machine learning & knowledge extraction

Article

When to Explore and When to Exploit: Adaptive Decisions in Bayesian Optimization

Antonio Candelieri [1], Francesco Archetti [2] and Iman Seyedi [2,*]

[1] Department of Economics Management and Statistics, University of Milano-Bicocca, 20126 Milan, Italy; antonio.candelieri@unimib.it
[2] Department of Computer Science Systems and Communication, University of Milano-Bicocca, 20126 Milan, Italy; francesco.archetti@unimib.it
* Correspondence: seyediman.seyedi@unimib.it

Abstract

Gaussian process-based Bayesian optimization (BO) is a sample-efficient sequential strategy for optimizing expensive black-box functions. The Gaussian process provides a probabilistic approximation of the unknown function, while an acquisition function balances exploration and exploitation to select the next evaluation point. Despite significant research efforts, no master acquisition function has been identified. This paper proposes a novel adaptive acquisition function that dynamically adjusts the exploration–exploitation trade-off based on the evolution of the optimization process, rather than using fixed or random scheduling. While implemented here within a GP-based BO framework, the core switching mechanism is surrogate-agnostic: the exploitative component requires only a surrogate point prediction, and the explorative component is entirely model-free. Unlike traditional approaches, where mechanisms like UCB/LCB lean toward exploration over iterations, or fixed strategies that switch from exploratory (EI) to exploitative (PI) behavior at predetermined points, the proposed method makes purely exploitative decisions using only the GP's prediction. However, it discards these decisions when they have low potential for significant improvement, instead focusing on uncertainty reduction. Notably, this approach uses inverse distance weighting for uncertainty quantification rather than the GP's predictive uncertainty, avoiding bias from the GP's predictions. Testing on benchmark functions demonstrates that the proposed acquisition function is almost always Pareto optimal, offering the most balanced trade-off between convergence to the global optimum and exploration capability compared to state-of-the-art alternatives.

Keywords: Bayesian optimization; Gaussian process; acquisition function; confidence bound

Academic Editor: Luke E. K. Achenie

Received: 29 April 2026
Revised: 25 June 2026
Accepted: 01 July 2026
Published: 3 July 2026

1. Introduction

Bayesian optimization (BO) [1–3] is the most widely adopted method for globally optimizing black-box, expensive, and multi-extremal functions. Thanks to its sample efficiency, it has been successfully applied to an extremely wide range of real-life applications, such as engineering [4–6], optimal control of complex systems [7–10], and automated machine learning [11–16]. Beyond its use in AutoML, Bayesian optimization belongs to the broader machine learning paradigm of sequential decision-making under uncertainty, where the exploration–exploitation trade-off addressed in this paper is a foundational challenge shared with reinforcement learning and active learning.

The basic BO algorithm is a sequential learning-and-optimization framework, based on two components: a probabilistic surrogate model, approximating the objective function depending on all the solutions evaluated so far (i.e., the learning component of BO), and an acquisition function (also known as an infill criterion or utility function), driving the choice of the next solution to evaluate (i.e., the optimization component), depending on the current probabilistic surrogate model. The basic BO algorithm is schematically reported in Algorithm 1.

Algorithm 1 Basic Bayesian optimization algorithm

Data:
$\Omega \subset \mathbb{R}^d$, the usually box-bounded search space.
$\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$, an initial set of n observations, with $x_i, \ldots, x_n$ usually from Latin hypercube sampling (LHS) and $y_i = f(x_i) + \varepsilon_i$, $\varepsilon_i \sim N(0, \lambda^2)$;
$N > n$.
while $n < N$ do
fit the probabilistic surrogate model $\mathcal{M}$ on $\mathcal{D}$;
obtain x_{n+1} as the optimizer of the acquisition function $\mathcal{A}_M(x)$; evaluate x_{n+1} and observe $y_{n+1} = f(x_i) + \varepsilon_i$;
$\mathcal{D} \leftarrow \mathcal{D} \cup \{x_{n+1}, y_{n+1}\}$;
$n \leftarrow n + 1$
end
Result:
(x^+, y^+): $y^+ = min\{y_i : (x_i, y_i) \in \mathcal{D}\}$ ("max" in the case of a maximization problem.)

During the last decade, research has focused on two topics: (a) extending the set of modeling techniques so that the most appropriate probabilistic surrogate model can be chosen with respect to the specific features of the target optimization problem, and (b) defining acquisition functions to effectively deal with the exploration–exploitation dilemma. This paper focuses on the second topic only and considers the Gaussian process (GP) regression [17,18] as a probabilistic modeling technique. This is not a limitation because the probabilistic surrogate model and the acquisition function can be independently chosen, without affecting the overall BO framework, as proven by all the widely known BO software libraries [19–23]. Moreover, although GP regression is well-suited for modeling functions over continuous search spaces, extensions have been recently proposed to also deal with more complicated ones [24–26].

As far as the acquisition function is concerned, the alternatives proposed over time can be conveniently divided into improvement-based and information-based acquisition functions [27,28]. While the former are devoted to searching for the optimum (i.e., the best objective function's value), the latter search for the optimizer (i.e., the solution providing the best objective function's value). Although apparently irrelevant, this distinction leads to completely different strategies, with information-based acquisition functions usually offering a greater sample efficiency but entailing a higher computational cost to determine the next solution to evaluate. From a theoretical perspective, it is important to note that a very recent work [29] shows that popular acquisition functions—improvement as well as information-based—are just special cases of a generalization of the Shannon entropy, from statistical decision theory.

This paper considers the improvement-based acquisition functions only, which combine the model's prediction and the associated predictive uncertainty to implement their own exploration–exploitation trade-off mechanisms. The common underlying issue con-

https://doi.org/10.3390/make8070193

sists of avoiding decisions exclusively based on the model's prediction because this would exclusively lead to local search, with the risk of remaining trapped in local optima. Thus, every improvement-based acquisition function includes an uncertainty bonus to guarantee global search and, consequently, convergence to the global optimum. The most recent research studies have proposed prefixed (usually increasing) or random scheduling of the uncertainty bonus. However, results are inconclusive, with every proposed approach beating the previous ones only on a subset of test problems, and often entailing one or more hyperparameters to be manually fine-tuned for every specific problem.

Contribution: Here, we summarize the main contribution of this paper. First, we have identified the main limitations of the current improvement-based acquisition functions:

- Both prefixed and random scheduling do not consider the impact of every sequential decision on the model and, consequently, the decision suggested by the acquisition function. On the contrary, we prove that this information is crucial for the successive decision.
- As already proven in [30,31], the decisions generated by improvement-based acquisition functions are Pareto rational with respect to two objectives: optimizing according to the model's prediction (i.e., exploitation) and maximizing the predictive uncertainty (i.e., exploration). Varying the uncertainty bonus translates into a move on the Pareto front only.
- On the contrary, human search is characterized by the chance to also make non-Pareto rational decisions, with respect to the same two objectives [32,33]. Moreover, the number of non-Pareto rational decisions does not follow any predefined or random scheduling, but it seems to be triggered by the impact of every decision.
- Improvement-based acquisition functions rely on the predictive uncertainty of the probabilistic surrogate model. However, it is biased by the prediction itself, especially when GP regression is used. Even more critical, variance starvation (i.e., predictive uncertainty close to zero) may occur with GP regression, nullifying any uncertainty bonus.
- As a consequence of these considerations, we propose a novel acquisition function that is able to adaptively provide exploitative or exploratory decisions, depending on the evolution of the sequential optimization process, instead of using a random or prefixed scheduling for the uncertainty bonus. More specifically, our master acquisition function performs as follows:
- It uses only the prediction of the GP model to make its decision about the next solution to evaluate (purely exploitative decision).
- It discards that decision if it has a small chance to significantly improve the current best solution (as explained in Section 3) and, instead, makes a different decision aimed at uncertainty reduction. To do this, we do not use the GP's predictive uncertainty but a different uncertainty quantification measure, namely the inverse distance weighting (IDW) (also introduced in Section 3), which is not biased by the GP's prediction.

In this paper, a number of state-of-the-art acquisition functions have been considered and applied to a number of test problems. We have proposed a Pareto analysis of the effectiveness of these acquisition functions, based on two metrics: convergence to the optimum and capacity to explore. This could become a more mathematically sound and general method to compare the acquisition functions for BO.

Finally, it is worth noting that the aim of this paper is not to be a survey or review of the topic.

https://doi.org/10.3390/make8070193

Related Work

Here, we summarize the most related works, to our knowledge, and their main contributions, achievements, and limitations. In ref. [34], the first proof of convergence of the well-known GP confidence bound (GP-CB) acquisition function is provided. Basically, the uncertainty bonus is logarithmically increased with the number of function evaluations in order to increase exploration and reduce the chance of getting stuck in local optima. More recently, the authors in [35] have empirically shown that a random scheduling of the bonus uncertainty (i.e., sampling from a Gamma distribution whose parameters are adjusted depending on the number of function evaluations) can outperform the previous scheduling on several test problems. In ref. [31], a Pareto analysis of every decision implied by traditional acquisition functions is performed, where the two objectives are the prediction of the GP model and the associated predictive uncertainty. As a result, varying the value of the uncertainty bonus translates into moving onto the Pareto front; thus, GP-CB only allows for Pareto decisions, regardless of the scheduling of the uncertainty bonus. According to this, ref. [30] proposed a new mechanism based on the ε-greedy strategy typically adopted in reinforcement learning. With probability $(1 - \varepsilon)$, a greedy decision is taken (i.e., the solution optimizing the GP's prediction), while a random choice is performed with probability ε. Two alternatives are possible in this case: randomly selecting among Pareto rational decisions or uniformly at random over the entire search space. This is interesting because in [32,33] it has been empirically proved that humans usually make non-Pareto-rational decisions when the chance to improve is small. Thus, while the ε-greedy method relies on the probability ε—that is, a hyperparameter to set before the optimization—humans can adaptively switch between Pareto and non-Pareto decisions along the optimization process. Finally, we would like to also cite a very recent work considering the alternation between two different acquisition functions, that is, expected improvement (EI) and probability of improvement (PI). Indeed, these two acquisition functions offer two different exploration–exploitation trade-offs, with PI more biased toward exploitation. More precisely, ref. [36] propose two schemes: alternating between PI and EI (aka round-robin) or starting with EI and then switching to PI, after a given number of iterations (i.e., a hyperparameter to be set before the optimization process).

The topic that has attracted growing attention is Bayesian optimization in high-dimensional (HD) spaces. The performance of BO degrades significantly when the dimension exceeds 20. The issue of HD has been addressed in [37,38] and more recently [39]. Recent results stress the importance of reconsidering 'classic approaches', with several studies demonstrating that vanilla Bayesian optimization methods can be surprisingly effective in high-dimensional settings, challenging the assumption that specialized techniques are always necessary for such problems [40–44].

