Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

A Gentle Introduction To Bayesian Optimization

Antonio Candelieri

Department of Economics, Management and Statistics
University of Milano-Bicocca
Piazza Ateneo Nuovo, 1
Milan, 20126, ITALY

ABSTRACT

Bayesian optimization is a sample efficient sequential global optimization method for black-box, expensive and multi-extremal functions. It generates, and keeps updated, a probabilistic surrogate model of the objective function, depending on the performed evaluations, and optimizes an acquisition function to choose a new point to evaluate. The acquisition function deals with the exploration-exploitation dilemma depending on surrogate's predictive mean and uncertainty. Many alternatives are available offering different trade-off mechanisms; different options are also possible for the probabilistic surrogate model: Gaussian Process regression is best suited for optimization over continuous search spaces while other approaches, such as Random Forests or Gaussian Prcesses with ah-hoc kernels, deal with complex search spaces spanned by nominal, numeric and conditional variables. This tutorial offers an introduction to these topics and a discussion on available tools, real-life applications, and recent advances, such as unknown constraints, multi-information sources and cost-awareness, and multi-objective optimization.

1 INTRODUCTION

Bayesian optimization (BO) is a *sample-efficient* sequential method well suited to optimize black-box, expensive and multi-extremal objective functions, under a limited *budget*, typically a maximum number of function evaluations (Archetti and Candelieri 2019; Frazier 2018; Shahriari et al. 2015). The basic idea is that every observation collected by *querying* the objective function can improve the *knowledge* about it, driving the choice of the next location to query while dealing with the well known *exploration-exploitation dilemma*. More precisely, *exploration* refers to choosing a location where the *uncertainty* about the objective function is large, while *exploitation* refers to choosing a location close to the current optimal solution. Thus, exploration and exploitation are associated to two different types of search, that is global and local, respectively. In other settings, such as evolutionary and metaheuristic approaches, exploration and exploitation are also known as *diversification* and *intensification* (Glover and Samorani 2019).

The reference problem considered in this tutorial is:

$$\max_{\mathbf{x}\in\Omega} f(\mathbf{x}) \tag{1}$$

where $\Omega \subset \Re^d$ is the so-called *search space*, $f(\mathbf{x}) : \Omega \to \Re$ is black-box, expensive and multi-extremal, and maximization is considered without loss of generality (i.e, $\max f(\mathbf{x}) = \min\{-f(\mathbf{x})\}$).

The rest of the tutorial is organized as follows: Section 2 introduces the basic components of the BO framework, Section 3 summarizes the most widely adopted methods for modelling $f(\mathbf{x})$ and Section 4 analyses the most relevant exploration-exploitation trade-off mechanisms in literature. Section 5 provides an overview of the application domains where BO is successfully applied, while Section 6 summarizes recent advances – most of them derived from challenging problems arising from the considered application

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

domains. Finally, relevant conclusions are provided, especially with respect to the most interesting and challenging research directions for BO.

2 THE BAYESIAN OPTIMIZATION FRAMEWORK

Bayesian optimization is also known as Sequential Model-based Optimization (SMBO) (Candelieri 2019; Hutter et al. 2013; Hutter et al. 2011), according to its sequential nature. Here, the basic framework is explained: assume BO is at a generic iteration and that we have already sequentially chosen n locations $\mathbf{X}_{1:n} = \{\mathbf{x}_i\}_{i=1,\dots,n}$, with $\mathbf{x}_i \in \Omega$, and observed the associated values $\mathbf{y} = \{y_i\}_{i=1,\dots,n}$, possibly noisy, that is $y_i = f(\mathbf{x}_i) + \varepsilon$, with ε a zero-mean Gaussian noise, $\varepsilon \sim \mathcal{N}(\mathbf{0}, \lambda^2)$. Since querying $f(\mathbf{x})$ is expensive and we have a limited budget, we need a "clever" strategy to choose the next location, $\mathbf{x}_{n+1}$ to query. A simple solution is to fit a regression – or interpolation – model $\widehat{f}(\mathbf{x}) \approx f(\mathbf{x})$ on the dataset of collected observations, namely $\mathbf{D}_{1:n} = (\mathbf{X}_{1:n}, \mathbf{y}_{1:n})$, and then solve $\max_{\mathbf{x}\in\Omega} \widehat{f}(\mathbf{x})$, instead of (1). When a deterministic regression method is used (e.g., an Artificial Neural Network or a Support Vector Machine), $\widehat{f}(\mathbf{x})$ is called *deterministic surrogate model* and the described strategy collapses on a simple *local search* (i.e., only exploitation), with the risk to converge and get stuck into a local maximum. To overcome this limitation, an additional model, $u(\mathbf{x})$, can be coupled to $\widehat{f}(\mathbf{x})$, specifically estimating the uncertainty throughout the search space, as proposed, for instance, in Bemporad (2020): basically, uncertainty increases with the distance from previously queried locations.

Using two decoupled models makes the role of the two antagonist search components – exploitation and exploration – evident. On the other hand, BO offers a more elegant mechanism, consisting in training a single *probabilistic surrogate model* which accounts for both exploitation – through the model's predictive mean, denoted with $\mu(\mathbf{x})$ – and exploration – through the model's predictive uncertainty, denoted with $\sigma(\mathbf{x})$. Thus, one can easily notice that, moving from deterministic to probabilistic modelling, $\widehat{f}(\mathbf{x})$ is replaced with $\mu(\mathbf{x})$, and $u(\mathbf{x})$ is replaced with $\sigma(\mathbf{x})$, but the main advantage is that only one model must be updated at each BO iteration.

While the probabilistic surrogate model offers an approximation of the objective function $f(\mathbf{x})$, along with an estimation about the predictive uncertainty, a specific mechanism is required to balance between "trusting in prediction" (exploitation) and "giving a chance to uncertainty" (exploration). This mechanism is called acquisition function (aka *utility function* or *infill criterion*), which can be denoted by $\alpha(x; (\mu(x), \sigma(x)))$ or simply $\alpha(x)$ by assuming the dependence on the updated probabilistic surrogate model. More precisely, selecting the next location to query requires to solve an internal – aka *auxiliary* or *ancillary* – optimization problem, whose computational cost is negligible with respect to querying $f(\mathbf{x})$:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\Omega} \alpha(\mathbf{x}) \tag{2}$$

The basic BO algorithm is summarized in the following Algorithm 1, while the details are discussed in the following Sections 3 and 4.

3 PROBABILISTIC SURROGATE MODEL

This section describes the probabilistic surrogate models most widely adopted in BO approaches and tools, starting from Gaussian Process (GP) regression (Gramacy 2020; Williams 2006), which is best suited for optimization over search spaces spanned by continuous dimensions, and moving to Random Forest (Ho 1995) and other Machine Learning regression methods able to deal with more complex search spaces spanned by discrete, mixed and *conditional* variables.

Algorithm 1: Basic Bayesian Optimization algorithm

Sample n_0 random locations in Ω, that is $\mathbf{X}_{1:n_0}$ (e.g., through Latin Hypercube Sampling);
Observe $\mathbf{y}_{1:n_0}$;
Organize observations into $\mathbf{D}_{1:n_0} = \{(\mathbf{x}_i, y_i)\}_{i=\ldots,n_0}$;
$n \leftarrow n_0$;
while *termination criterion is not met* **do**
  train a probabilistic surrogate model on $\mathbf{D}_{1:n}$ to obtain $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$;
  compute the acquisition function $\alpha(x, (\mu(\mathbf{x})), \sigma(\mathbf{x}))$;
  find $\mathbf{x}_{n+1} = \max_{\mathbf{x}\in\Omega} \alpha(\mathbf{x})$, with $\mathbf{x}_{n+1} \notin \mathbf{D}_{1:n}$ (especially in the noise-free setting);
  observe y_{n+1};
  $\mathbf{D}_{1:n+1} \leftarrow \mathbf{D}_{1:n} \cup \{(\mathbf{x}_{n+1}, y_{n+1})\}$;
  $n \leftarrow n+1$;
end
Result: $(\mathbf{x}^+, y^+) \in \mathbf{D}_{1:n} : y^+ = \max_{i=1,\ldots,n} \{y_i\}$

3.1 Gaussian Process Regression

A GP is a distribution over functions, completely specified by its mean and covariance (aka *kernel*), respectively denoted with $\mu(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$. A sample drawn from a GP consists in a collection of random variables, any finite number of which have a joint Gaussian distribution. Denote with $\bar{\mathbf{X}}_{1:q} = \{\bar{\mathbf{x}}_i\}_{i=1:q}$ a set of sampling locations, then a sample from the GP will be given by: $\widehat{f}(\bar{\mathbf{X}}_{1:q}) \sim \mathcal{N}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ (i.e., sampling from a Multivariate Normal (MVN) distribution).

