
Failure-Aware Gaussian Process Optimization with Regret Bounds

Shogo Iwazaki
MI-6 Ltd., Tokyo, Japan
iwazaki@mi-6.co.jp

Shion Takeno
RIKEN AIP, Tokyo, Japan
shion.takeno@riken.jp

Tomohiko Tanabe
MI-6 Ltd., Tokyo, Japan
tanabe@mi-6.co.jp

Mitsuru Irie
MI-6 Ltd., Tokyo, Japan
irie@mi-6.co.jp

Abstract

Real-world optimization problems often require black-box optimization with observation failure, where we can obtain the objective function value if we succeed, otherwise, we can only obtain a fact of failure. Moreover, this failure region can be complex by several latent constraints, whose number is also unknown. For this problem, we propose a failure-aware Gaussian process upper confidence bound (F-GP-UCB), which only requires a mild assumption for the observation failure that an optimal solution lies on an interior of a feasible region. Furthermore, we show that the number of successful observations grows linearly, by which we provide the first regret upper bounds and the convergence of F-GP-UCB. We demonstrate the effectiveness of F-GP-UCB in several benchmark functions, including the simulation function motivated by material synthesis experiments.

1 Introduction

The effectiveness of optimization methods based on Gaussian process (GP) regression for expensive-to-evaluate black-box functions has been repeatedly shown in a wide range of real-world applications, including robotics [34, 35], experimental design [14], and hyperparameter optimization [4, 41]. On the other hand, failure of the observation itself must often be considered. For example, in the optimization of hyperparameters for a complicated physical model, the evaluation may crash for some hyperparameters. Another example is materials development, in which experimental testing of new materials can reveal that the synthesis procedure fails.

Therefore, this study considers the optimization of a black-box function f with a black-box deterministic failure function $c : \mathcal{X} \rightarrow \{0, 1\}$, where \mathcal{X} is an input space. In this study, failure of observation at the input \mathbf{x} is represented as $c(\mathbf{x}) = 1$, while the success of observation is represented as $c(\mathbf{x}) = 0$. Then, our optimization problem can be formulated as follows:

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \quad s.t. \quad c(\mathbf{x}) = 0, \quad (1)$$

where the observation of f can be obtained only if $c(\mathbf{x}) = 0$, but the observation cost is consumed even if $c(\mathbf{x}) = 1$. Hence, the goal is to efficiently identify the optimal point \mathbf{x}^* while considering the observation failure.

One of the notable technical difficulties of problem (1) is how to handle the failure function c . To model c , existing studies [32, 2] use the GP classification (GPC) model [38] or its variants, which assume that c can be represented by a smooth latent function. However, the observation failure can

be caused by several latent constraints implicitly. In addition, since only the binary failure can be obtained, the number of latent constraints is also unknown. Therefore, modeling c with the usual GPC is hard and often unsuitable, which can degrade the optimization performance. Hence, an optimization method under a mild assumption on c is demanded. Furthermore, the existence of c makes theoretical analysis and even securing the number of successful observations difficult.

Our contribution. In this paper, we propose a novel GP-based optimization algorithm, failure-aware GP upper confidence bound (F-GP-UCB), which chooses the next evaluation in an input domain except for the adaptively adjusted neighborhood of the past observation failures. Furthermore, we show that the number of successful observations grows linearly under a very mild assumption of the failure function c that the optimal solution lies on an interior of a feasible region. Then, we provide the first regret upper bound of the GP optimization problem (1), which shows that F-GP-UCB converges to the optimal solution with high probability. We demonstrate the effectiveness of the F-GP-UCB algorithm with several benchmark functions, including a heuristic simulation function motivated by materials research of quasicrystals.

Related work. In the past few decades, many GP-based optimization algorithms have been developed [37, 42, 18, 8, 53]. In addition to the standard optimization settings, various extensions have been studied, such as parallel [9], high-dimensional [54], multi-fidelity [25], and robust optimization [6].

GP-based constrained optimization [10, 11, 19, 47] has a close relation to our study. It considers the black-box optimization under an inequality constraint $g(\mathbf{x}) \leq 0$, where g is also a black-box function. If the failure function in this paper is recast as $c(\mathbf{x}) = \mathbb{1}\{g(\mathbf{x}) > 0\}$, the optimization problem matches that of problem (1). However, there are two crucial differences. First, in constrained optimization, observations at an input can be obtained even if the constraint is not satisfied, i.e., $g(\mathbf{x}) > 0$. Under the setting of this paper, no information about the objective function can be obtained for input points that result in failure. Second, in constrained optimization problems, it is generally assumed that noisy observation of $g(\mathbf{x})$ is possible. In the setting of our study, we cannot obtain a direct observation from the latent function g .

There exist works on GP optimization that take into account failures. Lindberg and Lee [32] proposed an algorithm that combines the Expected Improvement (EI) criterion [37] with the posterior success probability of the classical GPC. Instead of GPC, Sacher et al. [39] used a support vector machine-based model. However, although c is deterministic, these classifiers assume that observation failure is essentially stochastic, i.e., the evaluations at the same input can fail and succeed. This model misspecification can degrade the optimization performance, as described in Bachoc et al. [2]. Then, Bachoc et al. [2] proposed the deterministic variant of GPC which models c as $c(\mathbf{x}) = \mathbb{1}\{g(\mathbf{x}) \geq 0\}$, where g is a latent function modeled by a GP. They also provide an EI-based strategy and prove convergence to an optimal solution. From a practical point of view, surrogate models of c navigate the optimization process efficiently when c is well represented through a smooth latent function g . However, practical applications often have too complex failure processes to model with one latent function (e.g., the failure function c depends on several latent functions.) Furthermore, it is difficult to know that such a complex failure structure exists beforehand. Finally, from a theoretical perspective, Bachoc et al. [2] give the convergence guarantee, but its rate and the regret-based analysis are not provided.

2 Preliminaries

Problem setup. Let $f : \mathcal{X} \rightarrow \mathbb{R}$ and $c : \mathcal{X} \rightarrow \{0, 1\}$ be an unknown fixed objective function and an unknown failure function, respectively, where $\mathcal{X} := [0, 1]^d$ is the input space. At each time step t , the user makes a selection $\mathbf{x}_t \in \mathcal{X}$ and obtains the failure label $c(\mathbf{x}_t)$. If $c(\mathbf{x}_t) = 0$, the user proceeds to make a noisy observation $y_t = f(\mathbf{x}_t) + \epsilon_t$ where ϵ_t is a noise term which is conditionally σ -sub-Gaussian. Our noise model is a mild one; examples include an arbitrary distribution over $[-\sigma, \sigma]$ and a Gaussian noise with variance below σ^2 . Note that, in the case of $c(\mathbf{x}_t) = 1$ (failure), the user obtains no further information.

