

Zhang, Shiqiang; Lee, Robert M.; Shafei, Behrang; Walz, David; Misener, Ruth

Article — Published Version

Dependence in constrained Bayesian optimization

Optimization Letters

Provided in Cooperation with:

Springer Nature

Suggested Citation: Zhang, Shiqiang; Lee, Robert M.; Shafei, Behrang; Walz, David; Misener, Ruth (2023) : Dependence in constrained Bayesian optimization, Optimization Letters, ISSN 1862-4480, Springer, Berlin, Heidelberg, Vol. 18, Iss. 6, pp. 1457-1473, <https://doi.org/10.1007/s11590-023-02047-z>

This Version is available at:

<https://hdl.handle.net/10419/318124>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by/4.0/>



Dependence in constrained Bayesian optimization

When do we need it and how does it help?

Shiqiang Zhang¹ · Robert M. Lee² · Behrang Shafei² · David Walz² · Ruth Misener¹

Received: 18 November 2022 / Accepted: 11 July 2023 / Published online: 20 September 2023
© The Author(s) 2023, corrected publication 2023

Abstract

Constrained Bayesian optimization optimizes a black-box objective function subject to black-box constraints. For simplicity, most existing works assume that multiple constraints are independent. To ask, *when and how does dependence between constraints help?*, we remove this assumption and implement probability of feasibility with dependence (Dep-PoF) by applying multiple output Gaussian processes (MOGPs) as surrogate models and using expectation propagation to approximate the probabilities. We compare Dep-PoF and the independent version PoF. We propose two new acquisition functions incorporating Dep-PoF and test them on synthetic and practical benchmarks. Our results are largely negative: incorporating dependence between the constraints does not help much. Empirically, incorporating dependence between constraints may be useful if: (i) the solution is on the boundary of the feasible region(s) or (ii) the feasible set is very small. When these conditions are satisfied, the predictive covariance matrix from the MOGP may be poorly approximated by a diagonal matrix and the off-diagonal matrix elements may become important. Dep-PoF may apply to settings where (i) the constraints and their dependence are totally unknown and (ii) experiments are so expensive that any slightly better Bayesian optimization procedure is preferred. But, in most cases, Dep-PoF is indistinguishable from PoF.

Keywords Bayesian Optimization · Multiple output Gaussian processes · Dependence · Expectation propagation

1 Introduction

Bayesian Optimization (BO) [1–3] optimizes an expensive-to-evaluate black-box function where derivative information cannot be obtained. BO uses a surrogate model, e.g. Gaussian Processes (GP) [4], to learn the response surface and designs an acquisition function to trade-off exploration vs. exploitation. BO maximizes the acquisition function at each iteration to locate the next sample. Classic BO studies unconstrained problems, but practical problems may involve constraints, e.g., a machine learning model may yield poor results for certain hyper-parameter combinations [5]. In chemistry, ingredient concentrations and environmental variables should lie inside safe intervals and/or satisfy extra constraints [6–8]. If these constraints are known a priori, they can be handled in the acquisition function [9–12]. Otherwise, constrained BO (cBO) can handle black-box constraints.

A popular treatment for cBO is using surrogate models to learn constraints and constructing an acquisition function that could lead to feasible region(s). Constrained Expected Improvement (cEI) [13], a widely-used acquisition function, multiplies Expected Improvement (EI) [14] and the Probability of Feasibility (PoF), i.e., the probability of a point satisfying all constraints. Variants of cEI may use alternatives to EI and/or different ways to calculate PoF. For example, a classic way to compute PoF is training a classifier [11, 15–20]. Training this classifier allows one to handle multiple constraints with binary or continuous outputs since the only information needed to train a classification model is whether a point is feasible or not. Alternatively, Boukouvala and Ierapetritou [9] applied a regression model to predict their proposed feasibility function. These type of approaches do not require prior knowledge and is flexible to types of constraints. As the number and complexity of constraints increases, however, a single model is insufficient to learn the feasible region(s), which could be multiple-piece, non-convex, and very small. One solution is equipping each constraint with a surrogate [5, 21–25]. Some papers propose acquisition functions beyond cEI [8, 22, 24]. By incorporating constraints into the objective, some works transform constrained problems to unconstrained ones [23, 25, 26].

Table 1 compares the aforementioned methods: note that combining an acquisition function from BO with PoF is common for cBO. For multiple models, most research assumes independence, which brings simplicity but loses correlations between constraints. The first cEI paper justified the independence assumption as *the covariance between predictions for pairs of responses are not so tractable* [13]. Other papers also mention dependence [8, 21, 25, 26]. Dependence certainly has merit, e.g. Multi-task BO [31] gains information about expensive tasks from cheap tasks by capturing the correlations between tasks. When Gelbart et al. [5] considered decoupled cases, in which scenario the objective and multiple constraints could be evaluated independently, they regarded all these functions as tasks and solved it using Multi-task BO. Our setting is different than Gelbart et al. [5] in two ways: we remove their independence assumption and assume that the objective and all the constraints are evaluated simultaneously. We ask: *What happens if we do not assume independence between multiple surrogate models?*

Table 1 Different methods in cBO

Method	$\alpha(\mathbf{x})$	PoF	Constraints	Status	Dependent	Models
[15]	IECI	✓	$p \geq 1, q = 0$	Unknown	★	GPC
[16]	cEI	✓	$p = 1, q = 0$	Unknown	★	RF
[17]	cEI	✓	$p \geq 1, q = 0$	Unknown	★	SVM
[18]	cEI	✓	$p \geq 1, q = 0$	Unknown	★	LS-SVM
[11]	cEI	✓	$p \geq 1, q = 0$	Both	★	GPC
[9]	FEI	×	$p \geq 1, q = 0$	Both	★	GPR
[5]	cEI	✓	$p \geq 1, q = 0$	Unknown	Independent	GPR $\times p$
[21]	cEI	✓	$p \geq 1, q = 0$	Unknown	Independent	GPR $\times p$
[26]	EI	×	$p \geq 1, q = 0$	Unknown	Independent	GPR $\times p$
[23]	EI	×	$p \geq 1, q \geq 1$	Unknown	Independent	GPR $\times (p + q)$
[25]	EI	×	$p \geq 1, q = 0$	Unknown	Independent	GPR $\times p$
[22]	EV	×	$p \geq 1, q = 0$	Unknown	Independent	GPR $\times p$
[24]	PES	×	$p \geq 1, q = 0$	Unknown	Independent	GPR $\times p$
[8]	cAS	✓	$p = 1, q = 1$	Unknown	★	GPR