Bayesian optimization in non-Euclidean settings has been studied to address optimization problems where the search space possesses complex geometric structures beyond traditional continuous domains. Early work by [45] introduced BO for combinatorial structures, while more recent advances have extended these methods to network and manifold geometries. Ref. [46] proposes a new approach to ensure that the surrogate model is tailored to the network geometry, providing a convenient framework for probabilistic modeling of functions on metric graphs that can be represented using finite elements to obtain a sparse approximation of the inverse covariance for updating the posterior surrogate model. This approach is tested with two acquisition functions: improved GP upper confidence bounds (IGPUCB) and GP-TS Thompson sampling. Related work on optimization over manifolds via graph Gaussian processes has been developed by [47], while [48] investigates Bayesian optimization with inexact acquisition functions, and [49] explores enhancing Gaussian process surrogates through random exploration. Ref. [50] further contributes to this area with global optimization on graph-structured data via

Gaussian processes with spectral representations. A specific area of BO on discrete search domains focuses on permutations, where specialized kernels and acquisition functions are needed. Ref. [51] introduced the acquisition weighted kernel for batch Bayesian optimization on permutations, while ref. [52] developed general frameworks for Bayesian optimization over permutation spaces. More recent work by [53] proposes scalable kernels derived from sorting algorithms for high-dimensional permutation spaces, ref. [54] introduces relational Bayesian optimization for permutations, and ref. [55] develops the merge kernel for Bayesian optimization on permutation spaces. More broadly, the notion of a surrogate variable appears in other areas of statistical inference, such as residual diagnostics for ordinal regression [56] and partial association analysis for mixed data [57], although these uses are conceptually distinct from the surrogate models employed in Bayesian optimization.

As already mentioned, the research on acquisition functions for BO is quite impressive. The information-based acquisition functions are not directly related, specifically, entropy search (ES) [58], predictive entropy search (PES) [59], max-value entropy search (MES) [60], and [61]. These methods are based on sampling from the GP posterior—usually via Thompson sampling—that is, the operation leading to their expensive computational cost. The issue of efficiently sampling from the GP posterior has been recently addressed in [62]. Moreover, all the look-ahead acquisitions, including the knowledge gradient (KG) [63], are also out of scope. Finally, the trust-region-based approaches, such as TRIKE [64], TREGO [65], and TuRBO [66] are outside the scope of our method.

2. Background

2.1. GP-Based BO

Assume that we are at a generic iteration n, and all the observations collected so far are stored in the dataset $\mathcal{D} = \{x_i, y_i\}$, with $|\mathcal{D}| = n$. To simplify the notation in the following, we rewrite the dataset as $\mathcal{D} = \{X, y\}$, where $X = x_1, \ldots, x_n$ and $y = y_1, \ldots, y_n$. Recall from Algorithm 1 that $y_i = f(x_i) + \varepsilon_i$, with $f : \mathbb{R}^d \rightarrow \mathbb{R}$ the objective function, possibly noisy (i.e., $\varepsilon^i \sim N(0, \lambda^2)$).

A GP regression model is used to approximate the black-box, expensive, and multiextremal objective function, depending on the dataset D. GP regression is a well-known kernel method [67,68], in which covariance between observations is codified through a kernel function, $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, which can be chosen among many alternatives [17,18]. In this paper, we consider the squared exponential (SE) kernel, $k(x, x') = \sigma_f^2 e^{\frac{-\|x-x'\|^2}{2\ell^2}}$, which assumes the smoothest approximation for the objective function. It is usually a reasonable choice in the case of experimental settings like that considered in this paper, while other, less smooth kernels can be more appropriate when addressing real-life problems.

A GP regression model is also said to be probabilistic, meaning that for any given input, it provides both a prediction and the associated (predictive) uncertainty, respectively given by [17]:

$$\mu(x) = k(x, X)\,[K + \lambda^2 I]^{-1}\, y \tag{1}$$

$$\sigma^2(x) = k(x, x) - k(x, X)[K + \lambda^2 I]^{-1}\, k(X, x) \tag{2}$$

where $k(x, X)$ is an n-dimensional vector such that its i_{th} component is $k(x, x_i)$, $k(X, x)$ is the transposed vector, and K is the $n \times n$ kernel matrix with entries $K_{i,j} = k(x_i, x_j)$. Finally, λ^2 is used to deal with noisy objective functions as well as to ensure that the matrix inversion operation can be performed.

https://doi.org/10.3390/make8070193

Fitting a GP model on the dataset $\mathcal{D}$ means conditioning the predictive mean $\mu(x)$ and the predictive uncertainty $\sigma(x)$—i.e., the square root of Equation (2)—to the available dataset D, that is, tuning the kernel's hyperparameters (i.e., σ_f^2 and ℓ in the SE kernel) and in case λ^2, usually via MLE or MAP. Equations (1) and (2) should indeed be intended as $\mu(x)^n$ and $\sigma^2(x)^n$, but we omit the suffix to keep the notation as simple as possible.

The resulting GP is the model generically denoted with $\mathcal{M}$ in Algorithm 1, on which the acquisition function $\mathcal{A}_{\mathcal{M}}(x)$ is based. According to the basic BO algorithm, the next solution to evaluate is:

$$x_{n+1} \in arg\ opt_{x \in \Omega}\ \mathcal{A}_{\mathcal{M}}(x) \quad (3)$$

Although most of the acquisition functions are expressed in terms of utility and are, consequently, maximized, some of them are expressed in terms of an optimistic estimate of the objective function; indeed, they are maximized or minimized according to the original optimization problem. For this reason, we use the general notation "optimize" in Equation (3).

2.2. Improvement-Based Acquisition Functions

Instead of following the chronological order, we start by introducing the GP-based confidence bound (GP-CB) acquisition function. It is also known as an optimistic policy because it provides the most optimistic approximation of the objective function, depending on M. More specifically, the lower confidence bound (GP-LCB) is used for minimization problems, and it is defined as [34]:

$$LCB_n(\mathrm{x}) = \mu(\mathrm{x}) - \sqrt{\beta_n}\, \sigma(\mathrm{x}) \quad (4)$$

In the case of a maximization problem, the upper confidence bound (GP-UCB) is obtained by simply replacing the difference with the sum. The role of the uncertainty bonus—represented by the predictive uncertainty $\sigma(x)$—is clear, as well as that of the hyperparameter β: The GP-LCB addresses the exploitation–exploration trade-off by scalarizing the two objectives (i.e., exploitation and exploration).

Most of the research focused on how to choose a suitable value for β over BO iterations. Here, we summarize the most relevant methods (against which we have compared our approach).

- Ref. [34] Theorem 1: The search space Ω consists of a finite number of choices. It is always possible to consider this case by explicitly taking into account the numerical precision used to represent solutions. The proposed scheduling for β is given by:

$$\beta_n = 2 \log\left(|\mathcal{G}| n^2 \pi^2 / (6\delta)\right) \quad (5)$$

with $|\mathcal{G}|$ the number of finite solutions in Ω, and with $\delta \in (0,1)$ such that the probability that the regret is sub-linear is not lower than $1 - \delta$. As is well known, regret is one of the metrics used to analyze the convergence of global optimization algorithms. It is briefly recalled in the following Section 2.3.

- Ref. [34] Theorem 2: The search space Ω is continuous, with an infinite number of solutions. In this case, the proposed scheduling for β is:

$$\beta_n = 2 \log\left(2n^2\pi^2/(3\delta)\right) + 2d \log\left(n^2 dbr \sqrt{(\log(4da)/\delta))}\right) \quad (6)$$

with $r : \Omega = [0, r]^d$, and $a, b > 0$ some constants to obtain a sub-linear regret with probability $1 - \delta$.

- The strongest criticism of the previous scheduling was noted in [69]: "the selection of β is not done to optimally balance exploration and exploitation, but is done such that the cumulative regret is bounded. While the regret bound provided is desirable, it is far larger than needed. This leads to sub-optimal real-world performance due to over-exploration". Indeed, in [34], the authors empirically divide β by 5 just to achieve better results. In contrast, the randomized GP-CB proposed in [69] samples smaller β values from a Gamma distribution whose parameters are set to ensure sub-linear convergence. Specifically:

$$\beta^n \sim \Gamma(\kappa^n, \theta) \tag{7}$$

with the shape parameter $\kappa^{(n)} = \frac{log\left(\frac{n^2+1}{\sqrt{2\pi}}\right)}{\log(1+\frac{\theta}{2})}$. In contrast, the scale parameter θ has to be fine-tuned for each target problem. Basically, increasing θ will increase exploration.

Other approaches are not based on the GP-CB and, instead, propose to switch—randomly or according to a predefined scheme—between different selection mechanisms. The approach proposed in [30] was inspired by the well-known ε-greedy strategy adopted in reinforcement learning. Specifically, two alternative methods have been proposed:

- ε −PF selection: In this case, the next decision x^{n+1} is the optimizer of $\mu(x)$, with probability $1 - \varepsilon$, or randomly selected from the Pareto front (PF) defined over the two objectives $\mu(x)$ and $\sigma(x)$, with probability ε. Recalling results from [31], this means using the GP-CB with a randomly selected value for β.
- ε-RS selection: In this case, the next decision x^{n+1} is the optimizer of $\mu(x)$, with probability $1 - \varepsilon$, or uniformly selected at random within the entire search space Ω, with probability ε.

Similarly, [36] have recently proposed a selection mechanism which does not rely on the GP-CB but, contrary to [30], their algorithm switches between two well-known acquisition functions, specifically PI and EI [70,71]:

$$PI\ (x) = \Phi\left(\frac{\Delta(x)}{\sigma(x)}\right) \tag{8}$$

$$EI\ (x) = \begin{cases} \Delta(x)\Phi\left(\frac{\Delta(x)}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{\Delta(x)}{\sigma(x)}\right) & \text{if } \sigma(x) > 0 \\ 0 & \text{if } \sigma(x) = 0 \end{cases} \tag{9}$$

where $\Delta(x) = \min_{i=1,\ldots,n}\{y_i\} - \mu(x)$ if the target problem is a minimization problem (otherwise $\Delta(x) = \mu(x) - \max_{i=1,\ldots,n}\{y_i\}$), and $\Phi(x)$ and $\phi(x)$ are the standard Gaussian cumulative distribution function and probability density function, respectively. We have reported here the two formulations adopted by the authors in [36]. However, it is important to note that some modifications have been proposed in the literature to increase exploration, especially for PI.

Moreover, two mechanisms have been proposed and investigated by the authors: alternating (aka round-robin) between the two acquisition functions and starting with EI to finally switch to PI when a certain percentage of the overall evaluations is reached. It is important to note that the latter scheduling is the opposite of that in [34]: it is more explorative at the beginning and more exploitative at the end, while the logic underlying the scheduling proposed in [34] increases the chance for exploration with the BO iterations.