The covariance function has a crucial role in GP modelling, because it implies a specific prior distribution over GP samples, drastically changing their smoothness. Many kernels are available as possible covariance functions (Gramacy 2020; Archetti and Candelieri 2019; Williams 2006), with each one having at least one internal hyperparameter to set up. Using a GP as a regression model requires to conditioning its mean and covariance to the available set of observations. While the type of covariance – i.e., the kernel – is chosen by the user, its hyperparameters are tuned by reducing the error between actual data and GP's predictions. This is usually done by Maximum Likelihood Estimation (MLE) or Maximum-a-Posteriori (MAP).

In BO the "dataset" on which training the GP consists of the observations collected so far, that is $\mathbf{D}_{1:n} = (\mathbf{X}_{1:n}, \mathbf{y}_{1:n})$. The conditioned (aka *trained*) GP can be then used to make predictions for any $\mathbf{x} \in \Omega$, according to the following two equations:

$$\mu(\mathbf{x}) = \mathrm{k}(\mathbf{x}, \mathbf{X}_{1:n}) \left[\mathbf{K}(\mathbf{X}_{1:n}, \mathbf{X}_{1:n}) - \lambda^2 \mathbf{I}\right]^{-1} \mathbf{y}_{1:n} \tag{3}$$

$$\sigma(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathrm{k}(\mathbf{x}, \mathbf{X}_{1:n}) \left[\mathbf{K}(\mathbf{X}_{1:n}, \mathbf{X}_{1:n}) - \lambda^2 \mathbf{I}\right]^{-1} \mathrm{k}(\mathbf{X}_{1:n}, \mathbf{x}) \tag{4}$$

where $\mu(\mathbf{x})$ is the prediction for $f(\mathbf{x})$ and $\sigma(\mathbf{x})$ is the uncertainty associated to the prediction. Moreover, $\mathrm{k}(\mathbf{x}, \mathbf{X}_{1:n})$ is an n-dimensional vector such that its i-th component is given by $k(\mathbf{x}, \mathbf{x}_i)$ – with $\mathbf{x}_i \in \mathbf{X}_{1:n}$ – and $\mathbf{K}(\mathbf{X}_{1:n}, \mathbf{X}_{1:n})$ is an $n \times n$ matrix whose entry $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

It is important to remark that equations (3) and (4) refer to the GP model updated at current iteration and are therefore to be intended as $\mu_n(\mathbf{x})$ and $\sigma_n(\mathbf{x})$. Figure 1 shows a 1-dimensional example of a GP trained on 6 observations and approximating $f(\mathbf{x})$. As informally explained, it can be noticed how (predictive) uncertainty increases with the distance from observations.

3.2 Random Forest

Random Forest (RF) is an *ensemble learning* method whose training algorithm consists into generating a multitude of Decision Trees (DTs) by combining *bagging* – to sample a subset from the available dataset

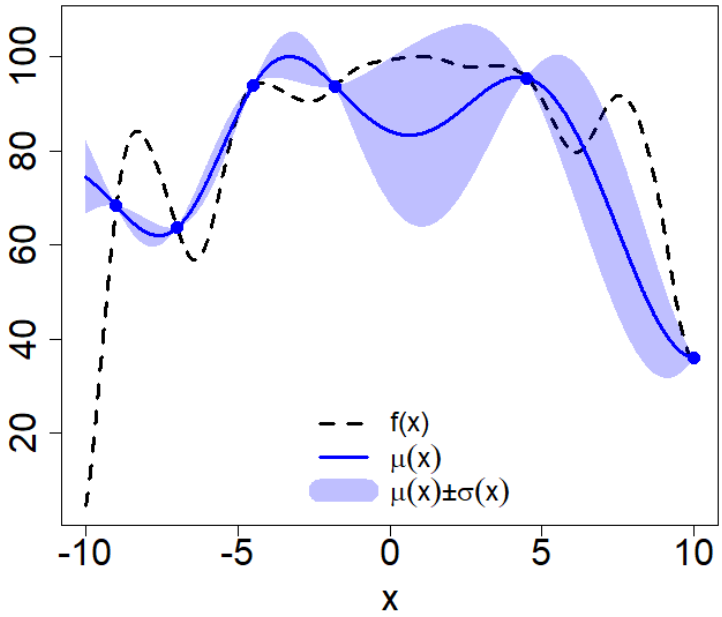

Figure 1: A simple 1-dimensional example (i.e., $\Omega = [-10, 10]$) showing a GP approximating $f(\mathbf{x})$ depending on 6 observations. Predictive mean, $\mu(\mathbf{x})$, and standard deviation, $\sigma(\mathbf{x})$, are depicted.

– and *random feature selection*. More precisely, every DT of the forest is trained on a different sample of the dataset and features. Injecting randomness simultaneously with bagging and random feature selection works surprisingly well for almost any kind of problems, allowing for generating a collection of DTs with controlled variance. As far as inference is concerned, an RF's prediction is computed as an aggregation of the individual trees' predictions. In the case of a RF regression model, mean and median are the most common aggregation operators. Moreover, being an ensemble of different regression models, it is also possible to compute the standard deviation of the individual trees' predictions. Thus, just like for GP, both RF's predictive mean, $\mu(\mathbf{x})$, and standard deviation, $\sigma(\mathbf{x})$, are available, making RF a suitable probabilistic surrogate model for BO, where the training dataset is $\mathbf{D}_{1:n} = (\mathbf{X}_{1:n}, \mathbf{y}_{1:n})$ and the features are the dimensions spanning the search space Ω.

While GP is best suited to model smooth objective functions in a search space spanned by continuous variables, it cannot "naturally" deal with discrete, mixed and conditional variables. A taxonomy about possible "workarounds" is presented in (Garrido-Merchán and Hernández-Lobato 2020), while in (Ru et al. 2020) a GP's kernel specifically defined to deal with mixed continuous and categorical variables is proposed, leading to the so-called CoCaBO algorithm (**Co**ntinuous-**Ca**tegorical **B**ayesian **O**ptimization).

Contrary to GP, RF can naturally deal with discrete, mixed and conditional variables. This has motivated the wide and successful adoption of RF in the Automated Machine Learning (AutoML) setting (He et al. 2021; Hutter et al. 2019). Indeed, searching for the best ML algorithm along with the optimal configuration of its hyperparameters, given a dataset, requires to optimize a black-box, potentially multi-extremal and largely expensive – in terms of computational resources and time – performance metric (e.g., accuracy on k-fold cross validation), exactly the problem (1). However, ML algorithms are characterized by mixed and conditional hyperparameters: an example of a discrete hyperparameter is the activation function of the units of an Artificial Neural Network. An example of a *conditional* hyperparameter is the number of units in the i-th layer of an Artificial Neural Network: this hyperparamenter is "active" if and only if the value of another hyperparameter "number-of-layers" is equal to or greater than i (because otherwise the i-th layer of the Artificial Neural Network does not exist). It is easy to understand how much complicated the search

space can be in the case of an AutoML task, with conditional hyperparameters implying a hierarchical organization among the search space's dimensions.

RF can be adopted as a probabilistic surrogate model also in the case that Ω has continuous dimensions. Differences with respect to a GP are clearly depicted in Figure 2: discontinuity in the RF probabilistic surrogate model is implied by the underlying ensemble of DTs of the forest. There is another relevant difference with GP regression: the RF's predictive mean and standard deviation are black-box, because they are computed as aggregation of the individual DTs' predictions. As better detailed in Section 4, working with a discontinuous and black-box probabilistic surrogate model has an impact on how the acquisition function is optimized to select the next location to query.