IECI Integrated expected conditional improvement, *FEI* feasibility expected improvement, *EV* expect volume, *PES* predictive entropy search [27], *cAS* constrained adaptive sampling, *GPC* GP classification model [4], *RF* random forest [28], *SVM* support vector machine [29], *LS-SVM* least-squares SVM [30], *GPR* GP regression model [4]

$\alpha(\mathbf{x})$: acquisition function. PoF: whether PoF is used (✓) or not (×). Constraints: number of inequality (p) and equality (q) constraints. Status: whether the constraints are known. Dependent: when multiple GPs are used, whether assuming they are independent (‘★’ for a single model). Models: the choice and number of models for constraints

To remove the widely-used independence assumption in cBO, we implement Multiple Output Gaussian Processes (MOGPs) [32] to capture the correlations between constraints, and use Expectation Propagation for Multivariate Gaussian Probabilities (EPMGP) [33] to calculate PoF. Our modified, dependent PoF, or ‘Dep-PoF’, replaces the commonly-used PoF in any acquisition function without disturbing other parts in classical cBO. To verify its effectiveness, we substitute Dep-PoF into two acquisition functions and test them on a variety of case studies.

Paper structure. Section 2 introduces the background. Section 3 defines Dep-PoF and the acquisition functions and then provides convergence analysis of Dep-PoF and complexity analysis. Section 4 shows the experimental results. We further discuss Dep-PoF in Sect. 5 and conclude in Sect. 6.

Supplementary materials. Open Resource 1 gives implementation details for all methods and models involved in this paper. Open Resource 2 includes more experimental results for model consideration. Open Resource 3 provides detailed information for all benchmarks used in this paper.

2 Background

2.1 Multiple output Gaussian processes (MOGPs)

A Gaussian Process (GP) generalizes the Gaussian probability distribution to infinite variables, any finite subset of which owns a joint Gaussian distribution [4]. Mathematically, a GP is a distribution $f(\cdot) \sim \mathcal{GP}(\mu(\cdot), k(\cdot, \cdot))$ over a scalar function $f : \mathcal{X} \rightarrow \mathbb{R}$ on input domain \mathcal{X} . All properties of a GP is fully specified by its mean function $\mu(\cdot)$ and covariance function, or kernel $k(\cdot, \cdot)$. Conditioned on an observed dataset $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^n$, a GP computes the predictive mean and covariance for any finite number of query points in closed form. GPs are powerful tools for scalar functions. For a multi-output, or vector-valued function, a naive and common-used treatment is using equal number of GPs to separately model each output. The correlations between related outputs are thereby ignored. Multiple Output Gaussian Processes (MOGPs) [32] can represent the output correlations.

Consider a multi-output function $\mathbf{f}(\cdot) : \mathcal{X} \rightarrow \mathbb{R}^p$ with input domain $\mathcal{X} = \mathbb{R}^d$. A prior over this function is a MOGP if the distribution of any vector with the form $\{f_{j_i}(\mathbf{x}_i)\}_{i=1}^n \in \mathbb{R}^n$ is Gaussian distributed, where $f_{j_i}(\mathbf{x}_i)$ denotes the j_i th output at the i th input \mathbf{x}_i . Like a GP, a MOGP is determined by its mean function and kernel (W.l.o.g., assume zero mean functions). For homotopic data, where we could observe all output dimensions for each input, MOGP kernels can be viewed as matrix-valued [34, 35]:

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^{p \times p} \\ (\mathbf{x}, \mathbf{x}') \mapsto k(\mathbf{x}, \mathbf{x}') = \mathbb{C}_{f(\cdot)}[\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}')].$$

For heterotopic data from where only a subset of outputs is observed each time, a common treatment is including the indexes of outputs in input space, i.e., 'output as input' [32, 35, 36]. In this paper, we consider homotopic data and inherit relevant notations of MOGP from [35] with corresponding implementation in GPflow [37].

2.2 Expectation propagation for multivariate Gaussian probabilities (EPMGP)

Gaussian cumulative probabilities are difficult to compute for dimensions greater than one. Numerical integration can achieve an accurate approximation but is computationally expensive [38]. Instead, we use Expectation Propagation for Multivariate Gaussian Probabilities (EPMGP) [33] to calculate PoF. Given a mean vector $\mathbf{m} \in \mathbb{R}^n$ and a covariance matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$, a multivariate Gaussian distribution $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{K})$, $\mathbf{x} \in \mathbb{R}^n$ is defined as:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{K}|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m})^T \mathbf{K}^{-1} (\mathbf{x} - \mathbf{m}) \right\}.$$

We are interested in the probability that a sample from $p(\mathbf{x})$ falls in a region $\mathcal{A} \subset \mathbb{R}^n$, that is:

$$F(\mathcal{A}) = \int_{\mathcal{A}} p(\mathbf{x}) d\mathbf{x} = \int_{l_1(\mathbf{x})}^{u_1(\mathbf{x})} \cdots \int_{l_n(\mathbf{x})}^{u_n(\mathbf{x})} p(\mathbf{x}) d\mathbf{x}_n \cdots d\mathbf{x}_1,$$

where l_i and u_i denote the lower and upper bounds of \mathcal{A} . Cunningham et al. [33] analytically approximate $F(\mathcal{A})$ using Expectation Propagation [39]. Their EPMGP algorithm approximates accurately for hyperrectangular \mathcal{A} , which admits its application to most common cases including computing cumulative density function (cdf).

2.3 The acquisition function

2.3.1 Constrained expected improvement (cEI)

cEI, which incorporates constraint uncertainty into EI [14], holds many names, e.g., Integrated Expected Conditional Improvement [15], Constraint-Weighted EI [5], and Expected Constrained Improvement [21]. cEI is defined by:

$$cEI(\mathbf{x}) = EI(\mathbf{x})PoF(\mathbf{x}),$$

where $PoF(\mathbf{x})$ is the probability of \mathbf{x} satisfying all constraints, and the target of EI is the optimal objective value for observed feasible points so far. Note cEI is not well-defined if all current samples are infeasible. In this case, the EI term is not considered until a feasible point is sampled. Gelbart [40] fully analyzes cEI.

2.3.2 Constrained adaptive sampling (cAS)

Acquisition functions using PoF commonly have a product form, about which we concern: (i) the probability term and improvement term directly influence each other, and (ii) the probability term has little exploration nature. Therefore, we consider constrained Adaptive Sampling (cAS) [8] with a sum form and compare its performance with cEI. cAS is a weighted sum of three terms [8]: (i) a border-find term, (ii) a PoF term for a single constraint (the authors do not consider multiple constraints), (iii) an exploration term. Open Resource 1 describes cAS.