2.3. The Disregarded Relevance of the Discrepancy

Regret as a convergence metric: As previously mentioned, regret is typically used as a natural performance metric for global optimization algorithms. The instantaneous regret at iteration n, associated with the decision $x^{(n)}$, is $r_n = y^* - y_n$, with $y^* = f(x^*) + \varepsilon$. The cumulative regret, up to iteration N, is $R_N = \sum_{n=1}^{N} r_n$. A desirable property is

$\lim_{N\to\infty} R_N = 0$ (i.e., no-regret). For finite N, the value R_N/N translates into a convergence rate. The critical issue is that over-exploration (as implied by the scheduling of β in [34]) increases the value R_N/N to avoid getting stuck in local optima, while other methods decrease it, but at the cost of a higher chance of achieving a sub-optimal solution.

Moreover, in a global optimization task, minimizing the cumulative regret can be counterintuitive from a human search point of view. If y^* is achieved after a few iterations, a human searcher will not continue to evaluate solutions close to the current best one (even if they do not know the value of y^*, a priori). Instead, they will perform evaluations to reduce uncertainty over the search space. This is because the task that they are solving is searching for the best, instead of accumulating some reward. It is the feeling about the impossibility of improving that triggers pure exploration in a human searcher, but this behavior is not modeled by traditional acquisition functions [32,33].

At the very end, an acquisition function is effective if it provides a balanced trade-off between exploration and exploitation over the entire optimization process. It is important to note that Pareto optimality, as used in this paper, means non-dominated with respect to the two selected metrics (AGAP and L2-discrepancy). A method can be Pareto optimal by being extreme in one objective, for example, excellent at convergence but poor at exploration, or vice versa. A method located in the central part of the Pareto front, with neither objective sacrificed, is generally preferable in practice when both exploration and convergence matter equally.

GAP metric: To better explain the previous statement, we first introduce the GAP metric, another common performance indicator to compare optimization algorithms.

$$GAP_n = \frac{y_0^+ - y^{+(n)}}{|y_0^+ - y^*|} \tag{10}$$

with $y_0^+ = best\left\{y_1, \ldots, y_{n_0}\right\}$ and $y_n^+ = best\left\{y_{n_0+1}, \ldots, y_n\right\}$o (with "best" replaced with "max" or "min" according to the original optimization problem; the absolute value is adopted to independently deal with both maximization and minimization). The GAP metric has important features: it is defined in [0, 1], so it can also be used to compare different approaches on completely different optimization problems, and it allows us to clearly identify at which iteration the best solution has been identified. However, it is not directly linked to R_N/N. Consider Figure 1, which is an anticipation of the results presented in the next section. The chart on the left depicts the GAP curves (i.e., GAP_n with $n = n_0, \ldots, N$) for BO using different acquisition functions. More precisely, each curve is the average GAP curve on 100 independent runs. Four of them achieve, almost immediately, the GAP value equal to 1 (i.e., CB, epsPF, epsRS, and mastering), but do not provide any information about the successive decisions and, consequently, the value of R_N/N. However, at the end of each optimization process, one can observe how many acquisition functions have used the remaining trials to explore, similar to a human searcher, instead of exploiting without any further improvement. To quantify the capacity of exploration, we use discrepancy [72], from the design of the experiments.

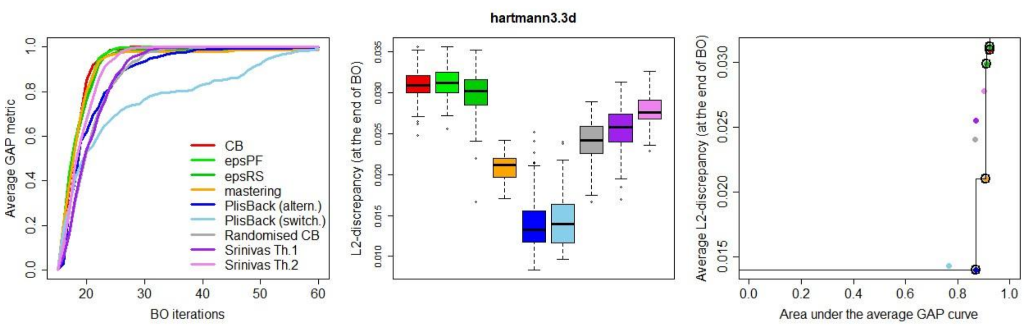

Figure 1. Comparing BO based on different improvement-based acquisition functions: (**left**) GAP metric curves averaged on 100 independent runs, (**middle**) L2-discrepancy at the end of the optimization process and over 100 independent runs, (**right**) Pareto analysis between area under the average GAP metric curve and L2-discrepancy.

Discrepancy measures how far a given distribution of points deviates from a perfectly uniform one. Many discrepancies have been proposed; we have decided to consider the so-called L2-discrepancy because it can be expressed analytically even for high dimensions. Formally [72]:

$$D_{L2}(\mathrm{X}) = \left[\int_{[0;1]^{2d}} \left(\frac{A(\mathrm{X}; J_{a,b})}{n} - Vol(J_{a,b})\right)^2 dadb\right]^{1/2} \quad (11)$$

where $Vol(J_{a,b})$ denotes the volume of a subset $J_{a,b} \subset [0;\ 1]^d, A(X;\ J_{a,b})$ is the number of points of X falling into $J_{a,b}$, and finally $a, b \in [0;\ 1]^d$ and $J_{a,b} = [a_1, b_1) \times \cdots \times [a_d, b_d)$. With respect to solutions generated by BO, the lower the discrepancy, the higher the exploration.

Coming back to Figure 1, the middle chart shows that only one of the four acquisition functions with the best GAP curves has a low discrepancy, averaged on 100 independent runs (i.e., the so-called "mastering").

Pareto analysis: exploration–exploitation balance: To finally evaluate the effectiveness of the searching process implied by any acquisition function in terms of both convergence to the optimum and exploration capability, we suggest performing a Pareto analysis. While it is natural to obtain a single value for the discrepancy (i.e., the average over 100 independent runs), we also need to condense a GAP curve into a single scalar value. We suggest computing the area under the GAP curve—analogously to the AUC, the area under the ROC curve used to compare machine learning algorithms—that is:

$$A_{GAP} = \frac{1}{N - n_0} \sum_{n=n_0+1}^{N} GAP_n \quad (12)$$

It is important to note that the higher the A_{GAP}, the better the GAP curve. Returning to Figure 1, on the right, every point represents a different acquisition function, providing a different pair of A_{GAP} (to be maximized) and L2-discrepancy (to be minimized). Only some acquisition functions are Pareto optimal (i.e., circled ones), with just one in the middle of the PF, representing the most well-balanced trade-off between exploration and exploitation.

This analysis will be the basis of the results presented in this paper. The main benefit is that the effectiveness of a set of acquisition functions can be evaluated in terms of the exploration–exploitation balance over the entire optimization process, instead of the single decision.

3. An Acquisition Function Mastering the Trade-Off Between Exploration and Exploitation

Here, we describe our proposed acquisition function, able to implement a BO process that is well-balanced in terms of exploration and exploitation, overall. Our acquisition function is based on the idea that the most effective and efficient exploitation–exploration trade-off mechanism does not follow any prefixed or random scheduling: it depends on the evolution of the optimization process, just like in human search. Starting from this consideration, we can state that, after a solution is evaluated, only one of the following three possible kinds of feedback can be experienced:

(1) x_{n+1} leads to an y_{n+1} in line with the expectation → happy but not surprised.
(2) x_{n+1} leads to an y_{n+1} better than expected → surprisingly happy.

https://doi.org/10.3390/make8070193

(3) x_{n+1} leads to an y_{n+1} worse than expected → unexpectedly poor.

Now we try to map these three situations in terms of implied modifications to the model $\mathcal{M}$ and, consequently, the decision suggested by the acquisition function $\mathcal{A}_{\mathcal{M}}(x)$.

(1) $\mathcal{M}$ will not significantly change, so the next decision will be biased toward exploitation (i.e., local search) if the uncertainty bonus is not large enough.
(2) Although we have collected a solution better than expected, this means that our model was not so accurate; the updated $\mathcal{M}$ could be significantly different from the previous. As a result, the next decision could be far away from the last evaluated solution, so some exploration (i.e., global search) is triggered by our last decision, regardless of the uncertainty bonus.
(3) The only difference with the previous case is that we have collected a solution worse than expected, but $\mathcal{M}$ will change anyway, triggering some exploration with respect to the last decision.

To summarize, if $\mathcal{M}$'s predictions are inaccurate with respect to the last decision, then the next decision could be explorative, meaning that it is far from the previous. On the contrary, if $\mathcal{M}$ is locally accurate, then the next decision will be biased toward an exploitative decision, and the uncertainty bonus—if not sufficiently large—just ensures searching within a neighborhood of the current best. Indeed, it is important to remark that M might fail to accurately approximate the objective function in promising regions because observations are sparse in those regions. This is the main motivation underlying the scheduling of β, in GP-CB, initially proposed by [34].

Algorithm 2 summarizes our approach. Basically, it works as follows: a pure exploitative decision is obtained by searching for the optimum of the GP's predictive mean, $\mu(x)$. Then the usefulness of this decision is considered: if it belongs to the neighborhood of the current best solution and this neighborhood already contains a sufficient number of solutions, then the decision is discarded and replaced by a decision aimed at minimizing the overall uncertainty, trying to cover unexplored regions of the search space.

