**0.0051**	**0.0098**	**0.0029**	**0.0126**	**34.3**	**46**
Sasena	Forrester et. al. [10]	0.3400	0.7150	0.5017	0.5017	220.0	220
	EGOcons	**0.0049**	**0.0092**	**0.0067**	**0.0122**	**29.8**	**41**
Mystery	Forrester et. al. [10]	**0.0054**	**0.0155**	0.0074	0.1020	78.6	**220**
	EGOcons	0.0055	**0.0155**	**0.0040**	**0.0162**	**63.4**	**220**
Reverse Mystery	Forrester et. al. [10]	0.5401	1.2289	0.4491	0.4496	**220.0**	**220**
	EGOcons	**0.2193**	**0.2228**	**0.2241**	**0.2252**	**220.0**	**220**
Haupt-Schewefel (mod.)	Forrester et. al. [10]	**0.0067**	**0.0197**	**0.0000**	**0.0002**	**66.9**	**220**
	EGOcons	0.0078	**0.0197**	**0.0000**	**0.0002**	76.35	**220**
HS100	Forrester et. al. [10]	**4.8732**	**7.7891**	**100.8683**	**267.4228**	**775.0**	**775**
	EGOcons	5.2258	7.7891	294.3924	504.8108	**775.0**	**775**
QCP4	Forrester et. al. [10]	1.8471	2.6208	2.0319	2.4964	**331.0**	**331**
	EGOcons	**1.8025**	**1.8030**	**1.9996**	**2.0000**	**331.0**	**331**

TABLE III: Comparison of the number of function calls made to satisfy $x_{1\%}$ termination condition using Simulated Annealing (SA) [30], Sequential Quadratic Programming (SQP) [30], DIRECT [30], superEGO 1 and 2 [20] and our proposed approach EGOcons.

Problem	**SA**	**SQP**	**DIRECT**	**sup.EGO 1**	**sup.EGO 2**	**EGOcons**
Newbranin	5371	363	76	22.2	22	**21.75**
Gomez#3	7150	831	93	66.3	36.5	**26.66**

TABLE IV: Comparison of the mean number of function calls made to satisfy $\boldsymbol{x}_{1\%}$ termination condition among EI, GEI with g values 2, 5, 10, Proposed method of Sasena et. al. [27] with Watson and Barnes (WB) criterion 1, 2, 3 and our proposed method EGOcons. The approaches which were unable to meet the termination condition within the maximum number of function evaluations were marked with " – "

Problem	**EI**	**GEI** (g=2)	**GEI** (g=5)	**GEI** (g=10)	**WB** (1)	**WB** (2)	**WB** (3)	**EGOcons**
Mystery	-	-	-	-	91	29	-	**27.49**
Gomez#3	62	71	93	-	78	60	-	**29.87**

TABLE V: Comparison of the mean number of function calls made to satisfy $f_{1\%}$ termination condition among EI, GEI with g values 2, 5, 10, Proposed method of Sasena et. al. [27] with Watson and Barnes (WB) criterion 1, 2, 3 and our proposed method EGOcons.

Problem	**EI**	**GEI** (g=2)	**GEI** (g=5)	**GEI** (g=10)	**WB** (1)	**WB** (2)	**WB** (3)	**EGOcons**
Mystery	**35**	-	-	36	88	28	-	42.97
Gomez#3	26	**25**	**25**	23	37	**25**	-	31.38

Since most of the studies deal with limited number of variables (upto 7), it is of interest to evaluate performance of EGOs and GP models in general for problems with higher dimension. Besides, better means to control boundary sampling at later stages of the search, development of efficient schemes to deal with correlated constraints and extensions to deal with multi-point sampling (batch) are some of the other directions currently being pursued by us.

ACKNOWLEDGMENT

The second author would like to acknowledge the support from the Ian Potter Foundation Travel Grant 2016 and the last author would like to acknowledge the support from ARC Discover Grant (DP130103850).

REFERENCES

[1] N. Cressie, "The origins of Kriging," *Mathematical Geology*, vol. 22, no. 3, pp. 239–252, 1990.

[2] M. Riedmiller, "Advanced supervised learning in multi-layer perceptronsfrom backpropagation to adaptive learning algorithms," *Computer Standards & Interfaces*, vol. 16, no. 3, pp. 265–278, 1994.

[3] G. E. P. Box and N. R. Draper, *"Empirical model-building and response surfaces"*. Wiley New York, 1987, vol. 424.

[4] A. Keane, "Design search and optimisation using radial basis functions with regression capabilities," in *Adaptive Computing in Design and Manufacture VI*. Springer, 2004, pp. 39–49.