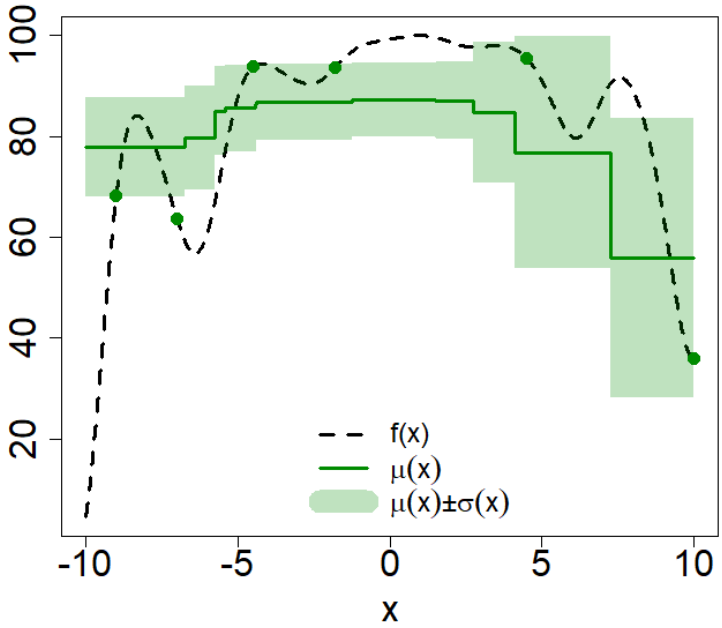

Figure 2: A simple 1-dimensional example (i.e., $\Omega = [-10, 10]$) showing an RF approximating $f\mathbf{x}$) depending on 6 observations. Predictive mean, $\mu((x)$, and standard deviation, $\sigma(\mathbf{x})$, are depicted.

3.3 Other Approaches

Following the idea of replacing the GP with an RF, any kind of probabilistic regression model or ensemble – as well as boosting or bagging – of (deterministic) regression models can be used as an alternative to GP as well as RF.

In Snoek et al. (2015) a Deep Learning model having a Bayesian linear regression layer as the last hidden layer is proposed with the aim to replace GP with a model that scales better but retains most of the GP's properties. Indeed, the resulting model is an *adaptive basis regression*, a statistical technique which scales linearly in the number of observations, and cubically in the basis function dimensionality. This allows to explicitly manage the trade-off between computational time and model capacity. The resulting algorithm, named DNGO, has been extensively tested on hyperparameters optimization of Convolutional Neural Networks. Empirical results show that DNGO provides the same modelling properties of a GP but with a significantly lower computational cost. In Springenberg et al. (2016), a Bayesian Neural Network has been suggested as an alternative to GP. The algorithm is called BOHAMIANN (Bayesian Optimization with HAMIltonian Artificial Neural Network) and has been tested with a three layers neural network with

50 tanh units.

Another motivation to replace GP with other modelling strategies is to overcome the limitation implied by the underlying *stationarity* assumption. Indeed, GP assumes the same kernel to be used throughout the entire search space Ω. An assumption which might be not desirable in many real-world problems. Different approaches are possible, from using non-stationary kernels (Higdon et al. 1999; Schmidt and O'Hagan 2003) to Treed-GP (Gramacy and Lee 2008; Civera et al. 2017; Civera et al. 2020), aimed at partitioning the search spaces into sub-regions – via DT regression – and then training a stationary GP model specifically for each sub-region model. Another possibility is to use Deep Gaussian Processes, as recently proposed in (Hebbal et al. 2019).

4 ACQUISITION FUNCTION

A plethora of acquisition functions has been proposed (Archetti and Candelieri 2019; Shahriari et al. 2015), offering different trade-off mechanisms between exploration and exploitation. This tutorial proposes – for the first time, at the author knowledge – an organization of acquisition functions into two families: the "mean-variance" and the "sampling-based". The acquisition functions belonging to the former family can be applied to whichever probabilistic surrogate model. On the contrary, those belonging to the latter family require to perform a sampling procedure, which is available only for some modelling strategy, such as GP regression modelling.

4.1 Mean-Variance Acquisition Functions

All the acquisition functions belonging to this family works by only considering the predictive mean, $\mu(\mathbf{x})$, and the standard deviation, $\sigma(\mathbf{x})$, of the probabilistic surrogate model. Therefore, any model providing these two components – analytically as well as black-box – can be used to compute the following acquisition functions. The common idea is to include an exploration "bonus" – aka uncertainty "bonus" – to the "exploitative" choice based on the predictive mean only.

Expected Improvement (EI) (Jones et al. 1998; Mockus et al. 1978) measures the expectation of the improvement over the current best observed value (aka *best seen*) $y^+ = \max_{i=1,..,n}\{y_i\}$, depending on the predictive distribution of the probabilistic surrogate model:

$$\alpha_{EI}(\mathbf{x}) = \left(\mu(\mathbf{x}) - y^+\right)\Phi(z) + \sigma(\mathbf{x})\phi(z)$$

if $\sigma(\mathbf{x}) > 0$, $\alpha_{EI}(\mathbf{x}) = 0$ otherwise, where $\Phi(z)$ and $\phi(z)$ are the probability distribution and cumulative distribution of the standard normal, respectively, and $z = \frac{\mu(\mathbf{x}) - y^+}{\sigma(\mathbf{x})}$ if $\sigma(\mathbf{x}) > 0$ and $z = 0$ otherwise.

The first term in $\alpha_{EI}(\mathbf{x})$ increases with the predictive mean decreasing, while the second term increases with the predictive uncertainty increasing. Thus, this acquisition function, in a sense, automatically balances between exploitation and exploration. To further increase exploration, an additional hyperparameter, ξ, can be added into EI equation, also replacing $(\mu(\mathbf{x}) - y^+)$ with $(\mu(\mathbf{x}) - y^+ - \xi)$ both in $\alpha_{EI}(\mathbf{x})$ and z equations. However, setting a suitable value for ξ is difficult, because it depends on the codomain of $f(\mathbf{x})$, which is unknown a-priori being $f(\mathbf{x})$ black-box.

A different solution is given in (Preuss and Von Toussaint 2018), which proposes to deterministically alternates between maximization of EI and maximization of GP's predictive standard deviation (i.e, uncertainty) to switch between exploitative and explorative decisions, reporting successful results.

Upper Confidence Bound (UCB) (Srinivas et al. 2012) is an acquisition function that manages exploration-exploitation by being *optimistic in the face of uncertainty*:

$$\alpha_{UCB}(\mathbf{x}) = \mu(\mathbf{x}) + \sqrt{\beta}\sigma(\mathbf{x})$$

Also UCB consists of two terms, representing – even in a more clear way – the "trust in predictions" (i.e., $\mu(\mathbf{x})$, exploitation) and the "uncertainty bonus" (i.e., $\sigma(\mathbf{x})$, exploration). As in EI, an hyperparamenter allows to increase/decrease the relevance of the uncertainty bonus, but in this case it is just a multiplier and it is therefore easier to identify a suitable value for it. Specifically, in (Srinivas et al. 2012) a logarithmic scheduling is provided with a convergence proof, under a limited budget of queries. However, (Berk et al. 2020) has recently obtained better performances by randomly sampling β from a given distribution, proving that this allows to identify more suitable β values and to outperform the original UCB on a range of synthetic and real-world problems.

Figure 3 shows differences between EI and UCB, also with respect to underlying the probabilistic surrogate model, specifically GP and RF. With respect to the GP, the next location selected according to EI is more biased towards exploitation than that chosen according to UCB. More precisely, in the case of EI, $\mathbf{x}_{n+1}$ is close to the the location associated to the best function evaluation observed so far, while UCB gives some more chance to exploration offering, consequently, a better global search mechanism. On the contrary, in the case of RF there is not any difference in the choice of the next location $\mathbf{x}_{n+1}$ between adopting EI or UCB.