Algorithm 2 Mastering the exploration–exploitation trade-off in BO

Data:
$\Omega = [0;\ 1]^d$ the search space (rescale the original search space in case);
$\mathcal{D} = \{x_i, y_i\}$, an initial set of n observations from LHS, and $y_i = f(x_i) + \varepsilon_i$, with $\varepsilon_i \sim N(0, \lambda^2)$;
$N_1 > N_2 : N_1 + N_2 = N > n$.
while $n < N_1$ do
(x^+, y^+): $y^+ = best\ y_i : (x_i, y_i) \in \mathcal{D}$;
$\mathcal{H}_w(x^+)$ is a hypercube centered in x^+ with a side equal to w;
Fit the GP model on $\mathcal{D}$ and obtain $\mu(x)$ and $\sigma(x)$;
$x_{n+1} = \underset{x \in \Omega}{\text{optimize}}\ \mu(x)$ ◁ exploitative decision;
if $x_{n+1} \in \mathcal{H}_w(x^+) \wedge |\{x_i \in X : x_i \in \mathcal{H}_w(x^+)\}| \geq \eta$ then
    $x_{n+1} = \underset{x \in \Omega}{\text{optimize}}\ z(x)$ ◁ *explorative decisionn*
end
observe $y_{n+1} = f(x_{n+1}) + \varepsilon_{n+1}$;
$\mathcal{D} \leftarrow \mathcal{D} \cup \{x_{n+1}, y_{n+1}\}$;
$n \leftarrow n + 1$
end
*** final refining: only exploitation ***
while $n < N_1 + N_2$ do
fit the GP model on $\mathcal{D}$ and obtain $\mu(x)$ and $\sigma(x)$;

$x_{n+1} = \underset{x \in \Omega}{\text{optimize}}\ \mu(x)$
observe $y_{n+1} = f(x_{n+1}) + \varepsilon_{n+1}$;
$\mathcal{D} \leftarrow \mathcal{D} \cup \{x_{n+1}, y_{n+1}\}$;
$n \leftarrow n + 1$
end
Result:

$$(x^+, y^+): y^+ = best\,\{y_i: (x_i, y_i) \in \mathcal{D}\}$$

We introduce the following useful notations: $\mathcal{H}_w(x^+)$ is the neighborhood of the current best solution x^+, that is a hyper-rectangle centered in x^+ and with sides $w \in \mathbb{R}^d$ (if the search space Ω is rescaled in $[0;\ 1]^d$, as in the experiments in this paper, then $\mathcal{H}_w(x^+)$ is a hypercube with a side equal to $w \in \mathbb{R}$), $\eta \in \mathbb{N}_+$ is the threshold on the number of solutions belonging to $\mathcal{H}_w(x^+)$ to discard the pure exploitative decision in favor of a pure explorative one. Although they could be thought of as hyperparameters of the approach, we provide a reasonable rule-of-thumb to set them, avoiding expensive fine-tuning procedures. Finally, we also introduce uncertainty quantification to minimize the overall uncertainty in case the exploitative decision is discarded. It is known as IDW, already adopted in global optimization [73] and has empirically resulted in more coherent modeling of uncertainty quantification in human searches [32,33]. It is important to clarify the conceptual role of IDW in mastering. IDW quantifies spatial sparsity: z(x) is large where past observations are scarce and small where they are dense. It is not an epistemic uncertainty in the Bayesian sense, which would require a posterior distribution over functions, nor a decision-theoretic exploration value. Rather, it serves as a surrogate-independent proxy for exploration potential, under the practical assumption that spatially under-explored regions carry residual uncertainty about the location of the optimum. Moreover, contrary to GP's predictive uncertainty, $\sigma(x)$, IDW is model-free and is able to avoid the variance starvation issue. The IDW, denoted with $z(x)$, is defined as follows [73]:

$$z(\mathrm{x}) = \begin{cases} 0 & \text{if } \mathrm{x} \in X \\ \frac{2}{\pi}\tan^{-1}\left(\frac{1}{\sum_{i=1}^{n} p_i(\mathrm{x})}\right) & \text{otherwise} \end{cases} \tag{13}$$

where $p_i(x) = \frac{e^{-\|x-x_i\|^2}}{\|x-x_i\|^2}$.

Another important feature of our algorithm is a final refining phase, meaning that the very last decisions are all purely exploitative. This is in line with more recent approaches, such as [36], and contrary to the historical ones, such as [34]. Indeed, our algorithm does not need to increase exploration with the BO iterations because it is automatically triggered in the case of a convergence to local optima. However, in the case of an objective function excessively triggering exploration, it could be helpful to spend the last BO iterations improving (aka refining) the best solution observed so far.

The rationale behind the neighborhood condition deserves further clarification. If the neighborhood $\mathcal{H}_w(x^+)$ already contains η or more observations, the GP has been queried extensively near the current best solution. In this regime, the exploitative step, minimizing the GP predictive mean, will almost certainly return a point within or very close to the same neighborhood, offering negligible new information about the objective function. The condition $|\{x_i \in X : x_i \in \mathcal{H}_w(x^+)\}| \geq \eta$ therefore operationalizes a simple principle: when the local region around the best solution is already well-sampled, a purely exploitative decision is unlikely to produce meaningful improvement and should be replaced by exploration. This is consistent with findings from human search studies [32,33], where subjects switch to exploration precisely when local improvement appears exhausted.

The refining phase serves a complementary purpose. The exploration trigger can fire repeatedly on functions with many local optima or flat regions, potentially redirecting budget away from the global optimum once it has been located. The final $N_2 = 5 \times d$ iterations of pure exploitation provide a safeguard: regardless of the exploration history, the algorithm commits to refining the best solution found before terminating. This mirrors the behavior of [36], which also ends with a more exploitative phase, and ensures that any exploratory detour in the middle of the optimization process does not prevent convergence to a high-quality final solution.

4. Experimental Setting

Experiments consider 10 well-known global optimization test functions, widely used in the literature, and with search spaces of different dimensionalities: Branin ($d = 2$), Camel3 ($d = 3$), Camel6 ($d = 6$), GoldPr ($d = 2$), Hartmann3 ($d = 3$), Hartmann4 ($d = 4$), Hartmann6 ($d = 6$), Rosenbrock ($d = 2$), Schwefel ($d = 2$), and StybTang ($d = 2$).

The proposed approach has been compared against eight state-of-the-art improvement-based acquisition functions, specifically: CB with constant β (i.e., $\beta = 1$), CB with β scheduled as in Theorem 1 and Theorem 2 of [34], randomized GP-CB [69], and εRS and εPF from [30], alternating and switching between EI and PI as proposed in [36].

As far as the hyperparameters of the BO algorithms considered are concerned, every paper shows that a unique optimal configuration does not exist. For each algorithm, the hyperparameter configuration offering the best performance on the largest number of test problems has been selected. All the hyperparameter configurations are explicitly reported in the code, freely available, as detailed in Section 6. For the proposed algorithm, the suggested hyperparameter values are $w = 0.1$ and $\eta = \lfloor 15 \times d \rfloor / 3$. Although a dedicated sensitivity analysis is beyond the scope of this paper, the consistency of results across all 10 test functions, spanning dimensionalities from d = 2 to d = 6 and substantially different geometric structures, with a single fixed configuration, provides indirect evidence of robustness. If the method were highly sensitive to w and η, uniform performance across such diverse problems would be unlikely.

For each test problem and for each BO algorithm, 100 independent runs have been performed. In each run, the algorithms share the same initialization of D. The initial size of D is $5 \times d$, while the overall number of function evaluations is $N = 20 \times d$. For the proposed algorithm, we split into $N_1 = 15 \times d$ and $N_2 = 5 \times d$ (for the refining phase).

5. Results

This section summarizes the empirical results obtained from the experiments. We started with the most important result, concisely summarized in Table 1: a cross indicates that a certain algorithm (column) was Pareto optimal on a certain test problem (row).

Surprisingly, CB with constant $\beta = 1$ resulted in Pareto optimality more times than other CB-based acquisition functions. However, it is important to remark that Pareto optimality here refers to the exploration–exploitation trade-off: an acquisition function leading to sub-optimal solutions (i.e., low area under the average GAP metric) could result in Pareto optimality if it provides really low L2-discrepancy. This happens with the problems Camel3, Camel6, and GoldPr, in which the constant $\beta = 1$ is surely higher than those provided by the other CB-based acquisition functions. As we said, high values of β lead to over-exploration. On the contrary, on the test problems Hartmann3, Hartmann4, Hartmann6, and Schwefel, constant $\beta = 1$ leads to under-exploration when compared to [34] Theorem 1 and Theorem 2.

https://doi.org/10.3390/make8070193

We can conclude that, although Pareto optimal on 7 out of 10 test problems, CB with constant $\beta = 1$ was almost always located at the extreme of the Pareto front, consequently representing an unbalanced trade-off between exploration and exploitation. This can be observed in Figures 2–11. Every figure consists of three charts as follows:

- On the left: the GAP curve averaged on 100 independent runs is depicted for each algorithm. Standard deviations are omitted for a clear visualization—in Appendix A, the box plots of the GAP value at the end of the optimization processes are reported: there are no significant differences among methods. The first point of the curves refers to the best solution over the LHS-initialized D.
- In the middle: the boxplot of the L2-discrepancies obtained over the 100 independent runs, separately for each algorithm.
- On the right: Pareto analysis among the algorithms with respect to: (a) area under the average GAP curve (converges to the optimum) and (b) L2-discrepancy (capacity to explore).

Table 1. Pareto optimal acquisition functions on 10 global optimization test problems. A cross is present if the acquisition function was Pareto optimal for the problem.

	CB Const. β	**Theorem 1 [a]**	**Theorem 2 [a]**	**Randomized CB**	**ε RS**	**ε PF**	**PI is Back! Alternating**	**PI is Back! Switching**	**Mastering (Proposed)**
Branin ($d = 2$)			x				x	x	x
Camel3 ($d = 3$)	x				x		x		x
Camel6 ($d = 6$)	x		x		x	x	x		x
GoldPr ($d = 2$)	x				x			x	x
Hartmann3 ($d = 3$)	x				x	x	x		x
Hartmann4 ($d = 4$)	x			x			x		x
Hartmann6 ($d = 6$)	x					x			x
Rosenbrock ($d = 2$)			x				x	x	
Schwefel ($d = 2$)	x		x	x	x	x	x		x
StybTang ($d = 2$)				x			x	x	x
	7/10	0/10	4/10	3/10	5/10	4/10	8/10	4/10	9/10

[a] From ref. [34].

Another acquisition function that resulted in Pareto optimality on a large number of test problems is the alternating schema between EI and PI [36]. It is interesting to highlight that this acquisition function is located on the opposite side of the Pareto front with respect to CB with constant $\beta = 1$, on all the test problems on which both algorithms resulted in Pareto optimality.

Finally, the proposed approach resulted in Pareto optimality on all the test problems but one (i.e., Rosenbrock). More importantly, it is almost always (6/9) located in the central part of the Pareto front (i.e., Camel3, camel6, GoldPr, Hartmann3, Hartmann4, Schwefel). This point is crucial because it means that the proposed approach, beyond Pareto optimality, offers the best exploration–exploitation balance.

As a final remark, only the methods proposed in [36] (alternating and switching) and CB with Theorem 2 scheduling [34] resulted in Pareto optimality on the Rosenbrock test problem, while mastering did not. The Rosenbrock function (d = 2) presents a narrow curved valley leading to the global optimum, with a basin of attraction that is geometrically thin in the orthogonal direction. In this setting, observations quickly accumulate along the valley floor, causing the neighborhood condition $|\mathcal{H}_w(x^+)| \geq \eta$ to trigger repeatedly and redirect budget toward orthogonal directions that are spatially sparse but informationally unproductive. The three methods that succeed share a common trait: they either use a strongly exploitative component (PI) or a decaying exploration bonus (Theorem 2), both of which bias the algorithm toward the valley floor rather than away from it. This identifies a general property that improvement-based AFs based on spatial sparsity as a proxy for exploration value require: the basin of attraction of the global optimum should be broad enough relative to the neighborhood size that spatial density and functional informativeness remain correlated throughout the run. Functions with narrow, geometrically isolated basins represent the natural boundary of mastering's applicability.