Remark There are many state-of-the-art acquisition functions, for example those in Table 1. Our idea is to compare PoF versus Dep-PoF, so we chose acquisition functions cEI and cAS because they directly incorporate PoF. Most acquisition functions, including IECI [15], EV [22], and PESC [24], incorporate the assumption of independence between the constraints. But the independence assumption in other methods is used in more sophisticated ways than cEI and cAS, for example approximating the acquisition function in PESC and integrating in IECI and EV. Initially, we considered a comparison incorporating additional acquisition functions. However, Eriksson and Poloczek [41] have already provided a detailed comparison between state-of-the-art methods including cEI and PESC. Our cEI approach may be compared, for instance, to the cEI line in [41].

3 Method

3.1 The model

We optimize a black-box objective with multiple ($p > 1$) black-box constraints:

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}) \quad \text{s.t. } c_i(\mathbf{x}) \leq 0, 1 \leq i \leq p,$$

over a compact set $\Omega \subset \mathbb{R}^d$ where $f, c_i : \Omega \rightarrow \mathbb{R}, 1 \leq i \leq p$. For any input $\mathbf{x} \in \Omega$, we can evaluate the objective and constraints with/without noise, but cannot get gradient information. Each black-box function evaluation is expensive. Under the independence assumption, we observe an i.i.d. $(p+1)$ -dimensional vector $(y_0(\mathbf{x}), y_1(\mathbf{x}), \dots, y_p(\mathbf{x}))$ given by $y_0(\mathbf{x}) \sim \mathcal{N}(f(\mathbf{x}), \sigma_0^2(\mathbf{x}))$ and $y_i(\mathbf{x}) \sim \mathcal{N}(c_i(\mathbf{x}), \sigma_i^2(\mathbf{x}))$, $i \in \{1, \dots, p\}$. Without assuming independence between constraints, the observed vector is $y_0(\mathbf{x}) \sim \mathcal{N}(f(\mathbf{x}), \sigma_0^2(\mathbf{x}))$ and $(y_1(\mathbf{x}), \dots, y_p(\mathbf{x})) \sim \mathcal{N}(\mathbf{m}(\mathbf{x}), \mathbf{K}(\mathbf{x}))$ with $\mathbf{m}(\mathbf{x}) = (c_1(\mathbf{x}), \dots, c_p(\mathbf{x}))$. The diagonal elements in $\mathbf{K}(\mathbf{x})$ are $\sigma_1^2(\mathbf{x}), \dots, \sigma_p^2(\mathbf{x})$, so we ask: *When are off-diagonal $\mathbf{K}(\mathbf{x})$ elements important?*

In practice, the input domain Ω is rescaled to $[0, 1]^d$. Under independence assumption, we train $p+1$ GPs for the $p+1$ black-boxes. Otherwise, we train one GP for objective and one MOGP for constraints. For noise-free cases, we tend to find a feasible point with minimal objective value. For noisy cases, one can relax feasible region(s) with tolerance ϵ_i and confidence $\delta_i, 1 \leq i \leq p$ [5, 24]:

$$\{\mathbf{x} \in \Omega \mid P(y_1(\mathbf{x}) \leq \epsilon_1) \geq 1 - \delta_1, \dots, P(y_p(\mathbf{x}) \leq \epsilon_p) \geq 1 - \delta_p\}. \quad (1)$$

3.2 Probability of feasibility with dependence (Dep-PoF)

By independence/dependence in the paper we refer to the prior distributions y_i over constraints c_i instead of constraints themselves. Before discussing PoF with/without dependence, we clarify the concept of PoF. If we could easily evaluate the constraints at \mathbf{x} , PoF reduces to a binary value:

$$PoF(\mathbf{x}) = I(c_1(\mathbf{x}) \leq 0, \dots, c_p(\mathbf{x}) \leq 0).$$

Here, there is no need to consider dependence or non-trivial probability as:

$$PoF(\mathbf{x}) = I(c_1(\mathbf{x}) \leq 0) \cdots I(c_p(\mathbf{x}) \leq 0),$$

where $I(\cdot)$ is indicator function. However, since c_i are black-boxes, we cannot afford checking the feasibility for any queried input. PoF cheaply approximates the constraint feasibility. Mathematically, Dep-PoF is defined:

$$PoF_{\text{dep}}(\mathbf{x}) = P(y_1(\mathbf{x}) \leq 0, \dots, y_p(\mathbf{x}) \leq 0).$$

With the independence assumption, PoF is the product of univariate Gaussian distributions' cdf:

$$PoF(\mathbf{x}) = \prod_{i=1}^p P(y_i(\mathbf{x}) \leq 0).$$

Otherwise, calculating Dep-PoF involves the cdf for the multivariate Gaussian distribution $(y_1(\mathbf{x}), \dots, y_p(\mathbf{x})) \sim \mathcal{N}(\mathbf{m}(\mathbf{x}), \mathbf{K}(\mathbf{x}))$, which is complicated. This partly explains why most works assume independence. EPMGP (see Sect. 2.2) approximates Dep-PoF ($l_i = -\infty, u_i = 0$) with an acceptable additional computational cost, and allows us to compute Dep-PoF and test whether dependence helps.

3.3 Constrained acquisition function

3.3.1 Constrained expected improvement with dependence (Dep-cEI)

Implementing Dep-PoF for cEI is straightforward. Replacing the PoF term in cEI with Dep-PoF, we define Dep-cEI as:

$$cEI_{\text{dep}}(\mathbf{x}) = EI(\mathbf{x})PoF_{\text{dep}}(\mathbf{x}),$$

where $EI(\mathbf{x})$ is calculated by the predicted mean and variance from the trained GP model, $PoF_{\text{dep}}(\mathbf{x})$ is computed by using EPMGP algorithm given the predicted mean and covariance matrix from the trained MOGP model.

3.3.2 Constrained adaptive sampling with dependence (Dep-cAS)

This section implements cAS to our problem setting. To optimize our continuous objective function, we replace the Ludl et al. [8] border-find term with the optimization term in Adaptive Sampling (AS) strategy proposed by [42]. We still call our modified version 'cAS', which consists of: (i) optimization, (ii) constraint, and (iii) exploration terms. Dep-cAS replaces the PoF in constraint term with Dep-PoF. Open Resource 1 gives details of cAS and Dep-cAS.

Note that Dep-cAS adds extra parameters, which is not preferred in machine learning even with the discussion of selecting these parameters [8] and the flexibility to different focus brought by weight parameters. The reason this paper introduces cAS and modifies it into Dep-cAS is neither to recommend nor criticize this acquisition function. By considering Dep-PoF in both cEI and cAS, we want to check if dependence helps in different settings from different fields.