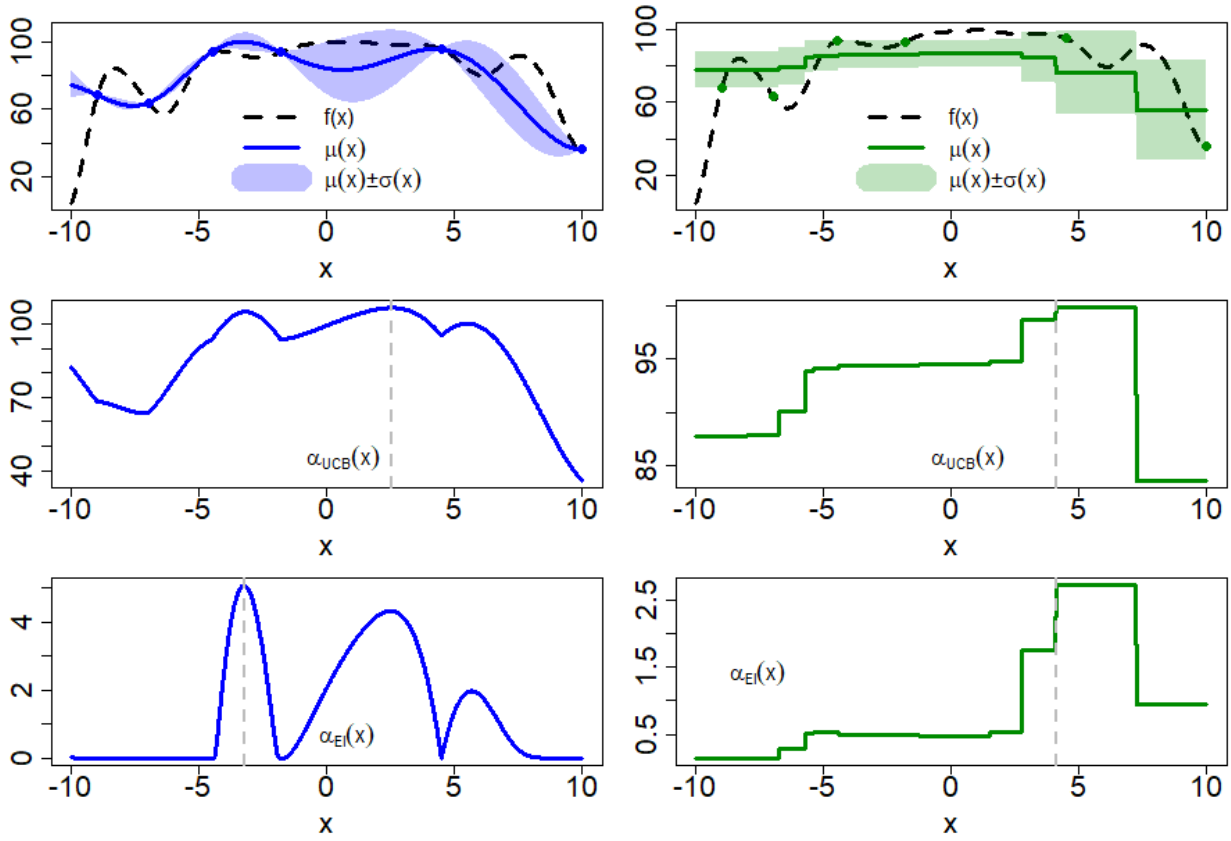

Figure 3: An example showing the differences between $\alpha_{EI}(\mathbf{x})$ and $\alpha_{UCB}(\mathbf{x})$, and the associated next query location $\mathbf{x}_{n+1}$ (vertical grey lines), depending on the probabilistic surrogate model (GP – on the left – and RF – on the right).

Recently, some papers proposed to address the selection of the next location to query as a bi-objective optimization problem: maximizing the predictive mean (exploitation) while maximizing uncertainty (ex-

ploration). For instance, Žilinskas and Calvin (2019) considers global optimization with respect to the theory of rational decision making under uncertainty. The important outcome is that EI results a special case of the bi-objective optimization setting, because it lays on the Pareto frontier of all the predictive mean and standard deviation pairs computed for – theoretically – every possible location in Ω. Moreover, taking a decision by randomly sampling from the Pareto frontier, as proposed in De Ath et al. (2021), De Ath et al. (2020) empirically proved to outperform other acquisition functions: the main motivation is that the Pareto frontier offers a significantly larger set of Pareto-efficient solutions than single-objective acquisition functions like EI and UCB.

4.1.1 Sampling-based Acquisition Functions

The most relevant difference between GP modelling and other alternative probabilistic surrogate models is that GP is a distribution and it is possible to drawn sample from it (as already described in Section 3). This consideration has enabled the design of many acquisition functions that, contrary to those belonging to the "mean-variance" family, can only be computed for a GP-based surrogate model. The most simple "sampling-based" acquisition function consists into drawing a sample from the GP trained on the current set of observations, and selecting $\mathbf{x}_{n+1}$ as the location maximizing this sample. This simple procedure makes BO collapsing into **Thompson Sampling** (TS), which is a sequential optimization method by itself. Recently, Russo and Van Roy (2016) has performed an analysis on TS, proving that it is biased towards exploitation and proposing an ε-greedy step to give more chance to exploration, leading to a more effective exploration-exploitation trade-off.

BO research has further extended the TS approach by considering to drawn more than one sample at each BO iteration, with the aim to estimate the location of the maximizer, instead of the value of the maximum. This means that the reference problem becomes $\arg\max_{\mathbf{x}\in\Omega} f(\mathbf{x})$ instead of $\max_{\mathbf{x}\in\Omega} f(\mathbf{x})$.

This paradigm shift has led to acquisition functions such as **Entropy Search** (ES) (Hennig and Schuler 2012), **Predictive Entropy Search** (PES) (Henrández-Lobato et al. 2014), **Max-Value Entropy Search** (MES) (Wang and Jegelka 2017), and **Knowledge-Gradient** (KG) (Frazier 2018; Scott et al. 2011).

Although these acquisition functions proved to improve sample-efficiency of the BO framework, they rely on GP sampling, that is sampling from a MVN distribution, a really expensive procedure whose computational cost increases with the dimensionality of the search space Ω. Furthermore, they can be applied on GP-based surrogate models, only.

4.1.2 Optimizing the Acquisition Function

At this point it should be clear that solving the auxiliary problem (2) depends on both the specific acquisition function considered and the probabilistic modelling strategy adopted. More specifically, the solver to use depends on the specific regression algorithm: due to the black-box and piecewise-constant nature of predictive mean $\mu(\mathbf{x})$ – and standard deviation $\sigma(\mathbf{x})$ – of a RF regression model, derivative-free global optimization solvers are needed, such as Adaptive Random Search (Zabinsky 2013; Zhigljavsky 2012) or Evolutionary methods (Tzanetos et al. 2020; Simon 2013). Although these methods requires a huge number of function evaluations to find a global optimum, it is important to remark that they are applied on $\alpha(\mathbf{x})$, which is not query-expensive – contrary to $f(\mathbf{x})$ – but only black-box and potentially multi-extremal.

Analogously to RF, any other ensemble/bagging/boosting based regression model will require to use a derivative-free algorithm. On the contrary, when the probabilistic surrogate model is a GP, both derivative-free and gradient-based algorithms can be used, specifically in the case of mean-variance acquisition functions.

4.2 Monitoring the Optimization Process

Contrary to gradient-based methods, BO does not guarantee an improvement of the current solution from an iteration to the next. This is basically due to the need to balance – at each iteration – between exploitation and exploration as well as to the fact that the surrogate model is just an approximation of the actual objective function.

Monitoring the evolution of the BO process over the sequentially performed function evaluations is crucial, especially when querying the objective function is significantly expensive (in terms of time and/or resources). For instance, one could think to adopt an early stopping criterion in the case that there is not any improvement after a fixed number of consecutive BO iterations.

A common practice consists in plotting the value of best seen at each iteration, y_n^+, with $n = 1, ..., N$ and N the overall number of function evaluations, that is best value observed so far, $y_n^+ = \max_{i=1,...,n} \{y_i\}$. Figure 4 shows an example related to the optimization of the same function reported in the previous pictures, with both GP and RF used as a surrogate model, and each one combined with EI and UCB, separately.

The value at iteration 0 is the best value observed on the so called *initial design*, that is a set of locations randomly chosen just to initialize the surrogate model. Common choices for selecting these initial locations are Latin Hypercube Sampling or Uniform Sampling, with a minimum sample size equal to the number of the search space's dimensions plus one. Due to this randomness in its initialization, BO can converge to a different optimal solution, even keeping fixed the surrogate model, the acquisition function, and the overall number of function evaluations. One of the best practice adopted in almost all the research studies is to perform multiple independent runs, starting from different initial designs, and then plotting the average and standard deviation of the best seen at each BO iteration.