The user’s goal is to efficiently identify the optimal solution \mathbf{x}^* over the unknown feasible region $S_c := \{\mathbf{x} \in \mathcal{X} \mid c(\mathbf{x}) = 0\}$. For convenience, we rephrase the problem (1) using S_c as follows:

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in S_c} f(\mathbf{x}). \quad (2)$$

We assume that $S_c \neq \emptyset$ and that there exists a unique solution \mathbf{x}^* . Furthermore, we define the failure region F_c as $F_c = \mathcal{X} \setminus S_c$.

Regret. We employ the *regret* to evaluate the algorithm's performance. The regret r_t at step t is defined as $r_t = f(\mathbf{x}^*) - f(\hat{\mathbf{x}}_t)$ if $\hat{\mathbf{x}}_t \in S_c$, otherwise $r_t = f(\mathbf{x}^*) - \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$, where $\hat{\mathbf{x}}_t \in \mathcal{X}$ is the algorithm's estimated solution¹. In our definition of r_t , the algorithm is supposed to incur the worst-case regret when $\hat{\mathbf{x}}_t$ is not in S_c . Similar definitions are also used in performance metrics of GP-based constrained optimization [19].

Regularity assumptions for the objective function. As a regularity assumption, we assume that f is an element of the reproducing kernel Hilbert space (RKHS) \mathcal{H}_k , corresponding to a known positive-definite kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, and has a bounded Hilbert norm $\|f\|_{\mathcal{H}_k} \leq B$. Furthermore, the kernel k is assumed to be normalized, namely $\forall \mathbf{x} \in \mathcal{X}, k(\mathbf{x}, \mathbf{x}) \leq 1$. These are common assumptions in existing GP optimization literature [42, 51, 8, 48].

Gaussian process modeling. Our algorithm uses the modeling information of f from GP [38], using only the successful observations. First, we assume zero-mean GP with the covariance function k as the prior of f . Next, we model the generating process of the value y_t queried at \mathbf{x}_t as $y_t = f(\mathbf{x}_t) + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$. We note that the assumptions on the GP prior on f and Gaussian noise, as stated above, are assumptions only for constructing the GP model, which can differ from the underlying true function f and the noise. Let $\mathcal{I}_t^{(S)} := \{i \in \{1, \dots, t\} \mid c(\mathbf{x}_i) = 0\}$ be the index set of successful observations, $\mathbf{X}_t^{(S)} := (\mathbf{x}_i)_{i \in \mathcal{I}_t^{(S)}}$ be the corresponding inputs, $\mathbf{y}_t^{(S)} := (y_i)_{i \in \mathcal{I}_t^{(S)}}$, be the corresponding outputs, and $n_t := |\mathcal{I}_t^{(S)}|$ be the number of successful observations. In addition, we also define the failure index set $\mathcal{I}_t^{(F)} := \{i \in \{1, \dots, t\} \mid c(\mathbf{x}_i) = 1\}$. Given $\mathbf{X}_t^{(S)}, \mathbf{y}_t^{(S)}$, the posterior distribution of $f(\mathbf{x})$ becomes a normal distribution, whose posterior mean $\mu_t(\mathbf{x})$ and posterior variance $\sigma_t^2(\mathbf{x})$ are given as follows:

$$\mu_t(\mathbf{x}) = \mathbf{k}(\mathbf{x}, \mathbf{X}_t^{(S)})^\top \left(\mathbf{K}(\mathbf{X}_t^{(S)}) + \sigma^2 \mathbf{I}_{n_t} \right)^{-1} \mathbf{y}_t^{(S)}, \quad (3)$$

$$\sigma_t^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x}, \mathbf{X}_t^{(S)})^\top \left(\mathbf{K}(\mathbf{X}_t^{(S)}) + \sigma^2 \mathbf{I}_{n_t} \right)^{-1} \mathbf{k}(\mathbf{x}, \mathbf{X}_t^{(S)}). \quad (4)$$

Here, \mathbf{I}_{n_t} is a $n_t \times n_t$ identity matrix. Furthermore, denoting the index of the j -th least element of $\mathcal{I}_t^{(S)}$ as i_j , $\mathbf{k}(\mathbf{x}, \mathbf{X}_t^{(S)}) \in \mathbb{R}^{n_t}$ represents a vector whose j -th element is $k(\mathbf{x}, \mathbf{x}_{i_j})$. Similarly, $\mathbf{K}(\mathbf{X}_t^{(S)}) \in \mathbb{R}^{n_t \times n_t}$ represents the kernel matrix whose jk -th element is $k(\mathbf{x}_{i_j}, \mathbf{x}_{i_k})$.

Lastly, we define the maximum information gain [42, 49] as a quantity representing the GP complexity. The maximum information gain $\gamma_{t,A}$ from observing t number of data points over the set $A \subset \mathbb{R}^d$ is defined as $\gamma_{t,A} = \max_{\mathbf{X} := (\mathbf{x}_1, \dots, \mathbf{x}_t) \in A^t} I(\mathbf{f}_{\mathbf{X}}, \mathbf{y}_{\mathbf{X}})$, where $I(\mathbf{f}_{\mathbf{X}}, \mathbf{y}_{\mathbf{X}}) := 0.5 \ln \det (\mathbf{I}_t + \sigma^{-2} \mathbf{K}(\mathbf{X}))$ is the mutual information between the latent function values $\mathbf{f}_{\mathbf{X}} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}(\mathbf{X}))$ and corresponding output values $\mathbf{y}_{\mathbf{X}} = \mathbf{f}_{\mathbf{X}} + \boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_t)$ of GP. In GP-based optimization, $\gamma_{t,A}$ is often used as a quantity that characterizes the confidence bound and regret bound of f . Furthermore, for a compact convex set A , the upper bound for $\gamma_{t,A}$ has been derived for some commonly used kernels. For example, $\gamma_{t,A} = \mathcal{O}((\ln t)^{d+1})$ for Gaussian kernel and $\gamma_{t,A} = \mathcal{O}(t^{d/(\nu+d)} (\ln t)^{2\nu/(\nu+d)})$ for Matérn kernel with the smoothness parameter $\nu > 1/2$ [42, 49]. The following Lemma 2.1 adapts the well-known result that gives the confidence bound of f for our problem setup, which is a direct consequence of Theorem 3.11 in [1].