3.4 Convergence analysis of Dep-PoF

The section is motivated by experimental results, where we found Dep-PoF and PoF share similar performance even in cases that the covariance matrix $\mathbf{K}(\mathbf{x})$ has non-zero non-diagonal elements (which means that dependence exists). The

following analysis only involves constraints and corresponding surrogate models as they are the only differences between Dep-PoF and PoF.

Intuitively, when we have sufficient samples and suitable hyper-parameters, GPs/MOGP well-represent the underlying functions. Mathematically, the mean $y_i(\mathbf{x})$ converges to true function $c_i(\mathbf{x})$ and the covariance $(\sigma_i^2(\mathbf{x}))$ for GPs, $\mathbf{K}(\mathbf{x})$ for MOGP) converges to 0 as the number of samples $n \rightarrow \infty$. Teckentrup [43] proved the convergence of GP with estimated hyper-parameters and provided explicit bounds for mean and covariance.

We analyze the performance of PoF and Dep-PoF in an ideal setting: the predictive mean functions match our constraints, i.e. $\mathbf{m}(\mathbf{x}) = (c_1(\mathbf{x}), \dots, c_p(\mathbf{x}))$. In this way, we can exclude the influence of mean and focus on the covariance matrix, which is the key difference between PoF and Dep-PoF. Numerically, we report the predictive accuracy for both models in Open Resource 2 to support this setting. Under this setting, Proposition 1 analyzes the convergence of Dep-PoF to PoF on non-boundary domain. Proposition 2 shows a 'decay' behaviour of PoF on the boundary.

Proposition 1 (Non-boundary domain) *For any non-boundary point \mathbf{x} , we have $PoF_{\text{dep}}(\mathbf{x}) \rightarrow PoF(\mathbf{x})$, as $n \rightarrow \infty$.*

Proof For a non-boundary feasible point \mathbf{x} , we have $c_i(\mathbf{x}) < 0$ for $1 \leq i \leq p$. Using the continuity of the cdf function of multivariate Gaussian distribution, for any $\delta > 0$, there exists $\epsilon > 0$ so that if $\|\mathbf{K}(\mathbf{x})\|_2 < \epsilon$,

$$PoF_{\text{dep}}(\mathbf{x}) = P(y_1(\mathbf{x}) \leq 0, \dots, y_p(\mathbf{x}) \leq 0) > 1 - \delta.$$

Recall that for any $\mathbf{x} \in \Omega$, we have $\|\mathbf{K}(\mathbf{x})\|_2 \rightarrow 0$, as $n \rightarrow \infty$, which means that there exists $N_\epsilon \in \mathbb{N}$ such that $\|\mathbf{K}(\mathbf{x})\|_2 < \epsilon$ as long as $n > N_\epsilon$, $\forall \mathbf{x} \in \Omega$.

Since $P(y_i(\mathbf{x}) \leq 0) \geq PoF_{\text{dep}}(\mathbf{x})$, $1 \leq i \leq p$, then $PoF(\mathbf{x}) > (1 - \delta)^p$. Both $PoF(\mathbf{x})$ and $PoF_{\text{dep}}(\mathbf{x})$ are bounded by $(1 - \delta)^p$ (since $1 - \delta \geq (1 - \delta)^p$) and 1. Therefore, we have

$$|PoF_{\text{dep}}(\mathbf{x}) - PoF(\mathbf{x})| < 1 - (1 - \delta)^p \rightarrow 0, \text{ as } \delta \rightarrow 0.$$

Same conclusion also holds for a non-boundary infeasible point \mathbf{x} . □

Proposition 2 (Boundary) *For any boundary point \mathbf{x} , denote its number of active constraints is q , then we have $PoF(\mathbf{x}) \rightarrow \frac{1}{2^q}$, as $n \rightarrow \infty$.*

Proof For a point \mathbf{x} on the boundary, without loss of generality, assume that the first q constraints are active, that is, $c_i(\mathbf{x}) = 0$ for $1 \leq i \leq q$ and $c_j(\mathbf{x}) < 0$ for $q < j \leq p$. Following the proof of Proposition 1, we can show that

$$P(y_{q+1}(\mathbf{x}) \leq 0, \dots, y_p(\mathbf{x}) \leq 0) \rightarrow 1, \text{ as } n \rightarrow \infty.$$

Since $\mathbf{m}(\mathbf{x}) = (c_1(\mathbf{x}), \dots, c_p(\mathbf{x}))$ and $c_i(\mathbf{x}) = 0$ for $1 \leq i \leq q$, we obtain that $P(y_i(\mathbf{x}) \leq 0) = \frac{1}{2}$ for $1 \leq i \leq q$. Thus, as $n \rightarrow \infty$, we have

$$PoF(\mathbf{x}) = \prod_{i=1}^q P(y_i(\mathbf{x}) \leq 0) \cdot P(y_{q+1}(\mathbf{x}) \leq 0, \dots, y_p(\mathbf{x}) \leq 0) \rightarrow \frac{1}{2^q}. \quad (2)$$

□

The 'decay' shown in (2) slows down the optimization process if the global optimum is a boundary point with multiple active constraints. A way to handle it is relaxing feasible regions (even for noise-free cases) as mentioned in (1), which turns the boundary optimum into an interior one. However, this relaxation requires more hyper-parameters and possibly finds an infeasible 'optimum'.

Dep-PoF also suffers from the decay of probability near boundary. When the dependence between constraints is slight, its decay is closed to (2). However, for some cases, dependence could relieve the decay and then speed up the process to find a boundary optimum. For instance, in Sect. 4.1 we construct a case with boundary optimum and repeated constraints. In such setting, Dep-PoF outperforms PoF since Dep-PoF is ideally equivalent to PoF for one constraint. Although this case seems to overestimate the importance of dependence, we regard it as an upper bound for the improvement that Dep-PoF may bring. Practically, the correlations between unknown constraints are also unknown, so as the location of optimum. If Dep-PoF shares similar performance to PoF for general cases but outperforms PoF for some special cases, then it still has merit.

3.5 Time complexity analysis

From a modeling perspective, the time complexity for training p independent GPs is $O(pn^3)$, while for training a MOGP is $O(p^3n^3)$, where n is the number of samples and p is the number of constraints. The complexity for MOGP could be reduced through approximated inference, e.g., by introducing (latent) inducing points [35]. This paper implements exact inference for both choices of model to eliminate their possibility of influencing the comparison between Dep-PoF and PoF. From a probability perspective, calculating PoF is $O(p)$ as computing the cdf for univariate Normal distribution is $O(1)$, while calculating Dep-PoF using EPMGP [33] requires $O(p^3 + mp^2)$, where m is the number of iterations within EPMGP. Obviously, the time complexity of EPMGP is incomparable with univariate case. For moderate number of constraints, nevertheless, EPMGP is efficient.