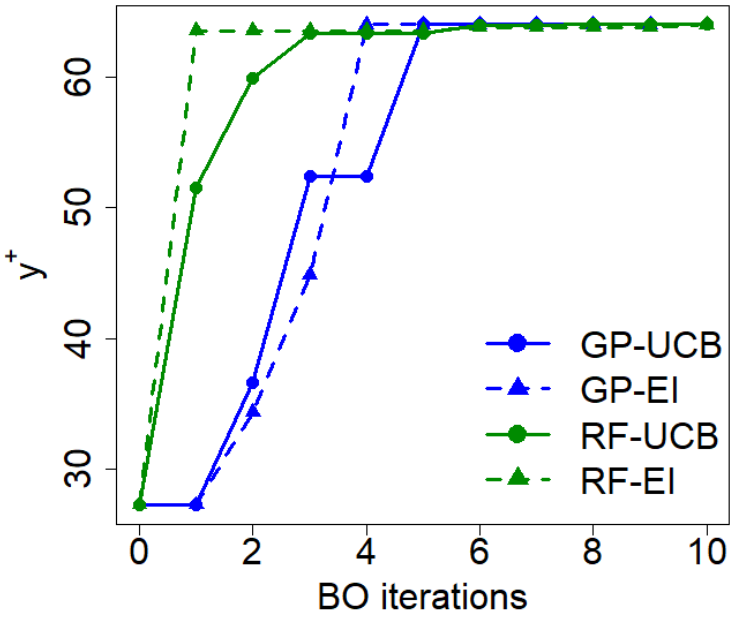

Figure 4: Best seen for four BO's settings, separately, different for surrogate model (GP or RF) and acquisition function (EI or UCB).

5 TOOLS AND APPLICATIONS

Research on BO has been largely active during last years, leading to several software tools and libraries, both open-source and commercial. Here, some of the most relevant ones are reported – the list cannot be exhaustive.

Spearmint, is a Bayesian optimization library written in Python under a collaboration between machine learning researchers at Harvard University and the University of Toronto.

Sequential Model-based Algorithm Configuration (SMAC3) Hutter et al. (2011) is a library for optimizing algorithm hyperparameters, originally written Java (SMAC) then reimplemented in Python 3 (SMAC3). This library is developed by the AutoML Group of theUniversity of Freiburg.

Robust Bayesian Optimization (RoBo) Klein et al. (2017) is a library written in Python and maintained by the AutoML Group of the University of Freiburg. Its structure is modular, allowing to easily add and exchange components such as surrogate models and acquisition functions.

Scikit-Optimize is a modular BO library written in Python, by the same authors of the most famous Python machine learning library named Scikit-Learn.

mlrMBO Bischla et al. (2017) is a flexible and comprehensive R toolbox for BO. mlrMBO is designed in a modular style, so each component can easily be replaced or adapted to specific use cases.

BayesOpt Martinez-Cantin (2014) is a library written in C++ extremely efficient, portable and flexible. It offers common interfaces for several programming languages C, C++, Python, MATLAB and Octave.

As far as commercial solutions are concerned, **SigOpt** is one of the most interesting and successful examples of *Bayesian optimization as a Service*. The core of this service was initially created for a project named MOE (Metrics Optimization Engine) by the Cornell University. The user has to interact through REST API in several languages like Java, R and Python with the most updated version of each language. The main idea behind the optimization service is that the user is totally "blind" with respect to the computational architecture infrastructure running the BO process. To use this service, the user has to subscribe an Academia or Enterprise account. Other two *Bayesian optimization as a service* solutions are **OPTaaS** – accessible via API in R and Python and published by the company Mind Foundry – and **VUKU** – produced by PROWLER.io company.

All these libraries and services can be considered as *general-purpose* BO tools, and have been largely adopted to develop specific applications. The number of domains in which BO has been – and is still – adopted are so many that is quite impossible to provide a complete overview. In the following a selection of the most relevant ones is reported.

BO has been successfully adopted for **structural design**, especially in aeronautics, such as in Chunna and Qifeng (2019), Palar et al. (2019), and Chaudhuri et al. (2019).

Another relevant application domain is **drug design/discovery**. Traditionally, this problem was addressed through high-throughput screening (HTS) for measuring *in vitro* the effect of hundreds of chemical reactions. HTS has largely failed to meet the initial expectations, leading to the development of computational techniques enabling *virtual* HTS. Advancements in computational chemistry have made it possible to compute *in silico* properties of pharmacological interest for a certain molecule. The large search space of possible candidate molecules and the black-box nature of the objective function made BO best suited for solving this optimization problem, with enormous cost-savings in the discovery phase thanks to the BO's sample efficiency (Griffiths and Hernández-Lobato 2020; Meldgaard et al. 2018; Pyzer-Knapp 2018).

Analogously, BO has been also used for **pharmaceutical product development** to reduce the number of experiments required to obtain the optimal formulation and process parameters (Sano et al. 2020).

Another domain characterized by optimization problems quite similar to those of drug design/discovery is **new material design**, examples of BO applications are reported in Packwood (2017) and Lookman et al. (2019), where possible combinations of components are so many and the cost for producing and testing a new single material are so high that a sample efficient search strategy is the only reasonable approach.

Automated control: among the many applications in which BO provides data efficient auto-tuning of control devices, a typical example is tuning a parameters of a Proportional Integral Derivative (PID) controller (Schillinger et al. 2017; Neumann-Brosig et al. 2019). Other examples are related to the optimal control of complex cyberphysical systems, such as water distribution networks (Candelieri et al. 2020; Candelieri et al. 2018).

Another relevant application domain which has recently taken advantages of BO is **Finance**. In Gonzalvez et al. (2019) GP-based BO is adopted to optimize, over time, the parameters of a simple trend-following trading strategy, to make it adaptive with the aim to produce larger returns compared against its basic implementation.

A specific application domain which has been largely exploiting BO is AutoML, leading to specific software libraries and services for addressing Hyperparameter Optimization (HPO), Combined Algorithm Selection and Hyperparameter optimization (CASH) and, more recently, Neural Architecture Search (NAS).

Among the most relevant tools one can consider **Auto-WEKA**, **Auto-sklearn** and **Auto-PyTorch** – all developed by the AutoML Group – as well as, **AutoKeras**, **Hyperopt**, **Google AutoML Tables**, **Amazon SageMaker Autopilot** and **Azure Automated Machine Learning**.

6 RECENT ADVANCES

In this section the most recent advances in BO are summarized. One of the areas that has been investigated is related to **BO under unknown constraints**, where the reference problem becomes:

$$\begin{aligned} \max_{\mathbf{x}\in\Omega} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \geq 0, i = 1,...,m \end{aligned}$$

where constraints $g_i(\mathbf{x})$ are black-box and expensive to query as $f(\mathbf{x})$. Most of the proposed approaches make two assumptions: (*i*) constraints are *decoupled* with respect to the objective function, meaning that feasibility can be evaluated separately from $f(\mathbf{x})$, and (*ii*) the constraints are statistically independent among them and on the objective function. In Gramacy and Lee (2011) the *integrated expected conditioned improvement* (IECI) approach is proposed, basically extending EI in order to also deal with feasibility. In Basudhar et al. (2012) a Probabilistic Support Vector Machine (PSVM) is used to calculate the so-called *probability of feasibility* and the optimization scheme alternates between a global search for the optimal solution, depending on both this probability and the estimated value of the objective function – modelled through a GP – and a local refinement of the PSVM through an adaptive local sampling scheme. In Gardner et al. (2014) a penalty approach has been considered where a penalty is assigned to the acquisition function in case of infeasibility, with the aim to move away from infeasible regions. Analogously to EI, also an extension for PES has been proposed, namely Predictive Entropy Search with Constraints (PESC) in Hernández-Lobato et al. (2015).