Lemma 2.1. Fix $f \in \mathcal{H}_k$ with $\|f\|_{\mathcal{H}_k} \leq B$. Suppose the observation $y_t = f(\mathbf{x}_t) + \epsilon_t$ has a noise ϵ_t that is conditionally σ -sub-Gaussian. Define $\{\beta_t\}_{t \in \mathbb{N}}$ as $\beta_t^{1/2} = B + \sigma \sqrt{2(\gamma_{t-1, \mathcal{X}} + 1 + \ln(1/\delta))}$ for $\delta \in (0, 1)$. Then the following holds with probability at least $1 - \delta$:

$$\forall t \geq 1, \forall \mathbf{x} \in \mathcal{X}, \text{lcb}_t(\mathbf{x}) \leq f(\mathbf{x}) \leq \text{ucb}_t(\mathbf{x}). \quad (5)$$

Here, $\text{lcb}_t(\mathbf{x})$ and $\text{ucb}_t(\mathbf{x})$ are defined as $\text{lcb}_t(\mathbf{x}) = \mu_{t-1}(\mathbf{x}) - \beta_t^{1/2} \sigma_{t-1}(\mathbf{x})$ and $\text{ucb}_t(\mathbf{x}) = \mu_{t-1}(\mathbf{x}) + \beta_t^{1/2} \sigma_{t-1}(\mathbf{x})$, respectively.

¹Some applications prefer to assess the performance via cumulative regret. In Appendix C, the analysis of the cumulative regret of our algorithm is provided.

Regularity assumption for failure function. A regret upper bound cannot be obtained without any assumption on c . As an extreme example, we consider the case where \mathbf{x}^* is an isolated point surrounded by F_c . Under this scenario, the worst case exists in which an arbitrary algorithm can never observe \mathbf{x}^* in a finite number of trials. Therefore, in order to give a convergence guarantee, at least \mathbf{x}^* must be contained in a subset of the feasible region S_c having a non-zero volume. In this paper, we focus on the case that \mathbf{x}^* is the interior point of S_c as in the following assumption.

Assumption 2.2. *There exists $\eta > 0$ such that $N_{\mathbf{x}^*; \eta} \subset S_c$, where $N_{\mathbf{x}^*; \eta} := \{\mathbf{x} \in \mathcal{X} \mid \|\mathbf{x} - \mathbf{x}^*\|_\infty < \eta\}$ is an open infinity ball with a radius η centered at \mathbf{x}^* ².*

The parameter η above depends on the size of the subset of the feasible region that \mathbf{x}^* belongs to and is an important quantity for theoretical analysis. Note that Assumption 2.2 is quite mild. For example, when c is defined as $c(\mathbf{x}) = \mathbb{1}\{g(\mathbf{x}) \geq 0\}$ with a continuous latent function g , there exists $\eta > 0$ such that $N_{\mathbf{x}^*; \eta} \subset S_c$ from the continuity of g . Since the analysis of [2] assumes the existence of the latent function g , which is a *continuous* sample path generated from GP, our assumption also subsumes their assumption. Finally, note that η is not needed for running the algorithm and is only used as a quantity that characterizes the regret bound.

3 Proposed algorithm

Our proposed algorithm F-GP-UCB is shown as pseudo-code in Algorithm 1. Roughly speaking, our F-GP-UCB searches the input domain excluding the adaptively adjusted neighborhood of past failure points based on the existing GP-UCB [42] strategy. Below, we start by describing the background of the algorithm construction.

Philosophy of algorithm construction. Since F_c is unknown, the algorithm needs to appropriately avoid the failure observation in F_c while balancing the trade-off between exploration and exploitation in a way that guarantees convergence. The difficulty of this problem lies in the fact that there exist cases where the small feasible region containing \mathbf{x}^* is surrounded by the past failure observations and is isolated as shown in the left plot of Fig. 1. Since the algorithm is unable to exclude the possibility that an arbitrary small feasible region exists between failure points only from Assumption 2.2, the algorithm should be constructed so that it simultaneously satisfies the following two points: the failure observation should be avoided; and the inputs evaluated by the algorithm should eventually cover the arbitrarily close area of the past failure point.

F-GP-UCB satisfies the above requirements by adaptively controlling the GP-UCB search region by eliminating the neighborhood of the past failure points, and shrinking the eliminated region as step increase. Next, we describe below the details of and ideas behind each step.

Selection of \mathcal{X}_t . At each step t , the F-GP-UCB algorithm firstly computes the search region \mathcal{X}_t as

$$\mathcal{X}_t = \{\mathbf{x} \in \mathcal{X} \mid \forall i \in \mathcal{I}_{t-1}^{(F)}, \|\mathbf{x}_i - \mathbf{x}\|_\infty \geq \theta_t b(t)\}. \quad (6)$$

Here, θ_t and $b : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ are the scale parameter and monotonically decreasing function, respectively. The parameter θ_t is controlled by the algorithm at every step so that it decreases monotonically with respect to t from its initial value θ_0 given by the user. It plays the role of guaranteeing that the search space \mathcal{X}_t satisfies $\mathcal{X}_t \neq \emptyset$. The function b is defined by the user before running the algorithm.

By using \mathcal{X}_t , the F-GP-UCB algorithm chooses \mathbf{x}_t as

$$\mathbf{x}_t = \arg \max_{\mathbf{x} \in \mathcal{X}_t} \text{ucb}_t(\mathbf{x}). \quad (7)$$

The middle and right plots in Fig. 1 show an example behavior of the F-GP-UCB algorithm. By using the monotonically decreasing $\theta_t b(t)$ to control the search space, the algorithm's behavior can be qualitatively described as follows. First, during the early phase where t is small, $\theta_t b(t)$ is large, corresponding to choosing an unexplored point for observation while avoiding a large neighborhood of the past failure points where there is a high possibility of failure. Then as t increases and the

²Our algorithm and theoretical analysis can be generalized for arbitrary norms over \mathbb{R}^d such as L_2 -norm and for a compact convex subset $\mathcal{X} \subset \mathbb{R}^d$, however the computational technique discussed in Sec.5 assumes the infinity norm.

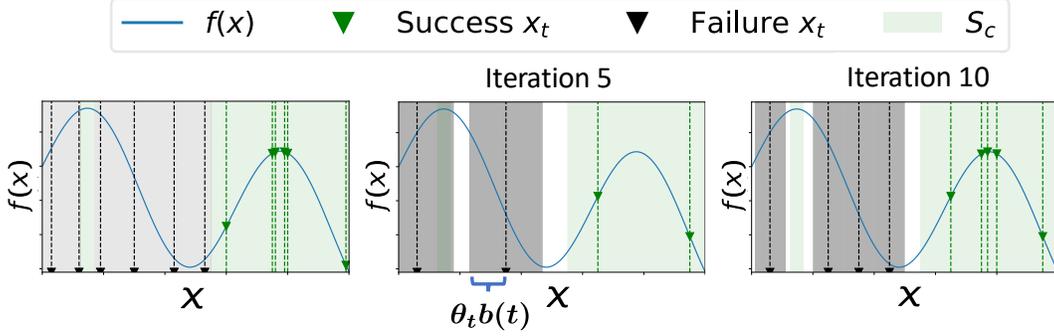


Figure 1: An example problem in one dimension. The left plot shows the situation where there exists an isolated feasible region, which includes the optimum. The green and grey shaded areas represent the feasible region S_c and the failure region F_c , respectively. The green (black) points represent the observed successful (failure) points. In this situation, it is necessary to identify the feasible region which hides among the observed failure points. The middle and right plots are example behaviors of the F-GP-UCB algorithm in $t = 5$ and $t = 10$, respectively. The gray shaded regions represent the neighborhood of the observed failure points which is excluded from the search space at the given step. The F-GP-UCB algorithm performs searches in all the feasible regions while avoiding the observed failure points by iteratively narrowing down the excluded search space in the neighborhood of the failure points.

remaining unexplored space shrinks, it is expected that the algorithm's behavior will become more aggressive in considering the possibility that the region near the observed failure points contains a feasible region that may be difficult to identify.