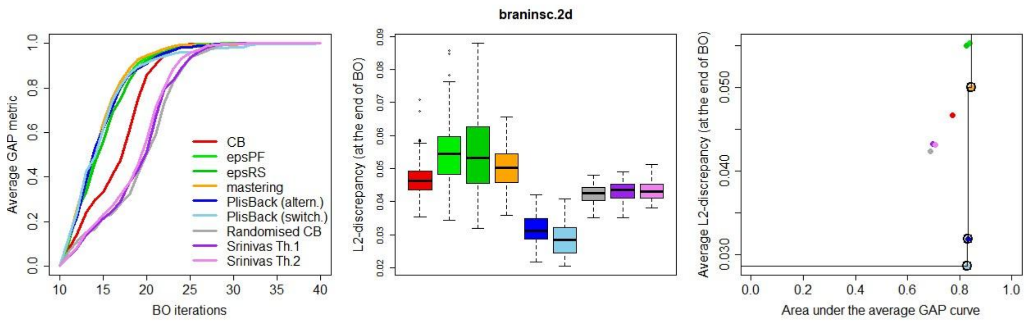

Figure 2. Branin test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

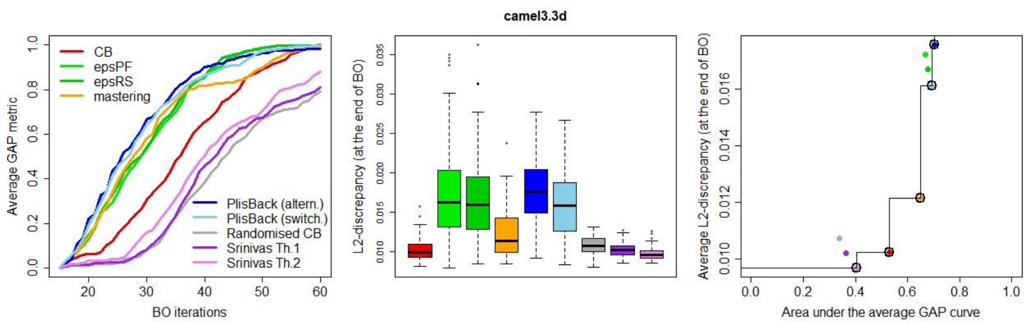

Figure 3. Camel3 test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

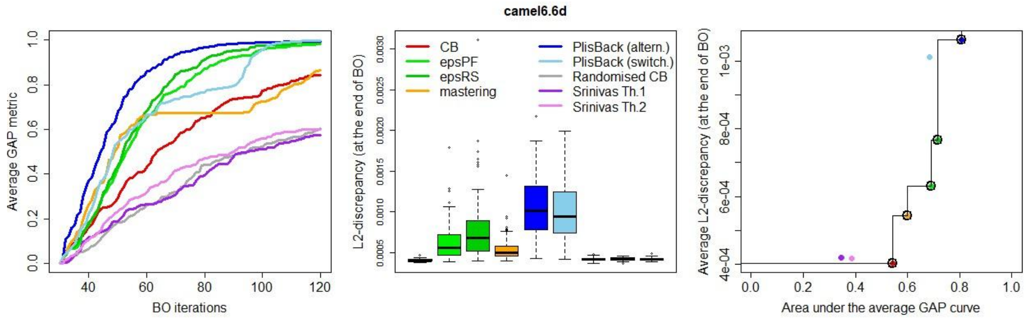

Figure 4. Camel6 test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

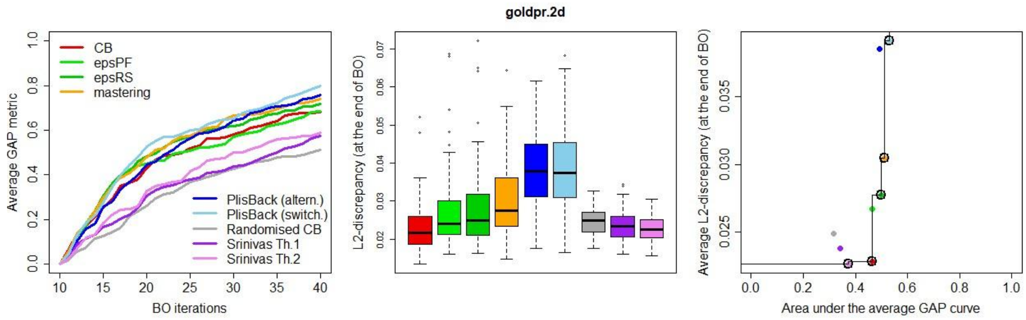

Figure 5. GoldPr test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

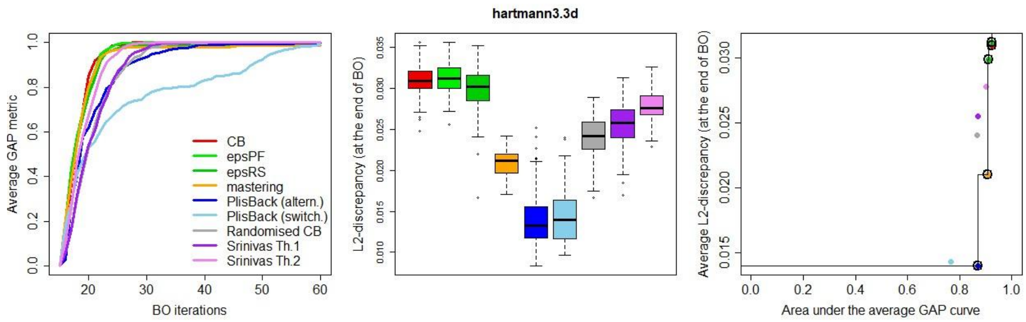

Figure 6. Hartmann3 test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

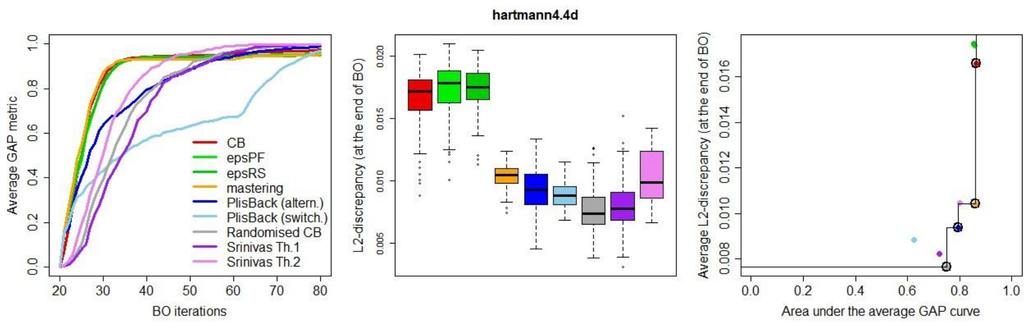

Figure 7. Hartmann4 test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

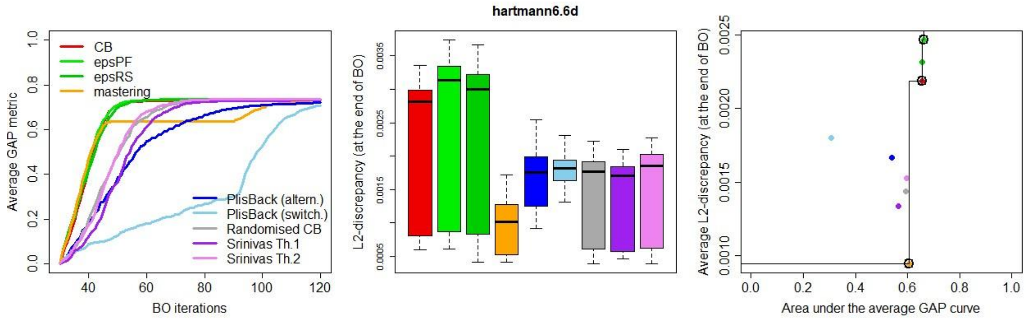

Figure 8. Hartmann6 test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

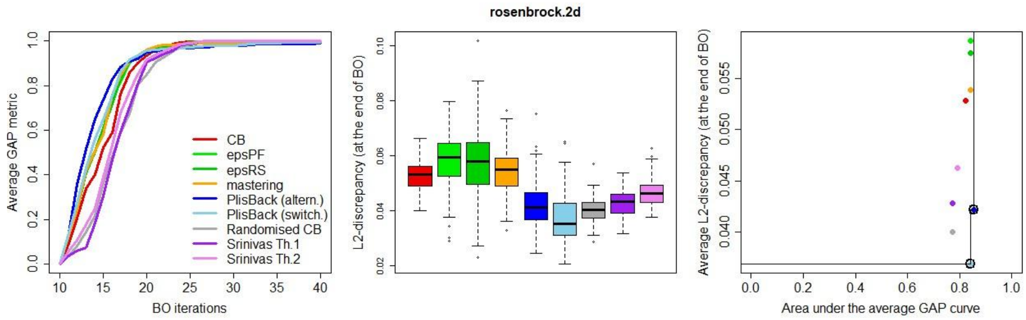

Figure 9. Rosenbrock test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

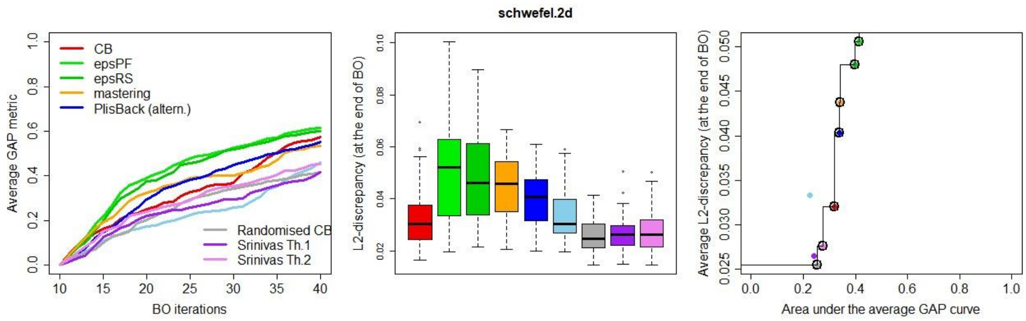

Figure 10. Schwefel test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

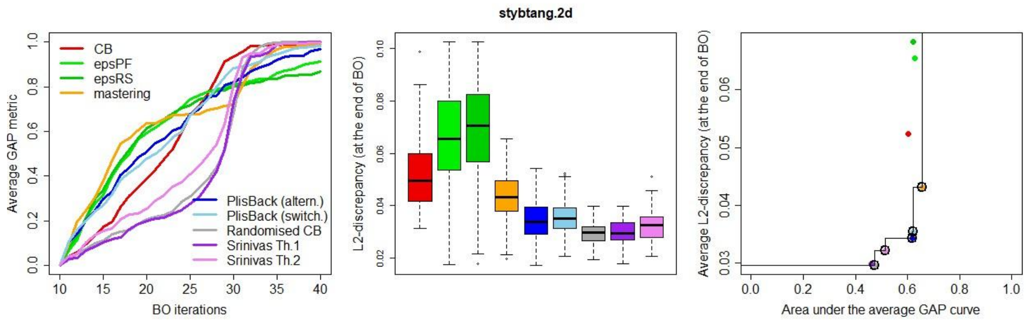

Figure 11. StybTang test function: Average GAP metric curves (**left**), L2-discrepancy at the end of the optimization processes (**middle**), and Pareto analysis between area under the GAP curve and average L2-discrepancy (**right**).