A related topic is the optimization of a *partially defined* objective function, also known as optimization *under crash constraints* or *over non-computable domains*. Here the constraints are associated to the computability/observability of the objective function (Sacher et al. 2018; Bachoc et al. 2019): if a location $\mathbf{x}$ violates one of the unknown constraints $g_i(\mathbf{x})$ then it is impossible to observe $f(\mathbf{x})$. A recent approach has been proposed in Candelieri (2019) which overcomes the limitations related to the assumptions of the previous research works. The approach is named SVM-CBO (Support Vector Machine based Constrained

BO) and is organized in two phases: the first is aimed to provide a first estimate of the feasible/computable region in Ω (i.e., feasibility determination) and the second is BO performed within this estimated region, only. Recently, extension of this approach has been proposed to optimize the hyperparameters of an Artificial Neural Network, while satisfying constraints related to its deployability on a *tiny* device, more precisely a Micro Controller Unit (MCU) having limited hardware resources (i.e., RAM and ROM) (Perego et al. 2020). This lead to the so-called **AutoTinyML** framework.

Analogously to resource constraints for *tiny-ML*, resource-efficiency has been becoming crucial for AutoML. As it typically runs on large – often cloud-based – computational platforms, it is "energivorous" and, consequently, also a significant emitter of CO_2. This issue makes AutoML strictly linked to the **Red-AI** (Dhar 2020) and **Green-AI** (Schwartz et al. 2019) topics. The analysis over 60 papers from top conferences (i.e., the 2018 Annual Meeting of the Association for Computational Linguistics, the 2018 Conference on Neurological Information Processing Systems (NeurIPS), and the 2019 Conference on Computer Vision and Pattern Recognition) has concluded that almost all the studies have prioritized accuracy over efficiency. As main result, the authors report that the total cost of producing accurate ML models increases linearly with (*i*) the cost of executing the model on a single example, (*ii*) the size of the training dataset and (*iii*) the number of AutoML experiments, which controls how many times the model is trained on the dataset. This warning has been recently received by the scientific community, especially after the astonishing results reported in Strubell et al. (2019), which has analysed the training process of many Natural Language Processing (NLP) models to estimate the energy cost in kilowatts required. When these figures are converted into approximate carbon emissions it comes out that the carbon footprint of training a single large NLP model is equal to around 300.000 kg of carbon dioxide emissions, that is the amount of CO_2 emitted by 125 round-trip flights between New York and Beijing or, equivalently, five American average cars in their lifetimes, including their manufacturing processes.

Indeed, a promising recent research direction for AutoML has been focused **multi-objective optimization**, in which measures of resource efficiency are used as objectives along with the accuracy of the trained models data (Elsken et al. 2018; Dong et al. 2018; Zhou et al. 2018).

As far as the BO research community is concerned, recent advances in the multi-objective optimization framework have been proposed, such as in Belakaria and Deshwal (2019), Belakaria et al. (2020), Paria et al. (2020),and Shu et al. (2020).

Multiple Information Source Optimization (MISO) is the setting arising when the problem (1) can be solved by using a set of less expensive approximations of $f(\mathbf{x})$, namely *information sources*, each own having its own specific *query cost*. The final goal of MISO is to solve (1) while keeping low the overall *cumulated query cost*. When the different sources come with an explicit information about their quality of approximation, usually named *fidelity*, MISO specializes in *multi-fidelity optimization* (Kennedy and O'Hagan 2000). Knowledge about fidelities can be exploited to sort hierarchically the sources leading to efficient and effective multi-fidelity optimization methods (Peherstorfer et al. 2017; Sen et al. 2018; Marques et al. 2018; Kandasamy et al. 2019).

(Lam et al. 2015) first addressed the situation where fildelity of each source is unknown and can change over the search space. The approach uses a separate GP for each information source and then *fuse* their predictions – and associated uncertainties – through the method proposed by (Winkler 1981), which came to represent the standard practice for the fusion of normally distributed data. Successively, (Poloczek et al. 2017) proposed to use a GP to capture the model discrepancy of each information source with respect to $f(\mathbf{x})$, while a single statistical model is used to perform BO jointly on the search space and the information sources. A kernel able to deal with both location and source is used to exploit correlations across different information sources. This allows reducing the uncertainty on all the information sources whenever a new function evaluation is performed, even if it comes from a less accurate source. Recently, Ghoreishi and Allaire (2019) has further extended these approaches by adapting KG to work in the MISO setting and also

considering black-box constraints. The main drawback in fusing GPs – that is the technique underlying all these MISO methods – is that it requires the computation of correlations according to an additional set of points, randomly selected, whose size affects both the computational cost and the smoothness of the resulting fused GP. More recently, a completely different approach has been proposed in Candelieri and Archetti (2021b), based on GP sparsification (Schreiter et al. 2016), instead of fusion. The idea is to select only "reliable" observations collected on the approximating sources to "augment" the set of those collected by querying $f(\mathbf{x})$. The model trained on this augmented set is named *Augmented Gaussian Process* (AGP). Selection of augmenting observations is based on a simplified discrepancy measure and GPs' predictive uncertainties. A UCB based acquisition function is also proposed, accounting also for cost of each source.

MISO is also linked to the Green-AI topic: in AutoML information sources can be small portions of a large dataset, used with the aim to moving towards the best hyperparameters configuration of a ML algorithm while reducing time, energy and costs for training each model. An example of **MISO for Green-AI** has been recently proposed in Candelieri et al. (2021).

MISO approaches are based on the assumption that each source has its own query cost and that it is constant throughout the entire search space. However, recent research papers have been giving evidence of problems where the cost of each source is location dependent, such as hyperparameter optimization of a ML algorithm. This has been leading to the recent research topic known as **cost-aware BO** (Candelieri and Archetti 2021a; Abdolshah et al. 2019; Lee et al. 2020; Guinet et al. 2020), where not only the objective function – and cheap sources, in the MISO setting – but also the cost function has to be modelled according to the cost incurred for querying the locations selected along the sequential optimization process.

7 CONCLUSIONS

Bayesian optimization has been summarized in this tutorial, discussing about different possible choices regarding the probabilistic surrogate model and the acquisition function. An overview on tools and applications has been given, remarking the relevance and effectiveness of BO in solving practical problems and enabling disruptive innovations such as Automated Machine Learning and Green-AI. Many challenging – and still open – research directions have been discussed, which will require, in the author's opinion, further investigation during the next years. Stimulating research questions have been coming from application domains, leading to innovative solutions, each addressing a different topics, such as unknown constraints, multi-objective and multi-information source optimization. The future will for sure ask research to a unifying BO approach, over the different considered settings.

REFERENCES

Abdolshah, M., A. Shilton, S. Rana, S. Gupta, and S. Venkatesh. 2019. "Cost-aware Multi-objective Bayesian Optimisation". *arXiv preprint arXiv:1909.03600*.

Archetti, F., and A. Candelieri. 2019. *Bayesian Optimization and Data Science*. Springer.

Bachoc, F., C. Helbert, and V. Picheny. 2019. "Gaussian Process Optimization with Failures: Classification and Convergence Proof". *Journal of Global Optimization* 78(3):483–506.

Basudhar, A., C. Dribusch, S. Lacaze, and S. Missoum. 2012. "Constrained Efficient Global Optimization with Support Vector Machines". *Structural Multidisciplinary Optimization* 46:201—221.

Belakaria, S., and A. Deshwal. 2019. "Max-value Entropy Search for Multi-objective Bayesian Optimization". In *International Conference on Neural Information Processing Systems (NeurIPS)*. NIPS 2019, December 8–14, Vancouver, Canada.

Belakaria, S., A. Deshwal, N. K. Jayakodi, and J. R. Doppa. 2020. "Uncertainty-aware Search Framework for Multi-objective Bayesian Optimization". In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 34, 10044–10052. AAAI 2020, February 7–12, New York, US.

Bemporad, A. 2020. "Global Optimization via Inverse Distance Weighting and Radial Basis Functions". *Computational Optimization and Applications* 77(2):571–595.

Berk, J., S. Gupta, S. Rana, and S. Venkatesh. 2020. "Randomised Gaussian Process Upper Confidence Bound for Bayesian Optimisation". *arXiv preprint arXiv:2006.04296*.

Bischla, B., J. Richterb, J. Bossekc, D. Hornb, J. Thomasa, and M. Langb. 2017. "mlrMBO: A Modular Framework for Model-Based Optimization of Expensive Black-Box Functions". *stat* 1050:15.