4 Experimental results

To consider the performance of Dep-PoF, we compare cEI, Dep-cEI, cAS, Dep-cAS, and Random Search for all benchmarks. All benchmarks are observed without noise. All methods begin with 10 initial points given by a Latin hypercube design and have a budget of 100 evaluations. The plots also show the optimal objective value for each benchmark. For any infeasible point, set its value to the maximal objective value among all evaluated feasible points. All experimental results are realized by using GPflow [37] and Scipy [44], and we report the mean with one

standard error over 50 replications. Open Resource 1 provides full implementation details and justifies using GPflow. In this section, d means the input dimension, and p means the number of constraints.

4.1 A toy example with repeated constraints

Although Sect. 3.2 discusses dependence between constraints, we do not know how to measure it for general cases. Therefore, to see the difference between Dep-PoF and PoF, we construct an extreme case based on the 2D Gardner [21] problem with 1 black-box constraint (see Open Resource 3 for details), where we repeat this single constraint several times. 2D Gardner problem itself is challenging due to two small feasible regions and two boundary optimum (one of which is a global optimum). After repeating its constraint p times, denote $p_0(\mathbf{x}) = P(c_1(\mathbf{x}) \leq 0)$, then $PoF_{\text{dep}}(\mathbf{x}) = p_0(\mathbf{x})$ while PoF becomes $PoF(\mathbf{x}) = p_0^p(\mathbf{x})$. Intuitively, as p increases, more samples are needed to increase the confidence for \mathbf{x} being feasible. Experimental results in Fig. 1 verify this intuition, from which we notice:

Dep-PoF helps exploiting feasible regions. When the samples are scarce, Dep-PoF improves the objective earlier since it calculates the true probabilities which admit it samples points more confidently. For PoF, however, since the probabilities decay near the boundary, it needs more samples to gain more information about feasible regions.

Dep-PoF performs better when p increases. Ideally, Dep-PoF for different p should be identical in this example. Their different performances seem to be counterintuitive. However, note that these (repeated) constraints will share information during training MOGP, which may explain better performance for larger p . This explanation is also supported by our experimental results in Open Resource 2.

4.2 Synthetic benchmarks

This section tests synthetic benchmarks from literature. Table 2 introduces these benchmarks and Fig. 2 diagrams part of our experimental results. Open Resource 3 gives full details about these problems and results for 2D benchmarks.

As shown in Fig. 2, cEI almost always outperforms cAS no matter what kind of PoF is used. This is not surprising since cAS only cares about immediate improvement based on samples with currently optimal values. Then we consider the impact of considering dependence between constraints (i.e., comparison between PoF and Dep-PoF). Among these benchmarks, Dep-PoF outperforms PoF in G7 for both acquisition functions and in G10 for cAS, probably because that both G7 and G10 have very small feasible region(s) (When computing G7's feasible ratio in Table 2, we only sample 1 feasible point in 10^6 uniform samples!).

In the benchmarks other than G7 and G10 problem, the performances of Dep-PoF and PoF are hard to differentiate. We identify several possible explanations: (i) as shown in Table 2, Gramacy, Sasena, and G4 problems have large feasible region(s). BO probably focuses on improving the objective instead of finding feasible solutions, (ii) as shown in Open Resource 2, some problems such as G8 and

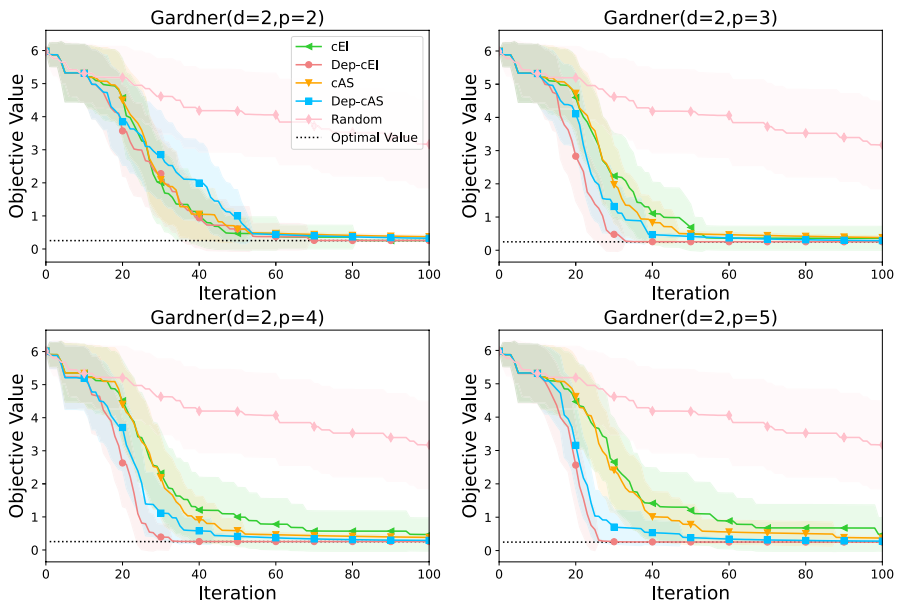


Fig. 1 Results for 2D Gardner function with repeated constraints, i.e. $p = \{2, \dots, 5\}$