6. Replicability of Experiments: Open-Source Code

To guarantee replicability of the experiments, the entire code is available for free at the following link, along with results and figures reported in this paper: https://github.com/acandelieri/MasteringExplorationExploitationInBO (accessed on 1 June 2023). Within the code, all the values of the hyperparameters for all the methods are also specified.

7. Discussion and Conclusions

The exploration–exploitation balance is still an open challenge in all the learning-and-optimization frameworks, from evolutionary algorithms to reinforcement learning and global optimization of black-box expensive functions. In this paper, the improvement-based acquisition functions adopted in Bayesian optimization have been investigated, considering both foundational and more recent methods. Limitations have been identified, specifically the fact that all the methods are based on fixed or random scheduling of the uncertainty bonus to guarantee convergence to the global optimum.

Results and insights from studies on human search suggested how to implement a novel mechanism to dynamically and adaptively switch between exploration and exploitation over the entire optimization process, depending on the feedback collected. The comparison among the acquisition functions considered has been performed by proposing a Pareto analysis between the convergence to the optimum (exploitation), measured as the area under the GAP metric curve, and the exploration capabilities, over the entire optimization process.

As a result, the proposed approach resulted in Pareto optimality in 9 out of 10 of the test problems and, even more importantly, it is usually located in the central part of the Pareto front, meaning that it is the one providing the most balanced trade-off between exploration and exploitation.

The main limitation of the proposed approach is that, at each iteration, first a pure exploitative decision is suggested (i.e., by optimizing the GP's predictive mean), if discarded because resulting useless, another pure explorative decision must be computed (i.e., by minimizing the uncertainty quantified via IDW). In the worst case, two internal optimization problems are solved in each BO iteration. While this is the worst case, overhead doubles the number of internal optimizer calls per iteration, and both subproblems are solved on cheap surrogates (the GP mean and the model-free IDW measure). In all target applications of BO, the cost of these internal optimizations is negligible compared to the cost of evaluating the black-box objective function, so the practical computational burden of mastering is equivalent to any single-acquisition-function BO method.

A further limitation, identified by the Rosenbrock result, is that the IDW-based exploration trigger implicitly assumes that spatial sparsity correlates with residual uncertainty about the optimum. This assumption fails on functions with narrow basins of attraction, where unvisited regions are spatially sparse but functionally irrelevant. Designing an exploration trigger that accounts for functional geometry, rather than purely spatial density, is an open direction for future work.

Perspectives: This work opens up several promising avenues for future research. The focus here was on the simplest single-output BO algorithms with a GP surrogate model. Still, it can be extended to various adaptations of related problems, such as constrained optimization, noisy evaluations, or alternative surrogate models. Regarding the latter, it is important to note that mastering's two components have different surrogate dependencies. The explorative component, based on IDW, depends only on the locations of past observations and is entirely surrogate-agnostic; replacing the GP with any alternative model leaves the exploration mechanism unchanged. The exploitative component relies on the surrogate's predictive mean $\mu(x)$; replacing the GP with an alternative surrogate (e.g., a random forest, a deep GP, or a Bayesian neural network) requires only substituting $\mu(x)$ with that model's point prediction, leaving the algorithm's structure intact. The quality of the exploitation signal will depend on the calibration of the chosen surrogate, as with any surrogate-based BO method. This partial independence from the GP is a direct consequence of the design choice to use IDW rather than GP predictive variance for the explorative step, a choice motivated precisely by avoiding the bias and variance starvation issues that arise when the explorative component is tied to the GP model. Although it cannot be considered the ultimate acquisition function, the design of the adaptive scheduling of MASTER could be generalized to other BO architectures and application contexts.

Challenging perspectives for BO and hot research targets are high-dimensional BO and BO over structured data. High-dimensional spaces have been traditionally difficult for BO, and exploiting structural assumptions such as additivity, locality, or sparsity, which can be naturally encoded in the MASTER mechanism to achieve better performance than "vanilla" BO without requiring many hyperparameters and still relying on basic GP priors, just scaling the length-scale hyperparameter. Beyond Bayesian optimization, the broader question of adaptive decision-making under uncertainty also arises in other combinatorial and network optimization settings. For instance, similar trade-offs between exploiting structural information and exploring alternative configurations appear in Gromov–Wasserstein-based approaches to combinatorial assignment problems [74] and in distributional resilience analysis for network structures [75]. Investigating whether the adaptive switching mechanism proposed in this paper could inform decision rules in these related optimization contexts is a direction for future work.

The issue of high-dimensional BO is also for structured spaces, such as graphs and permutation spaces. A first challenge for graph-based optimization problems is that evaluating the unknown objective at each node—via sampling, simulation, or physical experimentation—can be very expensive, and second, to leverage structural correlations and latent biases enclosed in the underlying graph topology into a kernel over graphs is needed to facilitate efficient search and optimization using limited observations and is more difficult than in continuous spaces. An important point is that the new acquisition is conceptually different from the assumptions regulating convergence of LCB/UCB and does not require checking whether the path of the generative GP depends on our assumption that f is a sample path of a GP instead of an element of the RKHS (reproducing kernel Hilbert space).

The MASTER acquisition function can be adapted to a class of shortest-path kernels so that it can handle optimization over attributed graphs, formulating the problem as mixed-integer optimization problems, enabling global exploration of the graph domain while maintaining solution feasibility even in the presence of problem-specific constraints and inexact acquisition. FunBO: discovering acquisition functions for Bayesian optimization with FunSearch. This work takes on the challenge of designing novel AFs that perform well across a variety of experimental settings.

It uses LLMs that can learn new AFs written in computer code using a limited number of observations for a set of objective functions. For all discovered AFs, an analytical expression can be tested. Funsearch is a recently proposed evolutionary algorithm for searching in the functional space by combining a pre-trained LLM used for generating new computer programs with an efficient evaluator that guards against hallucinations and scores fitness. FunBO exploits a set of auxiliary functions. FunBO sequentially prompts an LLM to improve an initial AF expressed in code. The evaluation score for AF gives programs that are stored and act as a training set for fine-tuning new AFs.

Author Contributions: Conceptualization, F.A. and A.C.; methodology, F.A. and A.C.; software, A.C.; validation, A.C. and I.S.; writing—original draft preparation, A.C. and F.A.; writing—review and editing, all; visualization, A.C. and I.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original data presented in the study are openly available on GitHub at https://github.com/acandelieri/MasteringExplorationExploitationInBO.git (accessed on 21 April 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

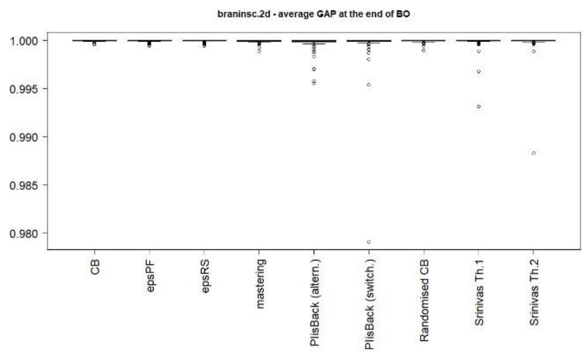

Figure A1. Branin test function: GAP values at the end of optimization processes (on 100 independent runs).

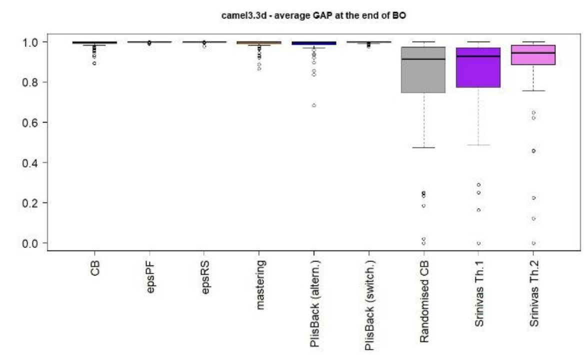

Figure A2. Camel3 test function: GAP values at the end of optimization processes (on 100 independent runs).

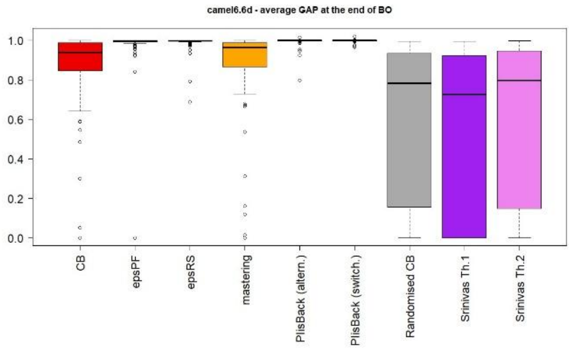

Figure A3. Camel6 test function: GAP values at the end of optimization processes (on 100 independent runs).

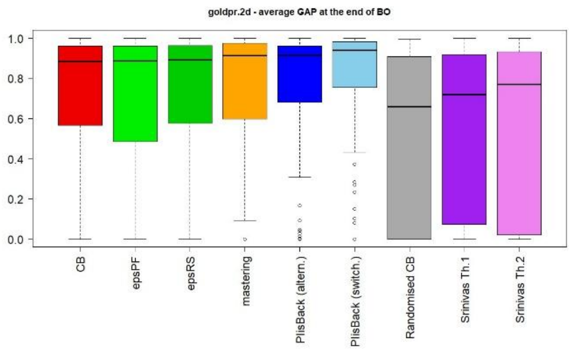

Figure A4. GoldPr test function: GAP values at the end of optimization processes (on 100 independent runs).

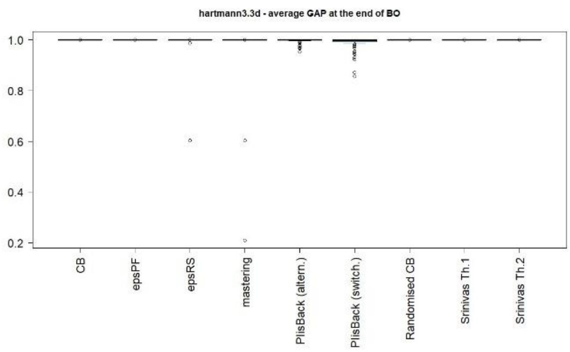

Figure A5. Hartmann3 test function: GAP values at the end of optimization processes (on 100 independent runs).