The choice of θ_0 and b affects the performance of F-GP-UCB. In Sec. 4, we show the convergence of the regret under an appropriate choice of b . In Sec. 5, the practical choices of θ_0 and b are discussed.

Shrinking θ_t . In the case where $\mathcal{X}_t = \emptyset$, \mathbf{x}_t is not defined. We ensure such cases do not occur by controlling the scale parameter θ_t . In our algorithm, the main role of θ_t is to guarantee that \mathbf{x}_t is well-defined as follows. Specifically, before the step t starts, θ_t is computed by halving the previous scale parameter θ_{t-1} until the union of all the neighborhoods does not fully cover \mathcal{X} . This procedure requires that we solve the set covering problem to find the cover of \mathcal{X} with the neighborhoods of the past failure points. Unfortunately, this problem is known to be NP complete [20]. In Sec. 5, we provide a practical approach to this problem. On the other hand, note that our analysis in Sec. 4 assumes that this procedure can be computed exactly.

Algorithm estimated solution. The estimated solution of the F-GP-UCB algorithm $\hat{\mathbf{x}}_t$ at step t is defined as follows:

$$\hat{\mathbf{x}}_t = \mathbf{x}_{\hat{t}} \text{ where } \hat{t} = \arg \max_{i \in \mathcal{I}_t^{(S)}} \text{lcb}_i(\mathbf{x}_i). \quad (8)$$

The definition of the estimated solution based on the observed points and lcb_i is often used in existing literature [6, 26, 21]. While the estimated solution of the proposed method resembles those in the

Algorithm 1 The F-GP-UCB algorithm

Input: $\theta_0 \in (0, 1)$, $b : \mathbb{R}_+ \rightarrow \mathbb{R}_+$, $\{\beta_t\}_{t \in \mathbb{N}_+}$.

- 1: Initialize GP prior and set $\mathcal{I}_0^{(S)} = \mathcal{I}_0^{(F)} = \emptyset$.
- 2: **for** $t = 1$ to T **do**
- 3: $\theta_t \leftarrow \theta_{t-1}$.
- 4: **while** $\mathcal{X} \subset \bigcup_{i \in \mathcal{I}_t^{(F)}} N_{\mathbf{x}_i; \tilde{\theta}_t b(t)}$ **do**
- 5: $\tilde{\theta}_t \leftarrow \tilde{\theta}_t / 2$.
- 6: **end while**
- 7: $\theta_t \leftarrow \tilde{\theta}_t$ and define \mathcal{X}_t as in (6).
- 8: Choose $\mathbf{x}_t = \arg \max_{\mathbf{x} \in \mathcal{X}_t} \text{ucb}_t(\mathbf{x})$.
- 9: **if** $c(\mathbf{x}_t) = 0$ **then**
- 10: Observe $y_t = f(\mathbf{x}_t) + \epsilon_t$ and update GP.
- 11: $\mathcal{I}_t^{(S)} \leftarrow \mathcal{I}_{t-1}^{(S)} \cup \{t\}$, $\mathcal{I}_t^{(F)} \leftarrow \mathcal{I}_{t-1}^{(F)}$.
- 12: **else**
- 13: $\mathcal{I}_t^{(S)} \leftarrow \mathcal{I}_{t-1}^{(S)}$, $\mathcal{I}_t^{(F)} \leftarrow \mathcal{I}_{t-1}^{(F)} \cup \{t\}$.
- 14: **end if**
- 15: **end for**
- 16: Calculate $\mathbf{x}_{\hat{T}}$ as in (8) if $\mathcal{I}_T^{(S)} \neq \emptyset$.

literature, it should be noted that the maximum of lcb_i is computed over $\mathcal{I}_t^{(S)}$. Note $\hat{\mathbf{x}}_t$ is not defined if all the points up to t are failures (i.e., $\mathcal{I}_t^{(S)} = \emptyset$). In Sec. 4, it is shown that there always exists $\hat{\mathbf{x}}_T$ for a sufficiently large step size T . Finally, although the estimated solution (8) is useful for discussing the theoretical performance, for practical purposes, $\hat{\mathbf{x}}_t = \mathbf{x}_{\hat{t}}$ where $\hat{t} = \arg\max_{i \in \mathcal{I}_t^{(S)}} \text{lcb}_t(\mathbf{x}_i)$ is often used instead [6, 21]. We use this modified definition of $\hat{\mathbf{x}}_t$ in the benchmark experiments in Sec. 6.

4 Theoretical analysis

The results of the theoretical analysis are discussed in this section. The complete proofs of the theorems and lemmas are described in Appendix A.

Key of regret analysis. To analyze the regret, we start by evaluating the growth rate of the number of successful observations n_t in the algorithm. By noting that $\theta_t b(t)$ is monotonically decreasing, we can find that the failure points of F-GP-UCB are separated by at least a distance $\theta_t b(t)$ apart from each other. Therefore, n_t can be evaluated via $\theta_t b(t)$ -packing number of F_c . The following Lemma 4.1 is based on the standard argument for the packing number, which gives the lower bound for the number of successful observations n_t .

Lemma 4.1. *When running Algorithm 1, $n_t \geq t - \mathcal{P}(\mathcal{X}, \|\cdot\|_\infty, \theta_t b(t)/4)$ holds for any $t \in \mathbb{N}_+$, where $\mathcal{P}(\mathcal{X}, \|\cdot\|_\infty, \epsilon)$ denotes the ϵ -packing number over the set \mathcal{X} with respect to the norm $\|\cdot\|_\infty$.*

Using the above Lemma 4.1, designing the choice of b so that $t - \mathcal{P}(\mathcal{X}, \|\cdot\|_\infty, \theta_t b(t)/4) = \Theta(t)$ holds leads to $n_t = \Omega(t)$, which means that the number of successful observations can be secured at the same rate as in the standard GP optimization setting. By using the fact $\mathcal{P}(\mathcal{X}, \|\cdot\|_\infty, \epsilon) = \lceil 1/\epsilon \rceil^d$ (Lemma A.4), the choice of $b(t) = o(t^{-1/d})$ is sufficient for $n_t = \Omega(t)^3$. While our result holds for any b satisfying $b(t) = o(t^{-1/d})$, we give results for b satisfying $b(t) = t^{-\alpha}$ with $\alpha \in (0, 1/d)$ in this section for simplicity. In Appendix A, the result for general \mathcal{X} and b is also provided.