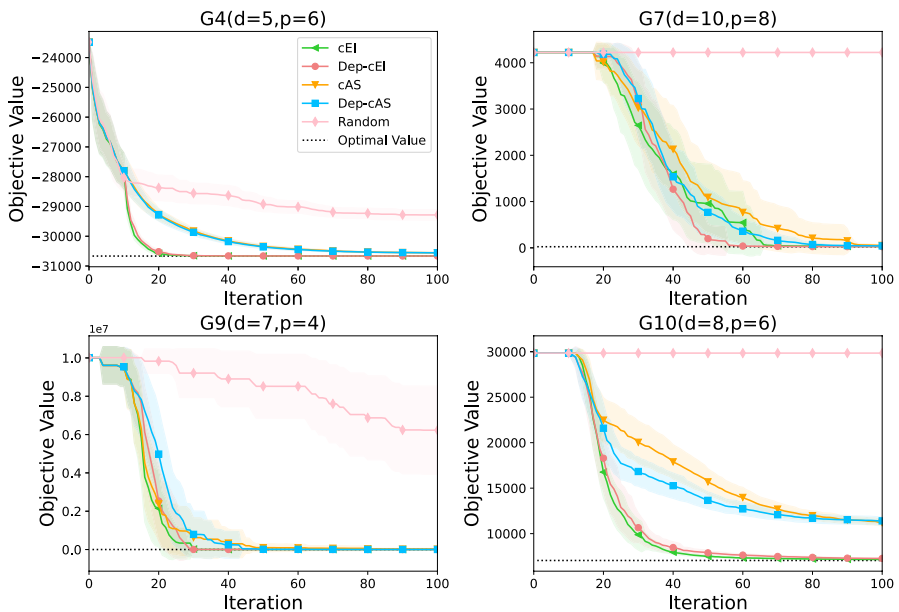


Fig. 2 Results for synthetic benchmarks

Table 2 Benchmarks. Recall that d is the input dimension and p is the number of constraints. For each problem, we also give the type of optimal solution and the active set that consists of the indexes of active constraints at optimum. To roughly show the size of feasible region(s), we uniformly sample 10^6 points and check their feasibility

Problem	d	p	Optimum	Active set	ρ (%)	Studied by
Gardner	2	1	Boundary	{1}	1.6226	[21, 25]
Gramacy	2	2	Boundary	{1}	45.6916	[25–27, 41]
Sasena	2	3	Boundary	{1, 3}	17.7891	[9, 17, 45]
G4	5	6	Boundary	{1, 6}	26.9618	[46, 47]
G6	2	2	Boundary	{1, 2}	1.1057	[46, 47]
G7	10	8	Boundary	{1, 2, 3, 4, 5, 7}	0.0001	[46–49]
G8	2	2	Interior	\emptyset	0.9614	[46, 47]
G9	7	4	Boundary	{1, 4}	0.5455	[18, 46–49]
G10	8	6	Boundary	{1, 2, 3, 4, 5, 6}	0.0003	[46–49]
Tension–compression	3	4	Boundary	{1, 2}	0.7644	[41, 47, 50, 51]
Pressure vessel	4	3	Boundary	{1, 3}	16.8690	[41, 47, 50, 51]
Welded beam	4	5	Boundary	{3}	37.4383	[41, 47, 50, 51]
Speed reducer	7	11	Boundary	{5, 6, 8}	0.1468	[41, 50, 51]

The ρ column shows the ratio of feasible points to all points

G9 problems have slight correlations between constraints, (iii) the strong correlations between constraints are negative, for example, G4 and G6 problems. If two constraints are negatively correlated, a larger probability of satisfying one constraint implies a smaller probability of satisfying another one. In this case, considering dependence can barely help find the feasible domain(s).

4.3 Real-world applications

Practically, we compare PoF and Dep-PoF on four real-world scenarios:

3D tension-compression spring design [41, 47, 50, 51]: minimize the weight of a tension or compression spring under 4 mechanical constraints.

4D pressure vessel design [41, 47, 50, 51]: minimize the cost of designing a cylindrical vessel subject to 4 constraints.

4D welded beam design [41, 47, 50, 51]: minimize the cost of design a welded beam satisfying 5 constraints.

7D speed reducer [41, 50, 51]: minimize the weight of a speed reducer constrained by 11 mechanical constraints.

Open Resource 3 provides more details about these applications. Figure 3 reports numerical results. Although Dep-PoF stills shares similar performance as PoF, both of them perform well in these practical problems with real constraints instead of artificial ones. It is noteworthy that these applications are also considered in [41], where the proposed SCBO algorithms and several state-of-art methods including cEI are tested and compared. The cEI's performance reported in [41] is usually not

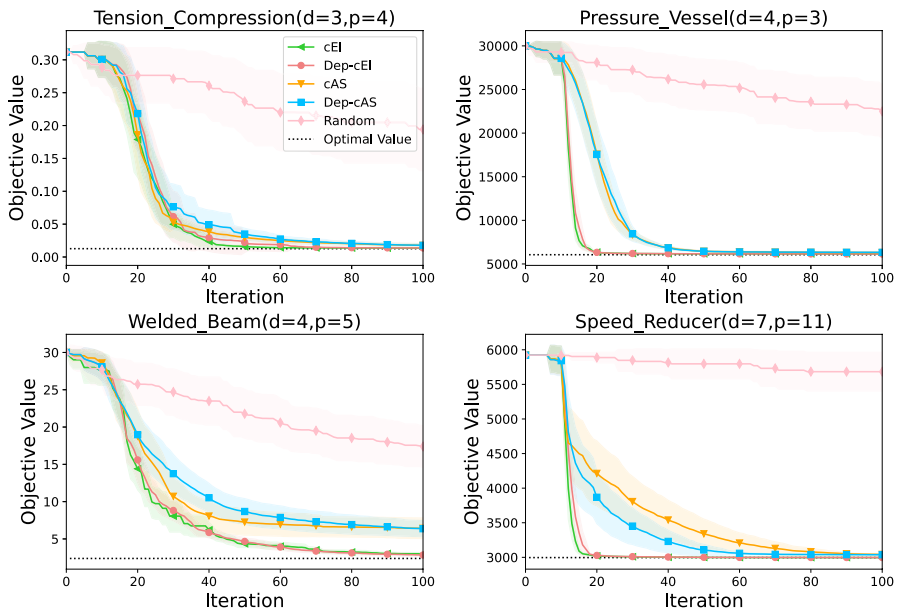


Fig. 3 Results for engineering benchmarks. Since the ranges of objective values for Pressure Vessel and Welded Beam are too wide, we set the maximal value to 30000 and 30, respectively

satisfying w.r.t. the speed and the extent of approaching the optimum. The differences between our results and theirs may result from our practical treatments (see Open Resource 1), whose effectiveness is shown here.

5 Discussion

Based on the convergence analysis and empirical results, we observe:

Dep-PoF is not worse than PoF. Theoretically, Dep-PoF represents accurate probability while PoF is a special case for diagonal covariance matrix \mathbf{K} . They are equivalent for cases with very little dependence. For cases with dependence, Dep-PoF is more likely to positively influence the results because \mathbf{K} is more accurate calculation. Practically, to conclude that Dep-PoF is not worse than PoF, several assumptions should be satisfied: (i) both models fit well, (ii) dependence is captured and learned, (iii) Dep-PoF is calculated correctly. Open Resource 2 verifies the first two assumptions are verified in Open Resource 2, and the results in [33] guarantee the last assumption.