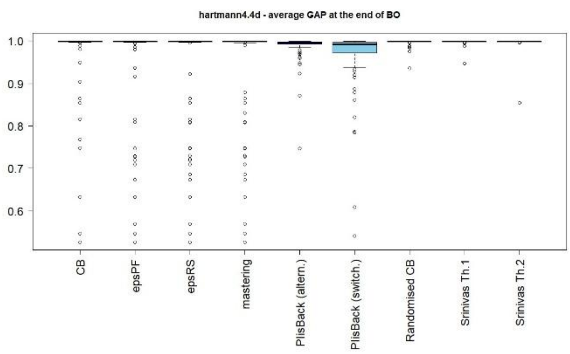

Figure A6. Hartmann4 test function: GAP values at the end of optimization processes (on 100 independent runs).

https://doi.org/10.3390/make8070193

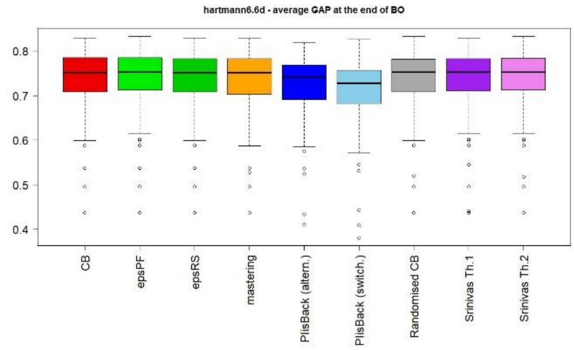

Figure A7. Hartmann6 test function: GAP values at the end of optimization processes (on 100 independent runs).

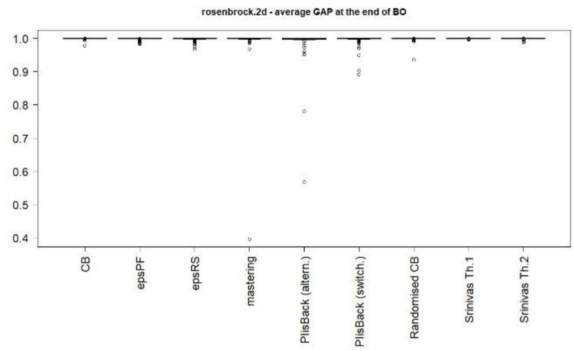

Figure A8. Rosenbrock test function: GAP values at the end of optimization processes (on 100 independent runs).

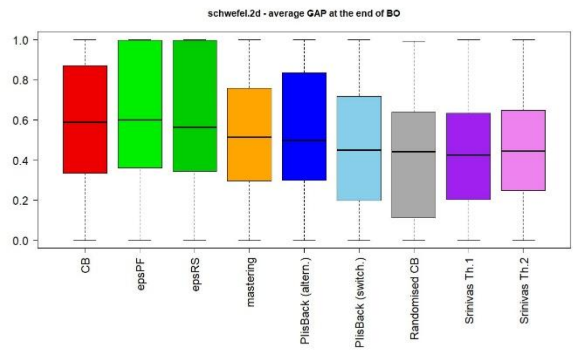

Figure A9. Schwefel test function: GAP values at the end of optimization processes (on 100 independent runs).

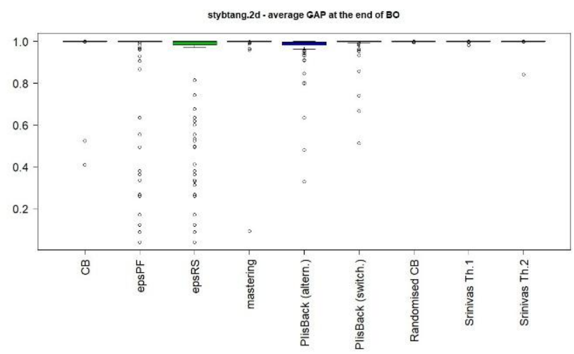

Figure A10. StybTang test function: GAP values at the end of optimization processes (on 100 independent runs).

References

1. Archetti, F.; Candelieri, A. *Bayesian Optimization and Data Science*; SpringerBriefs in Optimization; Springer International Publishing: Cham, Switzerland, 2019. https://doi.org/10.1007/978-3-030-24494-1.
2. Frazier, P.I. Bayesian Optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*; Gel, E., Ntaimo, L., Shier, D., Greenberg, H.J., Eds.; INFORMS: Catonsville, MD, USA, 2018, pp. 255–278. https://doi.org/10.1287/educ.2018.0188.
3. Garnett, R. *Bayesian Optimization*; Cambridge University Press: Cambridge, UK, 2023.
4. Lam, R.; Poloczek, M.; Frazier, P.; Willcox, K.E. Advances in Bayesian Optimization with Applications in Aerospace Engineering. In *2018 AIAA Non-Deterministic Approaches Conference, Kissimmee, FL, USA, 8–12 January 2018*; American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2018. https://doi.org/10.2514/6.2018-1656.
5. Wang, K.; Dowling, A.W. Bayesian optimization for chemical products and functional materials. *Curr. Opin. Chem. Eng.* **2022**, *36*, 100728.
6. Wang, Y.; Chen, T.-Y.; Vlachos, D.G. NEXTorch: A Design and Bayesian Optimization Toolkit for Chemical Sciences and Engineering. *J. Chem. Inf. Model.* **2021**, *61*, 5312–5319. https://doi.org/10.1021/acs.jcim.1c00637.
7. Baheri, A.; Bin-Karim, S.; Bafandeh, A.; Vermillion, C. Real-time control using Bayesian optimization: A case study in airborne wind energy systems. *Control Eng. Pract.* **2017**, *69*, 131–140.
8. Candelieri, A.; Galuzzi, B.; Giordani, I.; Archetti, F. Learning Optimal Control of Water Distribution Networks Through Sequential Model-Based Optimization. In *Learning and Intelligent Optimization*; Kotsireas, I.S., Pardalos, P.M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2020; Volume 12096, pp. 303–315. https://doi.org/10.1007/978-3-030-53552-0_28.
9. Candelieri, A.; Ponti, A.; Archetti, F. Safe-Exploration of Control Policies from Safe-Experience via Gaussian Processes. In *Learning and Intelligent Optimization*; Simos, D.E., Rasskazova, V.A., Archetti, F., Kotsireas, I.S., Pardalos, P.M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2022; Volume 13621, pp. 232–247. https://doi.org/10.1007/978-3-031-24866-5_18.
10. Candelieri, A.; Archetti, F.; Seyedi, I.; Ponti, A. Resource Allocation via Bayesian Optimization in Wasserstein Spaces vs. Semi-Bandit Feedback. *Big Data Cogn. Comput.* **2026**, *10*, 206. https://doi.org/10.3390/bdcc10070206.
11. Candelieri, A.; Perego, R.; Archetti, F. Green machine learning via augmented Gaussian processes and multi-information source optimization. *Soft Comput.* **2021**, *25*, 12591–12603. https://doi.org/10.1007/s00500-021-05684-7.
12. Candelieri, A.; Ponti, A.; Archetti, F. Fair and green hyperparameter optimization via multi-objective and multiple information source Bayesian optimization. *Mach. Learn.* **2024**, *113*, 2701–2731. https://doi.org/10.1007/s10994-024-06515-0.
13. Hutter, F.; Kotthoff, L.; Vanschoren, J. *Automated Machine Learning: Methods, Systems, Challenges*; Springer Nature: Cham, Switzerland, 2019.
14. Perego, R.; Candelieri, A.; Archetti, F.; Pau, D. AutoTinyML for microcontrollers: Dealing with black-box deployability. *Expert Syst. Appl.* **2022**, *207*, 117876.
15. Waring, J.; Lindvall, C.; Umeton, R. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artif. Intell. Med.* **2020**, *104*, 101822.
16. Zöller, M.-A.; Huber, M.F. Benchmark and survey of automated machine learning frameworks. *J. Artif. Intell. Res.* **2021**, *70*, 409–472.
17. Gramacy, R.B. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2020.
18. Williams, C.K.; Rasmussen, C.E. *Gaussian Processes for Machine Learning*, MIT Press: Cambridge, MA, USA, 2006; Volume 2.
19. Balandat, M.; Karrer, B.; Jiang, D.; Daulton, S.; Letham, B.; Wilson, A.G.; Bakshy, E. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21524–21538.
20. Bischl, B.; Richter, J.; Bossek, J.; Horn, D.; Thomas, J.; Lang, M. mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions. *arXiv* **2018**, arXiv:1703.03373. https://doi.org/10.48550/arXiv.1703.03373.
21. Dewancker, I.; McCourt, M.; Clark, S. Bayesian Optimization for Machine Learning: A Practical Guidebook. *arXiv* **2016**, arXiv:1612.04858. https://doi.org/10.48550/arXiv.1612.04858.
22. Golovin, D.; Solnik, B.; Moitra, S.; Kochanski, G.; Karro, J.; Sculley, D. Google Vizier: A Service for Black-Box Optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017*; ACM: New York, NY, USA, 2017; pp. 1487–1495. https://doi.org/10.1145/3097983.3098043.
23. Lindauer, M.; Eggensperger, K.; Feurer, M.; Biedenkapp, A.; Deng, D.; Benjamins, C.; Ruhkopf, T.; Sass, R.; Hutter, F. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. *J. Mach. Learn. Res.* **2022**, *23*, 1–9.