Candelieri, A. 2019. "Sequential Model-based Optimization of Partially Defined Functions under Unknown Constraints". *Journal of Global Optimization*:1–23.

Candelieri, A., and F. Archetti. 2021a. "MISO-wiLDCosts: Multi Information Source Optimization with Location Dependent Costs". *arXiv preprint arXiv:2102.04951*.

Candelieri, A., and F. Archetti. 2021b. "Sparsifying to Optimize over Multiple Information Sources: an Augmented Gaussian Process based Algorithm". *Structural and Multidisciplinary Optimization*:1–17.

Candelieri, A., B. Galuzzi, I. Giordani, and F. Archetti. 2020. "Learning Optimal Control of Water Distribution Networks Through Sequential Model-based Optimization". In *Learning and Intelligent Optimization (LION 2020) Lecture Notes in Computer Science*, 303–315: Springer, Cham.

Candelieri, A., R. Perego, and F. Archetti. 2018. "Bayesian Optimization of Pump Operations in Water Distribution Systems". *Journal of Global Optimization* 71(1):213–235.

Candelieri, A., R. Perego, and F. Archetti. 2021. "Green Machine Learning via Augmented Gaussian Processes and Multi-Information Source Optimization". *Soft Computing*:1–13.

Chaudhuri, A., A. N. Marques, R. Lam, and K. E. Willcox. 2019. "Reusing Information for Multifidelity Active Learning in Reliability-Based Design Optimization". In *AIAA Scitech 2019 Forum*, 1222. AIAA Scitech 2019, January 7–11, San Diego, US.

Chunna, L., and P. Qifeng. 2019. "Adaptive Optimization Methodology based on Kriging Modeling and a Trust Region Method". *Chinese Journal of Aeronautics* 32(2):281–295.

Civera, M., G. Boscato, and L. Z. Fragonara. 2020. "Treed Gaussian Process for Manufacturing Imperfection Identification of Pultruded GFRP Thin-walled Profile". *Composite Structures* 254:112882.

Civera, M., C. Surace, and K. Worden. 2017. "Detection of Cracks in Beams using Treed Gaussian Processes". In *Structural Health Monitoring & Damage Detection, Volume 7*, 85–97. Springer.

De Ath, G., R. M. Everson, J. E. Fieldsend, and A. A. Rahat. 2020. "ε-shotgun: ε-greedy Batch Bayesian Optimisation". In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 787–795. GECCO 2020, July 8–12, Cancun, Mexico.

De Ath, G., R. M. Everson, A. A. Rahat, and J. E. Fieldsend. 2021. "Greed is Good: Exploration and Exploitation Trade-offs in Bayesian Optimisation". *ACM Transactions on Evolutionary Learning and Optimization* 1(1):1–22.

Dhar, P. 2020. "The Carbon Impact of Artificial Intelligence". *Nat Mach Intell* 2:423–5.

Dong, J.-D., A.-C. Cheng, D.-C. Juan, W. Wei, and M. Sun. 2018. "Dpp-net: Device-aware Progressive Search for Pareto-optimal Neural Architectures". In *Proceedings of the European Conference on Computer Vision (ECCV)*, 517–531. ECCV 2018, September 8–14, Munich, Germany.

Elsken, T., J. H. Metzen, and F. Hutter. 2018. "Efficient Multi-Objective Neural Architecture Search via Lamarckian Evolution". In *International Conference on Learning Representations*. ICLR 2018, April 30 – May 3, Vancouver, Canada.

Frazier, P. I. 2018. "Bayesian Optimization". In *Recent Advances in Optimization and Modeling of Contemporary Problems*, 255–278. INFORMS.

Gardner, J., M. Kusner, Z. Xu, K. Weinberger, and J. Cunningham. 2014. "Bayesian Optimization with Inequality Constraints". In *International Conference on Machine Learning*, 937—945. ICML 2014, June 21–26, Bejing, China.

Garrido-Merchán, E. C., and D. Hernández-Lobato. 2020. "Dealing with Categorical and Integer-valued Variables in Bayesian Optimization with Gaussian Processes". *Neurocomputing* 380:20–35.

Ghoreishi, S. F., and D. Allaire. 2019. "Multi-Information Source Constrained Bayesian Optimization". *Structural and Multidisciplinary Optimization* 59(3):977–991.

Glover, F., and M. Samorani. 2019. "Intensification, Diversification and Learning in Metaheuristic Optimization". *Journal of Heuristics* 25(4):517–520.

Gonzalvez, J., E. Lezmi, T. Roncalli, and J. Xu. 2019. "Financial Applications of Gaussian Processes and Bayesian Optimization". *arXiv preprint arXiv:1903.04841*.

Gramacy, R., and H. Lee. 2011. "Optimization under Unknown Constraints". *Bayesian Statistics* 9.

Gramacy, R. B. 2020. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. Chapman and Hall/CRC.

Gramacy, R. B., and H. K. H. Lee. 2008. "Bayesian Treed Gaussian Process Models with an Application to Computer Modeling". *Journal of the American Statistical Association* 103(483):1119–1130.

Griffiths, R.-R., and J. M. Hernández-Lobato. 2020. "Constrained Bayesian Optimization for Automatic Cchemical Design using Variational Autoencoders". *Chemical science* 11(2):577–586.

Guinet, G., V. Peronne, and C. Archambeau. 2020. "Pareto-efficient Acquisition Functions for Cost-Aware Bayesian Optimization". *arXiv preprint arXiv:2011.11456*.

He, X., K. Zhao, and X. Chu. 2021. "AutoML: A Survey of the State-of-the-Art". *Knowledge-Based Systems* 212:106622.

Hebbal, A., L. Brevault, M. Balesdent, E. G. Talbi, and N. Melab. 2019. "Bayesian Optimization using Deep Gaussian Processes". *arXiv preprint arXiv:1905.03350*.

Hennig, P., and C. J. Schuler. 2012. "Entropy Search for Information-Efficient Global Optimization.". *Journal of Machine Learning Research* 13(6).

Henrández-Lobato, J. M., M. W. Hoffman, and Z. Ghahramani. 2014. "Predictive Entropy Search for Efficient Global Optimization of Black-box Functions". In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 1*, 918–926. NIPS 2014, December 8–13, Montréal, Canada.

Hernández-Lobato, J. M., M. Gelbart, M. Hoffman, R. Adams, and Z. Ghahramani. 2015. "Predictive Entropy Search for Bayesian Optimization with Unknown Constraints". In *International Conference on Machine Learning*, 1699–1707. ICML 2015, July 6–11, Lille, France.

Higdon, D., J. Swall, and J. Kern. 1999. "Non-stationary Spatial Modeling". *Bayesian statistics* 6(1):761–768.

Ho, T. K. 1995. "Random Decision Forests". In *Proceedings of 3rd international conference on document analysis and recognition*, Volume 1, 278–282. 1995, August 14–16, Montreal, Canada.

Hutter, F., H. Hoos, and K. Leyton-Brown. 2013. "An Evaluation of Sequential Model-based Optimization for Expensive Blackbox Functions". In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*, 1209–1216. GECCO 2013, July 6–10, Amsterdam, The Netherland.

Hutter, F., H. H. Hoos, and K. Leyton-Brown. 2011. "Sequential Model-based Optimization for General Algorithm Configuration". In *Learning and Intelligent Optimization (LION 2011) Lecture Notes in Computer Science*, edited by C. A. C. Coello, 507–523. Berlin, Heidelberg: Springer Berlin Heidelberg.

Hutter, F., L. Kotthoff, and J. Vanschoren. 2019. *Automated Machine Learning: Methods, Systems, Challenges*. Springer Nature.

Jones, D. R., M. Schonlau, and W. J. Welch. 1998. "Efficient Global Optimization of Expensive Black-box Functions". *Journal of Global optimization* 13(4):455–492.

Kandasamy, K., G. Dasarathy, J. Oliva, J. Schneider, and B. Poczos. 2019. "Multi-fidelity Gaussian Process Bandit Optimisation". *Journal of Artificial Intelligence Research* 66:151–196.

Kennedy, M. C., and A. O'Hagan. 2000. "Predicting the Output from a Complex Computer Code when Fast Approximations are Available". *Biometrika* 87(1):1–13.