Dep-PoF is very similar to PoF after sufficient samples. Intuitively, when surrogate models well-approximate the underlying constraints, Dep-PoF is largely determined by the predicted mean (equivalent to PoF) rather than the predicted covariance

matrix (where dependence matters). This 'convergence' indicates that Dep-PoF only helps in early stages of the optimization process.

Improvement versus Additional cost. Though EPMGP is effective, it cannot be as quick as computing the cdf for univariate case. Moreover, training a MOGP model is more expensive than training p GPs as MOGP involves more parameters. The improvement of Dep-PoF, if exists, may be limited compared to the extra computational effort. The reason we still consider dependence is for real expensive black-box functions requiring days to evaluate: If we truly know nothing about the underlying constraints, then Dep-PoF may help. Otherwise, if the dependence does not help or helps little, a user will not feel guilty to assume independence.

6 Conclusions

This paper removes the independence assumption between multiple black-box constraints and implements the Probability of Feasibility with dependence (Dep-PoF) via MOGP and EPMGP. Theoretically, we show that Dep-PoF converges to PoF with enough samples for a non-boundary region and we highlight their difference on/near the boundary. The former implies that Dep-PoF finally shares similar performance to PoF, while the latter explores the potential benefit of Dep-PoF for cases with boundary optimum and constraints with strong correlations.

The experimental results are largely negative: Dep-PoF and PoF typically perform similarly. These (negative) results are useful because they allow us to make a simplifying assumption without concerns. If the dependence between constraints is slight, it hardly helps. For stronger dependence, the surrogate model cannot fully capture it when the samples are scarce. After we have enough samples, the dependence is learned well but is not so important since Dep-PoF converges to PoF. These results encourage modeling simplicity and optimization efficiency.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11590-023-02047-z>.

Acknowledgements The research was funded by an Imperial College Hans Rausing PhD Scholarship to SZ, Engineering & Physical Sciences Research Council Grant EP/P016871/1, a BASF/Royal Academy of Engineering Research Chair to RM, and BASF.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References


1. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Process. Syst.* **25**, 52 (2012)
2. Shahriari, Bobak, Swersky, Kevin, Wang, Ziyu, Adams, Ryan P., De Freitas, Nando: Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **104**(1), 148–175 (2016)
3. Frazier, Peter I.: A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, (2018)
4. Rasmussen, Carl Edward, Williams, Christopher K.: *Gaussian Process*. Mach. Learn. MIT press Cambridge, MA (2006)
5. Gelbart, Michael A., Snoek, Jasper, Adams, Ryan P.: Bayesian optimization with unknown constraints. In: *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 250–259 (2014)
6. Thebelt, Alexander, Kronqvist, Jan, Mistry, Miten, Lee, Robert M., Sudermann-Merx, Nathan, Misener, Ruth: ENTMOOT: A framework for optimization over ensemble tree models. *Comput. Chem. Eng.* **151**, 107343 (2021)
7. Thebelt, Alexander, Wiebe, Johannes, Kronqvist, Jan, Tsay, Calvin, Misener, Ruth: Maximizing information from chemical engineering data sets: Applications to machine learning. *Chem. Eng. Sci.* **252**, 117469 (2022)
8. Ludl, Patrick Otto, Heese, Raoul, Höller, Johannes, Asprión, Norbert, Bortz, Michael: Using machine learning models to explore the solution space of large nonlinear systems underlying flowsheet simulations with constraints. *Front. Chem. Sci. Eng.* **58**, 1–15 (2021)
9. Boukouvala, Fani, Ierapetritou, Marianthi G.: Derivative-free optimization for expensive constrained problems using a novel expected improvement objective function. *AIChE J.* **60**(7), 2462–2474 (2014)
10. Boukouvala, Fani, Misener, Ruth, Floudas, Christodoulos A.: Global optimization advances in Mixed-Integer Nonlinear Programming, MINLP, and Constrained Derivative-Free Optimization. *CDFO Eur. J. Oper. Res.* **252**(3), 701–727 (2016)
11. Tran, Anh, Sun, Jing, Furlan, John M., Pagalthivarthi, Krishnan V., Visintainer, Robert J., Wang, Yan: pBO-2GP-3B: A batch parallel known/unknown constrained Bayesian optimization with feasibility classification and its applications in computational fluid dynamics. *Comput. Methods Appl. Mech. Eng.* **347**, 827–852 (2019)
12. Thebelt, Alexander, Tsay, Calvin, Lee, Robert M., Sudermann-Merx, Nathan, Walz, David, Tranter, Tom, Misener, Ruth: Multi-objective constrained optimization for energy applications via tree ensembles. *Appl. Energy* **306**, 118061 (2022)
13. Schonlau, Matthias, Welch, William J., Jones, Donald R.: Global versus local search in constrained optimization of computer models. *Lect. Notes Monogr. Ser.* **587**, 11–25 (1998)
14. Mockus, Jonas, Tiesis, Vytautas, Zilinskas, Antanas: The application of Bayesian methods for seeking the extremum. *Toward. Global Optim.* **2**(2), 117–129 (1978)
15. Gramacy, Robert B., Lee, Herbert K.H.: Optimization under unknown constraints. *arXiv preprint arXiv:1004.4027*, (2010)
16. Lee, Herbert K.H., Gramacy, Robert B., Linkletter, Crystal, Gray, Genetha A.: Optimization subject to hidden constraints via statistical emulation. *Pac. J. Optim.* **7**(3), 467–478 (2011)
17. Basudhar, Anirban, Dribusch, Christoph, Lacaze, Sylvain, Missoum, Samy: Constrained efficient global optimization with support vector machines. *Struct. Multidiscip. Optim.* **46**(2), 201–221 (2012)
18. Sacher, Matthieu, Duvigneau, Régis., Le Maitre, Olivier, Durand, Mathieu, Berrini, Elisa, Hauville, Frédéric., Astolfi, Jacques-André.: A classification approach to efficient global optimization in presence of non-computable domains. *Struct. Multidiscip. Optim.* **58**(4), 1537–1557 (2018)
19. Griffiths, Ryan-Rhys., Hernández-Lobato, José Miguel.: Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chem. Sci.* **11**(2), 577–586 (2020)
20. Heese, Raoul, Bortz, Michael: Adaptive sampling of pareto frontiers with binary constraints using regression and classification. In: *International Conference on Pattern Recognition*, pp. 3404–3411, (2021)
21. Gardner, Jacob R., Kusner, Matt J., Xu, Zhixiang Eddie, Weinberger, Kilian Q., Cunningham, John P.: Bayesian optimization with inequality constraints. *Int. Conf. Mach. Learn.* **2014**, 937–945 (2014)