Regret bound. The following Theorem 4.2 gives the regret bound of the F-GP-UCB algorithm.

Theorem 4.2. *Fix $\delta \in (0, 1)$, $\alpha \in (0, d^{-1})$, $\mathcal{X} = [0, 1]^d$, and $f \in \mathcal{H}_k$ such that $\|f\|_{\mathcal{H}_k} \leq B$. Let $\beta_t^{1/2} = B + \sigma \sqrt{2(\gamma_{t-1; \mathcal{X}} + 1 + \ln(1/\delta))}$ and $b(t) = t^{-\alpha}$. In addition, suppose that there exists $\eta > 0$ such that $N_{\mathbf{x}^*; \eta} \subset S_c$, where $N_{\mathbf{x}^*; \eta} = \{\mathbf{x} \in \mathcal{X} \mid \|\mathbf{x} - \mathbf{x}^*\|_\infty < \eta\}$. When applying Algorithm 1 under the above conditions, the estimated solution $\hat{\mathbf{x}}_T$ defined in (8) exists for all $T \geq \tilde{T}_s := \max\{(10/\bar{\theta}^d)^{1/(1-d\alpha)}, 2\} + 1$, where $\bar{\theta} = \min\{\theta_0, \eta/2\}$. Moreover, the following holds with probability at least $1 - \delta$:*

$$\forall T \geq \tilde{T}_s, r_T \leq \frac{2}{T} \left[2B(\tilde{T}_* - 1) + \sqrt{C_1 \beta_T T \gamma_{T; S_c}} \right], \quad (9)$$

where $\tilde{T}_* = (\theta_0/\eta)^{1/\alpha} + 1$ and $C_1 = 8/\ln(1 + \sigma^{-2})$.

In the analysis above, η is interpreted as the complexity of the failure function c , and $\bar{\theta}$ corresponds to the lower bound on θ_t while running the algorithm. The constants \tilde{T}_s and \tilde{T}_* are important quantities that characterize the regret and can be respectively interpreted as follows.

- The constant \tilde{T}_s is the upper bound on the number of steps until the first successful observation.
- The constant \tilde{T}_* is the upper bound on the number of steps needed until the search space \mathcal{X}_t of the algorithm always contains the optimal solution \mathbf{x}^* .

The regret bound in Theorem 4.2 can be intuitively seen as evaluating the algorithm execution in terms of two stages along the time axis. The first term of (9) represents the regret generated when the algorithm's search space \mathcal{X}_t does not contain \mathbf{x}^* . Specifically, the worst-case regret incurred during this phase is bounded by $2B$ using the fact that $\sup_{\mathbf{x} \in \mathcal{X}} |f(\mathbf{x})| \leq B$ from the kernel normalization

³Strictly speaking, we use the fact that θ_t becomes constant for sufficiently large t . See Lemma A.8 in Appendix A.

condition. The second term is the regret incurred during the process of identifying \mathbf{x}^* after \mathcal{X}_t includes \mathbf{x}^* and is similar to the regret bound of standard GP-UCB. The only difference is that, in our setup, the observations of GP are only on S_c , so the maximum information gain is defined over S_c .

By noting that $\gamma_{T;S_c} \subset \gamma_{T;\mathcal{X}}$ in the second term and using the known results concerning the upper bound on $\gamma_{T;\mathcal{X}}$, the more explicit form of regret upper bound can be obtained on a per-kernel basis. For example, in the case of the Gaussian kernel, we have that $\gamma_{T;\mathcal{X}} = \mathcal{O}((\ln T)^{d+1})$ and $r_T = \mathcal{O}\left(T^{-1}B\tilde{T}_* + T^{-1/2}(\ln T)^{(d+1)/2}\left(B + \sigma\sqrt{(\ln T)^{d+1} + \ln(1/\delta)}\right)\right)$. Thus, indeed we obtain $r_T \rightarrow 0$ (as $T \rightarrow \infty$). The F-GP-UCB algorithm is guaranteed to converge when the cumulative regret for ordinary GP-UCB (corresponding to the second term in (9)) becomes sub-linear under the chosen setup and the kernel⁴.

The case where \mathbf{x}^* exists on the boundary of S_c . The case where \mathbf{x}^* exists on the boundary S_c is not covered in Theorem 4.2, which is the case that often appears in a real-world application. If undesirable Lipschitz-style dependencies are allowed to emerge in the regret upper bound, our F-GP-UCB algorithm is also guaranteed to have convergence in such cases (Theorem B.2). The details of the result and discussion about its limitation are in Appendix B.

5 Practical considerations

In this section, we discuss several issues that arise in practical situations and provide their solutions. Due to the space limitation, we only give brief descriptions. The details are described in Appendix E.

Computation of θ_t . The computation of θ_t requires us to know whether the union of all the neighborhoods of the failure points covers \mathcal{X} . A simple solution might be to partition \mathcal{X} into a sufficiently fine grid by subdividing each axis and look for a feasible solution in each grid cell by brute force. However, as the step size t and dimension d become large, the number of grid cells grows rapidly. Furthermore, even a very fine grid will not be able to detect the case where a feasible solution exists on the cell boundary. Another heuristic approach is to use a constrained optimization solver which does not require a feasible solution as part of the initial points (e.g., penalty function method and augmented Lagrangian method [17]) and try to solve the problem (7) by using θ_{t-1} . In case that the solver is unable to find a feasible solution, it is decided that $\tilde{\mathcal{X}}_t \neq \emptyset$, multiply θ_{t-1} by $1/2$, then we try to solve problem (7) again. This is our chosen approach. A drawback of this approach is that, depending on the scale of θ_t , the constrained optimization problem must be solved many times. We show that, by performing an appropriate preprocessing, a feasible solution can be obtained within two rounds of constrained optimization in the worst case. We give the details in Appendix E.1.

Selection of b . As stated in Sec. 4, the F-GP-UCB algorithm is valid for an arbitrary monotonically decreasing function b such that $t - \mathcal{P}(\mathcal{X}, \|\cdot\|_\infty, \theta_t b(t)/4) = \Theta(t)$. However, if the rate is slower than $1/t^\alpha$ ($\alpha \in (0, 1/d)$), for example, $b(t) = 1/\ln(t+1)$, then the dominant term of our regret upper bound can become $\mathcal{O}(1/\ln T)$ (See Theorem B.3 in Appendix B and Corollary F.1 in Appendix F). This intuitively means that an exponentially large sample is needed to obtain an arbitrarily small regret and is not the desirable behavior. In this paper, we use b in the form of $b(t) = 1/t^\alpha$ with $\alpha = 1/(2d)$ as a practical choice. (Additional discussion about α is given in Appendix E.2.)