[5] S. M. Clarke, J. H. Griebsch, and T. W. Simpson, "Analysis of support vector regression for approximation of complex engineering analyses," *Journal of Mechanical Design*, vol. 127, no. 6, pp. 1077–1087, 2005.

[6] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.

[7] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[8] M. Schonlau, W. J. Welch, and D. R. Jones, "Global versus local search in constrained optimization of computer models," *Lecture Notes-Monograph Series*, pp. 11–25, 1998.

[9] M. J. Sasena, "Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations," Ph.D. dissertation, General Motors, 2002.

[10] A. Forrester, A. Sobester, and A. Keane, *"Engineering design via surrogate modelling: a practical guide"*. John Wiley & Sons, 2008.

[11] A. Basudhar, C. Dribusch, S. Lacaze, and S. Missoum, "Constrained efficient global optimization with support vector machines," *Structural and Multidisciplinary Optimization*, vol. 46, no. 2, pp. 201–221, 2012.

[12] C. Audet, A. J. Booker, J. Dennis Jr, P. D. Frank, and D. W. Moore, "A surrogate-model-based method for constrained optimization," *AIAA paper*, vol. 4891, 2000.

[13] J. Parr, A. Keane, A. I. Forrester, and C. Holden, "Infill sampling criteria for surrogate-based optimization with constraint handling," *Engineering Optimization*, vol. 44, no. 10, pp. 1147–1166, 2012.

[14] C. Lee, J. Ahn, H. Bae, and J. Kwon, "Efficient global optimization incorporating feasibility criterion for the design of a supersonic inlet," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, p. 0954410011424974, 2011.

[15] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[16] J. Parr, A. I. Forrester, and A. J. Keane, "Enhancing infill sampling criteria for surrogate-based constrained optimization," *Journal of Computational Methods in Sciences and Engineering*, vol. 12, no. 1-2, pp. 25–45, 2012.

[17] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review." in *IEEE congress on evolutionary computation*, 2008, pp. 2419–2426.

[18] A. G. Watson and R. J. Barnes, "Infill sampling criteria to locate extremes," *Mathematical Geology*, vol. 27, no. 5, pp. 589–608, 1995.

[19] M. J. Sasena, P. Papalambros, and P. Goovaerts, "Exploration of metamodeling sampling criteria for constrained global optimization," *Engineering Optimization*, vol. 34, no. 3, pp. 263–278, 2002.

[20] M. Sasena, P. Papalambros, and P. Goovaerts, "Global optimization of problems with disconnected feasible regions via surrogate modeling," in *Proceedings of 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, Paper No. AIAA-2002-5573*, 2002.

[21] S. G. Cho, J. Jang, J. Kim, M. Lee, J.-S. Choi, S. Hong, and T. H. Lee, "Statistical surrogate model based sampling criterion for stochastic global optimization of problems with constraints," *Journal of Mechanical Science and Technology*, vol. 29, no. 4, pp. 1421–1427, 2015.

[22] J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn *et al.*, "Design and analysis of computer experiments," *Statistical Science*, vol. 4, no. 4, pp. 409–423, 1989.

[23] A. I. J. Forrester, A. J. Keane, and N. W. Bressloff, "Design and analysis of 'noisy' computer experiments," *AIAA Journal*, vol. 44, no. 10, pp. 2331–2339, 2006.

[24] D. J. Toal, N. Bressloff, A. Keane, and C. Holden, "The development of a hybridized particle swarm for kriging hyperparameter tuning," *Engineering Optimization*, vol. 43, no. 6, pp. 675–699, 2011.

[25] Z. Feng, Q. Zhang, Q. Zhang, Q. Tang, T. Yang, and Y. Ma, "A multiobjective optimization based framework to balance the global exploration and local exploitation in expensive optimization," *Journal of Global Optimization*, vol. 61, no. 4, pp. 677–694, 2015.

[26] C. Chevalier, D. Ginsbourger, J. Bect, E. Vazquez, V. Picheny, and Y. Richet, "Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set," *Technometrics*, vol. 56, no. 4, 2014.

[27] M. J. Sasena, P. Y. Papalambros, and P. Goovaerts, "Metamodeling sampling criteria in a global optimization framework," in *Proceedings of 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Optimization, Long Beach, CA*, 2000.

[28] F. Boukouvala and M. G. Ierapetritou, "Derivative-free optimization for expensive constrained problems using a novel expected improvement objective function," *AIChE Journal*, vol. 60, no. 7, pp. 2462–2474, 2014.

[29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[30] D. R. Jones, "Direct global optimization algorithmdirect global optimization algorithm," in *Encyclopedia of Optimization*. Springer, 2001, pp. 431–440.