22. Picheny, Victor: A stepwise uncertainty reduction approach to constrained global optimization. In: *Artificial Intelligence and Statistics*. PMLR, pp. 787–795. (2014)
23. Picheny, Victor, Gramacy, Robert B., Wild, Stefan, Le Digabel, Sébastien: Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. In: *International Conference on Neural Information Processing Systems* (2016)
24. Hernández-Lobato, José Miguel., Gelbart, Michael A., Adams, Ryan P., Hoffman, Matthew W., Ghahramani, Zoubin: A general framework for constrained Bayesian optimization using information-based search. *J. Mach. Learn. Res.* **17**, 1–53 (2016)
25. Ariafar, Setareh, Coll-Font, Jaume, Brooks, Dana, Dy, Jennifer: ADMMBO: Bayesian optimization with unknown constraints using ADMM. *J. Mach. Learn. Res.* **20**(123), 1–26 (2019)
26. Gramacy, Robert B., Gray, Genetha A., Le Digabel, Sébastien., Lee, Herbert KH., Ranjan, Pritam, Wells, Garth, Wild, Stefan M.: Modeling an augmented Lagrangian for blackbox constrained optimization. *Technometrics* **58**(1), 1–11 (2016)
27. Hernández-Lobato, José Miguel, Gelbart, Michael A., Hoffman, Matthew, Adams, Ryan P., Ghahramani, Zoubin: Predictive entropy search for Bayesian optimization with unknown constraints. In: *International Conference on Machine Learning*, (2015)
28. Breiman, Leo: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
29. Hearst, Marti A., Dumais, Susan T., Osuna, Edgar, Platt, John, Scholkopf, Bernhard: Support vector machines. *IEEE Intell. Syst. Appl.* **13**(4), 18–28 (1998)
30. Suykens, Johan AK., Vandewalle, Joos: Least squares support vector machine classifiers. *Neural Process. Lett.* **9**(3), 293–300 (1999)
31. Swersky, Kevin, Snoek, Jasper, Adams, Ryan P.: Multi-task Bayesian optimization. *Adv. Neural Inf. Process. Syst.* **26**, 2004–2012 (2013)
32. Álvarez, Mauricio A., Rosasco, Lorenzo, Lawrence, Neil D.: Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.* **4**(3), 195–266 (2012)
33. Cunningham, J.P., Hennig, P., Lacoste-Julien, S.: Gaussian probabilities and expectation propagation. *arXiv preprint arXiv:1111.6832*, (2011)
34. Micchelli, Charles A., Pontil, Massimiliano: On learning vector-valued functions. *Neural Comput.* **17**(1), 177–204 (2005)
35. van der Wilk, M., Dutoit, V., John, S.T., Artemev, A., Adam, V., Hensman, J.: A framework for interdomain and multioutput Gaussian processes. *arXiv:2003.01115*, (2020)
36. Bonilla, Edwin V., Chai, K., Williams, C.: Multi-task Gaussian process prediction. *Adv. Neural Inf. Process. Syst.* **20**, 752 (2007)
37. Matthews, Alexander G de G., van der Wilk, Mark, Nickson, Tom, Fujii, Keisuke, Boukouvalas, Alexis, León-Villagrà, Pablo, Ghahramani, Zoubin, Hensman, James: GPflow: A Gaussian process library using TensorFlow. *J. Mach. Learn. Res.* **18**(40), 1–6 (2017)
38. Genz, Alan, Bretz, Frank: *Comput. Multivar. Normal Probab.*, vol. 195. Springer Science & Business Media (2009)
39. Minka, Thomas P.: Expectation propagation for approximate Bayesian inference. *Uncertain. Artif. Intell.* **17**, 362–369 (2001)
40. Gelbart, Michael A.: *Constrained Bayesian optimization and applications*. PhD thesis, Harvard University, (2015)
41. Eriksson, David, Poloczek, Matthias: Scalable constrained Bayesian optimization. *International Conference on Artificial Intelligence and Statistics*, pp. 730–738 (2021)
42. Heese, Raoul, Walczak, Michał, Seidel, Tobias, Asprion, Norbert, Bortz, Michael: Optimized data exploration applied to the simulation of a chemical process. *Comput. Chem. Eng.* **124**, 326–342 (2019)
43. Teckentrup, Aretha L.: Convergence of Gaussian process regression with estimated hyperparameters and applications in Bayesian inverse problems. *SIAM/ASA J. Uncert. Quantif.* **8**(4), 1310–1337 (2020)
44. Virtanen, Pauli, Gommers, Ralf, Oliphant, Travis E., Haberland, Matt, Reddy, Tyler, Cournapeau, David, Burovski, Evgeni, Peterson, Pearu, Weckesser, Warren, Bright, Jonathan, et al.: SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**(3), 261–272 (2020)
45. Sasena, Michael James: *Flexibility and efficiency enhancements for constrained global design optimization with kriging approximations*. PhD thesis, University of Michigan, (2002)
46. Michalewicz, Zbigniew, Schoenauer, Marc: Evolutionary algorithms for constrained parameter optimization problems. *Evolut. Comput.* **4**(1), 1–32 (1996)

47. Hedar, Abdel-Rahman., Fukushima, Masao: Derivative-free filter simulated annealing method for constrained continuous global optimization. *J. Glob. Optim.* **35**(4), 521–549 (2006)
48. Michalewicz, Zbigniew: Genetic algorithms, numerical optimization, and constraints. In: *Proceedings of the sixth international conference on genetic algorithms*, Vol 195, pp. 151–158. Citeseer, (1995)
49. Michalewicz, Zbigniew, Nazhiyath, Girish: Genocop III: A co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In: *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, vol 2, pp. 647–651. IEEE, (1995)
50. Coello Coello, Carlos A., Montes, Efrén Mezura.: Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Adv. Eng. Inf.* **16**(3), 193–203 (2002)
51. Lemonge, Afonso CC., Barbosa, Helio JC., Borges, Carlos CH., Silva, Francilene BH.: Constrained optimization problems in mechanical engineering design using a real-coded steady-state genetic algorithm. *Mech. Comput.* **29**(95), 9287–9303 (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Shiqiang Zhang¹  · Robert M. Lee² · Behrang Shafei² · David Walz² · Ruth Misener¹

✉ Shiqiang Zhang
s.zhang21@imperial.ac.uk

Robert M. Lee
robert-matthew.lee@basf.com

Behrang Shafei
behrang.shafei@basf.com

David Walz
david-simon.walz@basf.com

Ruth Misener
r.misener@imperial.ac.uk

¹ Imperial College London, South Kensington SW7 2AZ, UK

² BASF SE, Ludwigshafen am Rhein, Germany