24. Ramirez, J.C.; Le Riche, R.; Roustant, O.; Perrin, G.; Durantin, C.; Glière, A. A comparison of mixed-variables Bayesian optimization approaches. *Adv. Model. Simul. Eng. Sci.* **2022**, *9*, 6. https://doi.org/10.1186/s40323-022-00218-8.
25. Nguyen, D.; Gupta, S.; Rana, S.; Shilton, A.; Venkatesh, S. Bayesian optimization for categorical and category-specific continuous inputs. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 5256–5263. https://doi.org/10.1609/aaai.v34i04.5971.
26. Ru, B.; Alvi, A.; Nguyen, V.; Osborne, M.A.; Roberts, S. Bayesian optimisation over multiple continuous and categorical inputs. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2020; pp. 8276–8285. Available online: http://proceedings.mlr.press/v119/ru20a (accessed on 26 January 2026).
27. Seyedi, I.; Candelieri, A.; Archetti, F. Distributionally Robust Bayesian Optimization via Sinkhorn-based Wasserstein Barycenter. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 3. https://doi.org/10.3390/make7030090.
28. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; de Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **2015**, *104*, 148–175.
29. Neiswanger, W.; Yu, L.; Zhao, S.; Meng, C.; Ermon, S. Generalizing bayesian optimization with decision-theoretic entropies. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 21016–21029.
30. De Ath, G.; Everson, R.M.; Rahat, A.A.M.; Fieldsend, J.E. Greed Is Good: Exploration and Exploitation Trade-offs in Bayesian Optimisation. *ACM Trans. Evol. Learn. Optim.* **2021**, *1*, 1–22. https://doi.org/10.1145/3425501.
31. Žilinskas, A.; Calvin, J. Bi-objective decision making in global optimization based on statistical models. *J. Glob. Optim.* **2019**, *74*, 599–609. https://doi.org/10.1007/s10898-018-0622-5.
32. Candelieri, A.; Ponti, A.; Archetti, F. Explaining Exploration–Exploitation in Humans. *Big Data Cogn. Comput.* **2022**, *6*, 155.
33. Candelieri, A.; Ponti, A.; Archetti, F. Uncertainty quantification and exploration–exploitation trade-off in humans. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 6843–6876. https://doi.org/10.1007/s12652-021-03547-5.
34. Srinivas, N.; Krause, A.; Kakade, S.M.; Seeger, M.W. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Trans. Inf. Theory* **2012**, *58*, 3250–3265.
35. Berk, J.; Nguyen, V.; Gupta, S.; Rana, S.; Venkatesh, S. Exploration Enhanced Expected Improvement for Bayesian Optimization. In *Machine Learning and Knowledge Discovery in Databases*; Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2019; Volume 11052, pp. 621–637. https://doi.org/10.1007/978-3-030-10928-8_37.
36. Benjamins, C.; Raponi, E.; Jankovic, A.; van der Blom, K.; Santoni, M.L.; Lindauer, M.; Doerr, C. PI is back! Switching Acquisition Functions in Bayesian Optimization. *arXiv* **2022**, arXiv:2211.01455. https://doi.org/10.48550/arXiv.2211.01455.
37. Siivola, E.; Paleyes, A.; González, J.; Vehtari, A. Good practices for Bayesian optimization of high dimensional structured spaces. *Appl. AI Lett.* **2021**, *2*, e24. https://doi.org/10.1002/ail2.24.
38. Malu, M.; Dasarathy, G.; Spanias, A. Bayesian optimization in high-dimensional spaces: A brief survey. In *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*; IEEE: New York, NY, USA, 2021; pp. 1–8. https://doi.org/10.1109/IISA52424.2021.9555522.
39. Yang, G.; Qin, H.; Fu, Y. Trust-Region Bayesian Optimization for High-Dimensional Black-Box Problems: Integrating Deep Kernel Learning with Adaptive Gradient Mechanisms. In *International Conference on Machine Learning and Intelligent Computing*; PMLR: Cambridge, MA, USA, 2025; pp. 644–659. Available online: https://raw.githubusercontent.com/mlresearch/v278/main/assets/yang25f/yang25f.pdf (accessed on 27 January 2026).
40. Doumont, C.; Fan, D.; Maus, N.; Gardner, J.R.; Moss, H.; Pleiss, G. We Still Don't Understand High-Dimensional Bayesian Optimization. *arXiv* **2025**, arXiv:2512.00170. https://doi.org/10.48550/arXiv.2512.00170.
41. Hvarfner, C.; Hellsten, E.O.; Nardi, L. Vanilla Bayesian Optimization Performs Great in High Dimensions. *arXiv* **2024**, arXiv:2402.02229. https://doi.org/10.48550/arXiv.2402.02229.
42. Papenmeier, L.; Nardi, L.; Poloczek, M. Bounce: Reliable high-dimensional Bayesian optimization for combinatorial and mixed spaces. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 1764–1793.
43. Papenmeier, L.; Poloczek, M.; Nardi, L. Understanding High-Dimensional Bayesian Optimization. *arXiv* **2025**, arXiv:2502.09198. https://doi.org/10.48550/arXiv.2502.09198.
44. Xu, Z.; Wang, H.; Phillips, J.M.; Zhe, S. Standard Gaussian Process is All You Need for High-Dimensional Bayesian Optimization. *arXiv* **2025**, arXiv:2402.02746. https://doi.org/10.48550/arXiv.2402.02746.
45. Baptista, R.; Poloczek, M. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2018; pp. 462–471. Available online: https://proceedings.mlr.press/v80/baptista18a (accessed on 14 December 2025).

46. Li, W.; Sanz-Alonso, D.; Yang, R. Bayesian Optimization on Networks. *arXiv* **2025**, arXiv:2510.27643. https://doi.org/10.48550/arXiv.2510.27643.
47. Kim, H.; Sanz-Alonso, D.; Yang, R. Optimization on Manifolds via Graph Gaussian Processes. *SIAM J. Math. Data Sci.* **2024**, *6*, 1–25. https://doi.org/10.1137/22M1529907.
48. Kim, H.; Liu, C.; Chen, Y. Bayesian Optimization with Inexact Acquisition: Is Random Grid Search Sufficient? *arXiv* **2025**, arXiv:2506.11831. https://doi.org/10.48550/arXiv.2506.11831.
49. Kim, H.; Sanz-Alonso, D. Enhancing Gaussian Process Surrogates for Optimization and Posterior Approximation via Random Exploration. *SIAM/ASA J. Uncertain. Quantif.* **2025**, *13*, 1054–1084. https://doi.org/10.1137/24M1677009.
50. Hong, S.; Mei, Y.; Imani, M.; Lan, T. Global Optimization on Graph-Structured Data via Gaussian Processes with Spectral Representations. *arXiv* **2025**, arXiv:2511.07734. https://doi.org/10.48550/arXiv.2511.07734.
51. Oh, C.; Bondesan, R.; Gavves, E.; Welling, M. Batch Bayesian optimization on permutations using the acquisition weighted kernel. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 6843–6858.
52. Deshwal, A.; Belakaria, S.; Doppa, J.R.; Kim, D.H. Bayesian optimization over permutation spaces. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 6515–6523. https://doi.org/10.1609/aaai.v36i6.20604.
53. Xie, Z.; Chen, L. From Sorting Algorithms to Scalable Kernels: Bayesian Optimization in High-Dimensional Permutation Spaces. *arXiv* **2025**, arXiv:2507.13263. https://doi.org/10.48550/arXiv.2507.13263.
54. Huang, B.-W.; Fang, W.-Z.; Liao, H.-C.; Yu, T.-L. Relational Bayesian Optimization for Permutation. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation, Lisbon, Portugal, 15–19 July 2023*; ACM: New York, NY, USA, 2023; pp. 471–474. https://doi.org/10.1145/3583133.3590606.
55. Xie, Z.; Chen, L. Merge Kernel for Bayesian Optimization on Permutation Space. *arXiv* **2025**, arXiv:2507.13263v1.
56. Liu, D.; Zhang, H. Residuals and Diagnostics for Ordinal Regression Models: A Surrogate Approach. *J. Am. Stat. Assoc.* **2018**, *113*, 845–854. https://doi.org/10.1080/01621459.2017.1292915.
57. Li, S.; Fan, Z.; Liu, I.; Morrison, P.S.; Liu, D. Surrogate method for partial association between mixed data with application to well-being survey analysis. *Ann. Appl. Stat.* **2024**, *18*, 2254–2276.
58. Hennig, P.; Schuler, C.J. Entropy search for information-efficient global optimization. *J. Mach. Learn. Res.* **2012**, *13*, 1809–1837.
59. Hernández-Lobato, J.M.; Hoffman, M.W.; Ghahramani, Z. Predictive entropy search for efficient global optimization of black-box functions. *Adv. Neural Inf. Process. Syst.* **2014**, *27*. Available online: https://proceedings.neurips.cc/paper_files/paper/2014/hash/6488484c982e9af5c35689523ba1abfe-Abstract.html (accessed on 12 December 2025).
60. Wang, Z.; Jegelka, S. Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2017; pp. 3627–3635.
61. Hvarfner, C.; Hutter, F.; Nardi, L. Joint entropy search for maximally-informed Bayesian optimization. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 11494–11506.
62. Wilson, J.; Borovitskiy, V.; Terenin, A.; Mostowsky, P.; Deisenroth, M. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2020; pp. 10292–10302. Available online: http://proceedings.mlr.press/v119/wilson20a.html (accessed on 27 January 2026).
63. Frazier, P.; Powell, W.; Dayanik, S. The Knowledge-Gradient Policy for Correlated Normal Beliefs. *Inf. J. Comput.* **2009**, *21*, 599–613. https://doi.org/10.1287/ijoc.1080.0314.
64. Regis, R.G. Trust regions in Kriging-based optimization with expected improvement. *Eng. Optim.* **2016**, *48*, 1037–1059. https://doi.org/10.1080/0305215X.2015.1082350.
65. Diouane, Y.; Picheny, V.; Riche, R.L.; Perrotolo, A.S.D. TREGO: A trust-region framework for efficient global optimization. *J. Glob. Optim.* **2023**, *86*, 1–23.
66. Eriksson, D.; Pearce, M.; Gardner, J.; Turner, R.D.; Poloczek, M. Scalable global optimization via local Bayesian optimization. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5496 – 5507.
67. Hofmann, T.; Schölkopf, B.; Smola, A.J. Kernel methods in machine learning. *Ann. Stat.* **2008**, *36*, 1171–1220. https://doi.org/10.1214/009053607000000677.
68. Scholkopf, B.; Smola, A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2018.
69. Berk, J.; Gupta, S.; Rana, S.; Venkatesh, S. Randomised Gaussian Process Upper Confidence Bound for Bayesian Optimisation. *arXiv* **2020**, arXiv:2006.04296. https://doi.org/10.48550/arXiv.2006.04296.
70. Kushner, H.J. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *J. Basic Eng.* **1964**, *86*, 97–106. https://doi.org/10.1115/1.3653121.

71. Jones, D.R.; Schonlau, M.; Welch, W.J. Efficient Global Optimization of Expensive Black-Box Functions. *J. Glob. Optim.* **1998**, *13*, 455–492. https://doi.org/10.1023/A:1008306431147.
72. Fang, K.-T.; Liu, M.-Q.; Qin, H.; Zhou, Y.-D. *Theory and Application of Uniform Experimental Designs*; Lecture Notes in Statistics; Springer: Singapore, 2018; Volume 221. https://doi.org/10.1007/978-981-13-2041-5.
73. Bemporad, A. Global optimization via inverse distance weighting and radial basis functions. *Comput. Optim. Appl.* **2020**, *77*, 571–595. https://doi.org/10.1007/s10589-020-00215-w.
74. Seyedi, I.; Candelieri, A.; Messina, E.; Archetti, F. Gromov–Wasserstein Meets Combinatorial Optimization: A Scalable Solver for the Capacitated Quadratic Assignment Problem. *Mathematics* **2026**, *14*, 1972.
75. Seyedi, I.; Candelieri, A.; Archetti, F. Fused Unbalanced Gromov–Wasserstein-Based Network Distributional Resilience Analysis for Critical Infrastructure Assessment. *Mathematics* **2026**, *14*, 417. https://doi.org/10.3390/math14030417.