Adaptive tuning of θ_0 . Although the choice of θ_0 does not affect the no-regret guarantee, it has a large impact on the practical behavior when the number of steps is small. For example, if θ_0 is set to a large value when the failure region F_c forms a relatively small region, then there is a risk of excessively avoiding the feasible region that should be searched. It is practically difficult to find a desirable choice of θ_0 as it depends on some conditions that are unknown to the user, such as the failure region F_c . In order to alleviate this issue while retaining the theoretical convergence, we use the strategy to adaptively choose θ_t based on the posterior standard deviation $\sigma_{t-1}(\mathbf{x}_t)$ of the observed points. We note that such a strategy using $\sigma_{t-1}(\mathbf{x}_t)$ to adaptively select an unknown parameter is also developed in [54] to set the unknown kernel hyperparameters for ordinary GP optimization.

⁴As with the standard GP-UCB, $\beta_t = \mathcal{O}(\sqrt{\gamma_{T;\mathcal{X}}})$ in F-GP-UCB, which leads to the regret of $\tilde{\mathcal{O}}(\gamma_{T;\mathcal{X}}/\sqrt{T})$. In Appendix D, we discuss the possibility of addressing this issue based on several existing works [48, 30].

Algorithm 4 in Appendix E.3 shows the pseudo-code of our modified strategy. In Algorithm 4, the user specifies the possible range for $\theta_t \in [\theta_{\min}, \theta_{\max}]$, the threshold for the posterior standard deviation $h_\sigma > 0$, the threshold for the step number $q \in \mathbb{N}_+$, and a scaling factor $w \in (0, 1)$. As the initial value, we set $\theta_0 = \theta_{\max}$. After that, if the observed point is lower than the posterior standard deviation for q consecutive number of times, θ_t is multiplied by w to lower its value, but no less than the minimum value θ_{\min} . Intuitively, suppose the posterior standard deviation is repeatedly excessively small. In that case, there exists a possibility that θ_0 was set to an excessively large value, which may lead to the unwanted exclusion of the feasible region from the search space. In this procedure, by setting θ_0 to a large value of θ_{\max} at the start, we expect it to get adjusted towards an appropriate neighborhood scale. In our simulation experiments, we set $w = 0.75$, $h_\sigma = 0.02$, $q = 3$, $\theta_{\min} = 0.0001$, and $\theta_{\max} = 0.5$. We emphasize that this procedure does not affect our convergence guarantee (Theorems F.2 in Appendix E).

Lastly, the pseudo-code, including all the considerations provided in this section, is described in Algorithm 5 of Appendix E.

6 Numerical experiments

In this section, we show the performance of F-GP-UCB through numerical experiments. The detailed settings of the experiments and additional results can be found in Appendices G and H, respectively. First, for comparison, we use EI [37] and GP-UCB [42] as the baseline algorithms which do not consider failures. In each of these algorithms, the GP model is constructed using only the successful observations; the failure observations are not used. We also compare F-GP-UCB with the EFI-GPC-EP and EFI-GPC-Sign algorithms that leverage GPC models [2]. EFI-GPC-EP is the algorithm that uses the classic GPC model with probit likelihood, and posterior approximation is based on Expectation Propagation (EP). EFI-GPC-Sign uses a variant of GPC proposed in [2]. Both EFI-GPC-EP and EFI-GPC-Sign choose the next input based on the posterior success probability of GPC and the EI value of f as in [2] (Details are in Appendix H).

In EFI-GPC-EP and EFI-GPC-Sign, as the prior for the latent function of GPC, we use zero-mean GP with Gaussian kernel $k_c(\mathbf{x}, \mathbf{y}) := \sigma_c^2 \exp(-\|\mathbf{x} - \mathbf{y}\|^2 / (2l_c^2))$. Before the experiment, we fix the kernel parameters of GPC σ_c and l_c by marginal likelihood maximization using observations at a randomly generated Sobol sequence. All the kernel hyperparameters of the GPC fixed by this procedure are given in Appendix H. Furthermore, it is well-known that the theoretical choice of β_t in GP-UCB is excessively conservative. We set $\beta_t = 2 \ln(2t)$ as in [24] for GP-UCB and F-GP-UCB. The other parameters in F-GP-UCB are fixed as described in Sec. 5. For the evaluation of the algorithms, we confirm the behavior of the regret at each step $r_t = f(\mathbf{x}^*) - f(\hat{\mathbf{x}}_t)$. We note that $\hat{\mathbf{x}}_t$ is the estimated solution $\hat{\mathbf{x}}_t = \mathbf{x}_{\hat{t}}$ where $\hat{t} = \operatorname{argmax}_{i \in \mathcal{I}_t^{(s)}} \operatorname{lcb}_t(\mathbf{x}_i)$, which is slightly modified from the theoretical value as discussed in Sec. 3. If we get that $\mathcal{I}_t^{(s)} = \emptyset$, then we set $r_t = f(\mathbf{x}^*) - \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$. This definition is the same as the utility gap metric for constrained GP optimization [19]. In each experiment, we report the average performance of 20 numbers of optimization trials with different seeds. In each trial, we generate one point uniformly at random over \mathcal{X} and use it as the initial point. Lastly, we add an artificial noise $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.0001$ to the observation of f . We also fix the noise variance hyperparameter of the GP model for f to 0.0001.

Synthetic experiments with the Branin function. We perform synthetic benchmark experiments with the Branin function whose input space is scaled to $[0, 1]^2$. The failure function used in these experiments is shown in Fig. 2. This failure function has a feasible region on the upper right side which is easily identifiable. In addition, the failure function also contains multiple small isolated feasible regions, one of which contains the optimal solution. We use the Gaussian kernel $k(\mathbf{x}, \mathbf{y}) := \sigma_f^2 \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2 / (2l_f^2))$ to construct a GP model of f . The parameters σ_f and l_f are fixed beforehand by marginal likelihood maximization over a Sobol sequence of 1024 points. The subsequent experiments also use the Gaussian kernel; and the parameters are fixed as described above.

Figure 2 shows the result. Under this benchmark setting, F-GP-UCB has a better performance. We also give the detailed behavior of F-GP-UCB in Appendix G. We note that, in EI and GP-UCB, the regret stops decreasing in the early stage. This is because these algorithms get stuck on the same failure point.

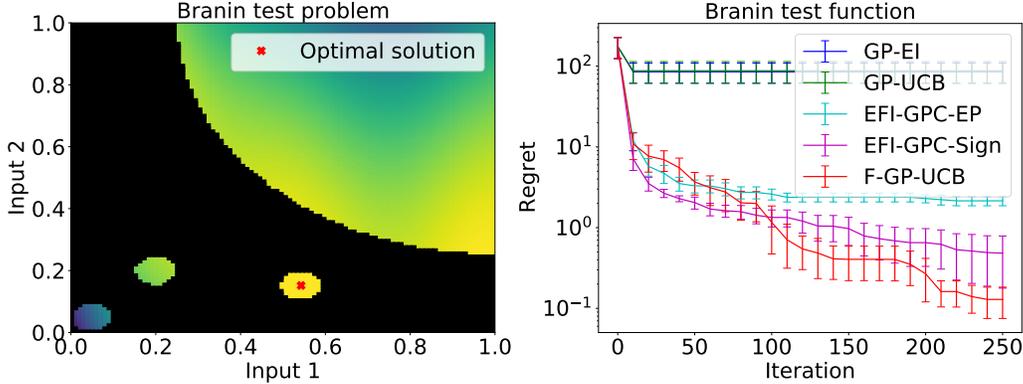


Figure 2: The left plot shows the Branin function whose input space is scaled to $[0, 1]^2$. The shaded regions represent the failure regions. The right plot shows the regret in the synthetic problem using the Branin function, where the average of 20 experiments with different random seeds is shown. The error bars correspond to two standard errors.

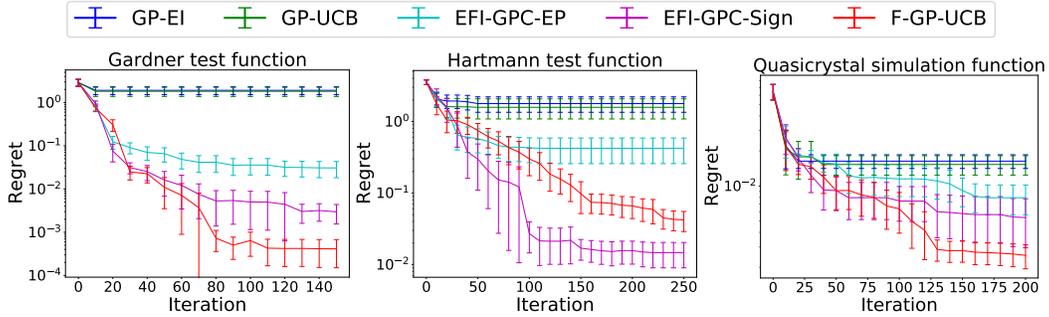


Figure 3: Experiment results of benchmark function of the constrained optimization problems (left and middle) and quasicrystal simulation function (right). The left and middle plots show the result of Gardner and Hartmann test functions, respectively. The error bars correspond to two standard errors.

Test functions for constrained optimization. We adopt the two benchmark settings which are used in existing GP-based constrained optimization literature. Gardner [10] is a 2D test problem whose objective and constraint functions are defined as a combination of sine and cosine functions. Hartmann [29] is a 3D test problem⁵ whose feasible region becomes a unit hypersphere. To adapt the constrained problem to our settings, the feasible region in each benchmark is defined as the success region, while the rest is defined as the failure region. Fig. 3 shows the result. Within the compared algorithms, F-GP-UCB has the best performance for the Gardner function, while EFI-GPC-Sign has the best performance for the Hartmann function. This can be understood from the fact that the feasible region of the Hartmann function, which is a unit hypersphere, and can be modeled easily by GPC. However, F-GP-UCB performs better compared to EFI-GPC-EP, which uses a classic GPC model. Moreover, the regret of F-GP-UCB continues to decrease as the number of iterations increases.

Numerical experiments with quasicrystals. We perform a numerical experiment involving quasicrystals in the Al-Cu-Mn system. We consider the optimization of the phonon thermal conductivity for quasicrystals. In this setting, the input space is the relative composition of the three elements, restricted to be around the feasible region of quasicrystal formation. The feasible region is the composition values that form quasicrystals based on data [16]. Outside of this region, quasicrystals do not form and so we consider it as a failure region, as the material property could not be probed with methods designed for quasicrystals. For the objective function, we use an empirical formula for the phonon thermal conductivity which we extract from [46]. The details of the setting are described in

⁵The original paper [29] considers a 6D problem and this setting leads to an excessively large failure region. We modify the setting by using a 3D version of the Hartmann function with unit hypersphere constraint.

Appendix H.2. In Fig. 3, the right plot shows the result of the numerical experiments for quasicrystals. It shows that F-GP-UCB performs the best among the compared algorithms.

7 Conclusions

In this paper, we propose a novel GP-based optimization algorithm in the presence of unknown failure regions in the search space. Our algorithm only requires a very mild assumption of the failure function that the optimal solution lies on an interior of a feasible region. We show that our algorithm achieves a convergence with high probability and provides the first regret upper bound by appropriately adjusting the search space. Its effectiveness is verified empirically through numerical experiments, including the heuristic simulation experiment motivated by the material research of quasicrystals.

Acknowledgements

This work was partially supported by RIKEN Center for Advanced Intelligence Project.

References

- [1] Yasin Abbasi-Yadkori. *Online learning for linearly parametrized control problems*. PhD thesis, University of Alberta, 2013.
- [2] François Bachoc, Céline Helbert, and Victor Picheny. Gaussian process optimization with failures: classification and convergence proof. *Journal of Global Optimization*, 78(3):483–506, 2020.
- [3] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:21524–21538, 2020.
- [4] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyperparameter optimization. In *Advances in Neural Information Processing Systems*, volume 24, 2011.
- [5] Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, pages 1–35, 2021.
- [6] Ilija Bogunovic, Jonathan Scarlett, Stefanie Jegelka, and Volkan Cevher. Adversarially robust optimization with Gaussian processes. *Advances in Neural Information Processing Systems*, 31, 2018.
- [7] Romain Camilleri, Kevin Jamieson, and Julian Katz-Samuels. High-dimensional experimental design and kernel bandits. In *International Conference on Machine Learning*, pages 1227–1237. PMLR, 2021.
- [8] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 844–853, 2017.
- [9] Thomas Desautels, Andreas Krause, and Joel W. Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research*, 15 (119):4053–4103, 2014.
- [10] Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. Bayesian optimization with inequality constraints. *Proceedings of the 31st International Conference on Machine Learning*, 32(2):937–945, 22–24 Jun 2014.
- [11] Michael A. Gelbart, Jasper Snoek, and Ryan P. Adams. Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI’14, page 250–259, Arlington, Virginia, USA, 2014. AUAI Press. ISBN 9780974903910.

- [12] Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1(2):141–149, 1992.
- [13] Susana Gomez and Alejandro Velasco Levy. The tunnelling method for solving the constrained global optimization problem with several non-connected feasible regions. In *Numerical Analysis*, pages 34–47. Springer, 1982.
- [14] Javier González, Joseph Longworth, David C James, and Neil D Lawrence. Bayesian optimization for synthetic gene design. *arXiv preprint arXiv:1505.01627*, 2015.
- [15] GPpy. GPpy: A Gaussian process framework in python. <http://github.com/SheffieldML/GPy>, since 2012.
- [16] Benjamin Grushko and SB Mi. Al-rich region of Al–Cu–Mn. *Journal of Alloys and Compounds*, 688:957–963, 2016.
- [17] PC Haarhoff and JD Buys. A new method for the optimization of a nonlinear function subject to nonlinear constraints. *The Computer Journal*, 13(2):178–184, 1970.
- [18] Philipp Hennig and Christian J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(57):1809–1837, 2012.
- [19] José Miguel Hernández-Lobato, Michael A. Gelbart, Ryan P. Adams, Matthew W. Hoffman, and Zoubin Ghahramani. A general framework for constrained Bayesian optimization using information-based search. *Journal of Machine Learning Research*, 17(1):5549–5601, 2016.
- [20] Jörg Hoffmann and Sebastian Kupferschmid. A covering problem for hypercubes. In *IJCAI*, pages 1523–1524, 2005.
- [21] Shogo Iwazaki, Yu Inatsu, and Ichiro Takeuchi. Mean-variance analysis in Bayesian optimization under uncertainty. In *International Conference on Artificial Intelligence and Statistics*, pages 973–981. PMLR, 2021.
- [22] Steven G. Johnson. *The NLOpt nonlinear-optimization package*, 2011. URL <http://ab-initio.mit.edu/nlopt>.
- [23] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.
- [24] Kirthevasan Kandasamy, Jeff Schneider, and Barnabas Poczos. High dimensional Bayesian optimisation and bandits via additive models. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 295–304, Lille, France, 07–09 Jul 2015.
- [25] Kirthevasan Kandasamy, Gautam Dasarathy, Junier Oliva, Jeff Schneider, and Barnabas Poczos. Multi-fidelity Gaussian process bandit optimisation. *Journal of Artificial Intelligence Research*, 66:151–196, 2019.
- [26] Johannes Kirschner, Ilija Bogunovic, Stefanie Jegelka, and Andreas Krause. Distributionally robust Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2174–2184. PMLR, 2020.
- [27] Shunya Kusakawa, Shion Takeno, Yu Inatsu, Kentaro Kutsukake, Shogo Iwazaki, Takashi Nakano, Toru Ujihara, Masayuki Karasuyama, and Ichiro Takeuchi. Bayesian Optimization for Cascade-Type Multistage Processes. *Neural Computation*, 34(12):2408–2431, 11 2022. ISSN 0899-7667. doi: 10.1162/neco_a_01550.
- [28] Madison Lee, Shubhanshu Shekhar, and Tara Javidi. Multi-scale zero-order optimization of smooth functions in an rkhs. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 288–293. IEEE, 2022.
- [29] Benjamin Letham, Brian Karrer, Guilherme Ottoni, and Eytan Bakshy. Constrained Bayesian optimization with noisy experiments. *Bayesian Analysis*, 14(2):495–519, 2019.

- [30] Zihan Li and Jonathan Scarlett. Gaussian process bandit optimization with few batches. In *International Conference on Artificial Intelligence and Statistics*, pages 92–107. PMLR, 2022.
- [31] Zihan Li and Jonathan Scarlett. Regret bounds for noise-free cascaded kernelized bandits. *CoRR*, abs/2211.05430, 2022.
- [32] David V Lindberg and Herbert KH Lee. Optimization under constraints by applying an asymmetric entropy measure. *Journal of Computational and Graphical Statistics*, 24(2):379–393, 2015.
- [33] Chang Liu, Erina Fujita, Yukari Katsura, Yuki Inada, Asuka Ishikawa, Ryuji Tamura, Kaoru Kimura, and Ryo Yoshida. Machine learning to predict quasicrystals from chemical compositions. *Advanced Materials*, 33(36):2102507, 2021.
- [34] Daniel J Lizotte, Tao Wang, Michael H Bowling, Dale Schuurmans, et al. Automatic gait optimization with Gaussian process regression. In *IJCAI*, volume 7, pages 944–949, 2007.
- [35] Ruben Martinez-Cantin, Nando de Freitas, Arnaud Doucet, and José A Castellanos. Active policy learning for robot planning and exploration under uncertainty. In *Robotics: Science and systems*, volume 3, pages 321–328, 2007.
- [36] Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 362–369, 2001.
- [37] Jonas Moćkus. On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404, 1975.
- [38] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001.
- [39] Matthieu Sacher, Régis Duvigneau, Olivier Le Maitre, Mathieu Durand, Elisa Berrini, Frédéric Hauville, and Jacques-André Astolfi. A classification approach to efficient global optimization in presence of non-computable domains. *Structural and Multidisciplinary Optimization*, 58(4): 1537–1557, 2018.
- [40] Sudeep Salgia, Sattar Vakili, and Qing Zhao. A domain-shrinking based bayesian optimization algorithm with order-optimal regret performance. *Advances in Neural Information Processing Systems*, 34:28836–28847, 2021.
- [41] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25, 2012.
- [42] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1015–1022, 2010.
- [43] Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with Gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 997–1005, Lille, France, 07–09 Jul 2015.
- [44] Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe Bayesian optimization with Gaussian processes. In *International conference on machine learning*, pages 4781–4789. PMLR, 2018.
- [45] Krister Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12(2):555–573, 2002.
- [46] Yoshiki Takagiwa, Ryota Maeda, Satoshi Ohhashi, and An-Pang Tsai. Reduction of thermal conductivity for icosahedral Al-Cu-Fe quasicrystal through heavy element substitution. *Materials*, 14(18):5238, 2021.

- [47] Shion Takeno, Tomoyuki Tamura, Kazuki Shitara, and Masayuki Karasuyama. Sequential and parallel constrained max-value entropy search via information lower bound. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20960–20986, 17–23 Jul 2022.
- [48] Sattar Vakili, Nacime Bouziani, Sepehr Jalali, Alberto Bernacchia, and Da-shan Shiu. Optimal order simple regret for gaussian process bandits. *Advances in Neural Information Processing Systems*, 34:21202–21215, 2021.
- [49] Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in Gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR, 2021.
- [50] Sattar Vakili, Jonathan Scarlett, and Tara Javidi. Open problem: Tight online confidence intervals for RKHS elements. In *Conference on Learning Theory*, pages 4647–4652. PMLR, 2021.
- [51] Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nello Cristianini. Finite-time analysis of kernelised contextual bandits. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 654–663, 2013.
- [52] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.
- [53] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3627–3635, 06–11 Aug 2017.
- [54] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Freitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.