Klein, A., S. Falkner, N. Mansur, and F. Hutter. 2017. "Robo: A flexible and Robust Bayesian Optimization Framework in Python". In *NIPS 2017 Bayesian Optimization Workshop*. 2017, December 4–9, Long Beach, US.

Lam, R., D. L. Allaire, and K. E. Willcox. 2015. "Multifidelity Optimization using Statistical Surrogate Modeling for non-hierarchical Information Sources". In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 0143. AIAA Scitech, 2015, January 5–9, Kissimmee, US.

Lee, E. H., V. Perrone, C. Archambeau, and M. Seeger. 2020. "Cost-aware Bayesian Optimization". *arXiv preprint arXiv:2003.10870*.

Lookman, T., P. V. Balachandran, D. Xue, and R. Yuan. 2019. "Active learning in Materials Science with Emphasis on Adaptive Sampling using Uncertainties for Targeted Design". *Computational Materials* 5(1):1–17.

Marques, A. N., R. R. Lam, and K. E. Willcox. 2018. "Contour Location via Entropy Reduction Leveraging Multiple Information Sources". *arXiv preprint arXiv:1805.07489*.

Martinez-Cantin, R. 2014. "BayesOpt: a Bayesian Optimization Library for nonlinear Optimization, Experimental Design and Bandits.". *J. Mach. Learn. Res.* 15(1):3735–3739.

Meldgaard, S. A., E. L. Kolsbjerg, and B. Hammer. 2018. "Machine Learning Enhanced Global Optimization by Clustering Local Environments to Enable Bundled Atomic Energies". *The Journal of chemical physics* 149(13):134104.

Mockus, J., V. Tiesis, and A. Zilinskas. 1978. "The Application of Bayesian Methods for Seeking the Extremum". *Towards global optimization* 2(117-129):2.

Neumann-Brosig, M., A. Marco, D. Schwarzmann, and S. Trimpe. 2019. "Data-efficient Autotuning with Bayesian Optimization: An Industrial Control Study". *IEEE Transactions on Control Systems Technology* 28(3):730–740.

Packwood, D. 2017. *Bayesian Optimization for Materials Science*. Springer.

Palar, P. S., Y. B. Dwianto, R. G. Regis, A. Oyama, and L. R. Zuhal. 2019. "Benchmarking Constrained Surrogate-based Optimization on Low Speed Airfoil Design Problems". In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1990–1998. GECCO 2019, July 13–17, Prague, Czech Republic.

Paria, B., K. Kandasamy, and B. Póczos. 2020. "A Flexible Framework for Multi-objective Bayesian Optimization using Random Scalarizations". In *Uncertainty in Artificial Intelligence*, 766–776. UAI 2020, August 3–6 (online).

Peherstorfer, B., B. Kramer, and K. Willcox. 2017. "Combining Multiple Surrogate Models to Accelerate Failure Probability Estimation with Expensive High-Fidelity Models". *Journal of Computational Physics* 341:61–75.

Perego, R., A. Candelieri, F. Archetti, and D. Pau. 2020. "Tuning Deep Neural Network's Hyperparameters Constrained to Deployability on Tiny Systems". In *International Conference on Artificial Neural Networks*, 92–103: Springer, Cham.

Poloczek, M., J. Wang, and P. I. Frazier. 2017. "Multi-Information Source Optimization". In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4291–4301. NIPS 2017, December 4–9, Long Beach, US.

Preuss, R., and U. Von Toussaint. 2018. "Global Optimization Employing Gaussian Process-based Bayesian Surrogates". *Entropy* 20(3):201.

Pyzer-Knapp, E. O. 2018. "Bayesian Optimization for Accelerated Drug Discovery". *IBM Journal of Research and Development* 62(6):2–1.

Ru, B., A. Alvi, V. Nguyen, M. A. Osborne, and S. Roberts. 2020. "Bayesian Optimisation over Multiple Continuous and Categorical Inputs". In *International Conference on Machine Learning*, 8276–8285. ICML 2020, July 12–18 (online).

Russo, D., and B. Van Roy. 2016. "An Information-theoretic Analysis of Thompson Sampling". *The Journal of Machine Learning Research* 17(1):2442–2471.

Sacher, M., R. Duvigneau, O. Le Maitre, M. Durand, E. Berrini, F. Hauville, and J. Astolfi. 2018. "A Classification Approach to Efficient Global Optimization in Presence of non-computable Domains". *Structural Multidisciplinary Optimization* 58(4):1537—1557.

Sano, S., T. Kadowaki, K. Tsuda, and S. Kimura. 2020. "Application of Bayesian Optimization for Pharmaceutical Product Development". *Journal of Pharmaceutical Innovation* 15(3):333–343.

Schillinger, M., B. Hartmann, P. Skalecki, M. Meister, D. Nguyen-Tuong, and O. Nelles. 2017. "Safe Active Learning and Safe Bayesian Optimization for Tuning a PI-controller". *IFAC-PapersOnLine* 50(1):5967–5972.

Schmidt, A. M., and A. O'Hagan. 2003. "Bayesian Inference for non-stationary Spatial Covariance Structure via Spatial Deformations". *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65(3):743–758.

Schreiter, J., D. Nguyen-Tuong, and M. Toussaint. 2016. "Efficient Sparsification for Gaussian Process Regression". *Neurocomputing* 192:29–37.

Schwartz, R., J. Dodge, N. A. Smith, and O. Etzioni. 2019. "Green AI". *arXiv preprint arXiv:1907.10597*.

Scott, W., P. Frazier, and W. Powell. 2011. "The Correlated Knowledge Gradient for Simulation Optimization of Continuous Parameters using Gaussian Process Regression". *SIAM Journal on Optimization* 21(3):996–1026.

Sen, R., K. Kandasamy, and S. Shakkottai. 2018. "Multi-fidelity Black-box Optimization with Hierarchical Partitions". In *International conference on machine learning*, 4538–4547. ICML 2018, July 10–15, Stockholm, Sweden.

Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. 2015. "Taking the Human out of the Loop: A Review of Bayesian Optimization". *Proceedings of the IEEE* 104(1):148–175.

Shu, L., P. Jiang, X. Shao, and Y. Wang. 2020. "A New Multi-objective Bayesian Optimization Formulation with the Acquisition Function for Convergence and Diversity". *Journal of Mechanical Design* 142(9).

Simon, D. 2013. *Evolutionary Optimization Algorithms*. John Wiley & Sons.

Snoek, J., O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams. 2015. "Scalable Bayesian Optimization Using Deep Neural Networks". In *International conference on machine learning*, 2171–2180. ICML 2015, July 6–11, Lille, France.

Springenberg, J. T., A. Klein, S. Falkner, and F. Hutter. 2016. "Bayesian Optimization with Robust Bayesian Neural Networks". *Advances in neural information processing systems* 29:4134–4142.

Srinivas, N., A. Krause, S. M. Kakade, and M. W. Seeger. 2012. "Information-theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting". *IEEE Transactions on Information Theory* 58(5):3250–3265.

Strubell, E., A. Ganesh, and A. McCallum. 2019. "Energy and Policy Considerations for Deep Learning in NLP". In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650. ACL 2019, July 28 – August 2, Florence, Italy.

Tzanetos, A., I. Fister Jr, and G. Dounias. 2020. "A Comprehensive Database of Nature-Inspired Algorithms". *Data in Brief* 31:105792.

Wang, Z., and S. Jegelka. 2017. "Max-value Entropy Search for Efficient Bayesian Optimization". In *International Conference on Machine Learning*, 3627–3635. ICML 2017, August 6–11, Sydney, Australia.

Williams, C. K. 2006. *Gaussian Processes for Machine Learning*. Taylor & Francis Group.

Winkler, R. L. 1981. "Combining Probability Distributions from Dependent Information Sources". *Management Science* 27(4):479–488.

Zabinsky, Z. B. 2013. *Stochastic Adaptive Search for Global Optimization*, Volume 72. Springer Science & Business Media.

Zhigljavsky, A. A. 2012. *Theory of Global Random Search*, Volume 65. Springer Science & Business Media.

Zhou, Y., S. Ebrahimi, S. Ö. Arık, H. Yu, H. Liu, and G. Diamos. 2018. "Resource-efficient Neural Architect". *arXiv preprint arXiv:1806.07912*.

Žilinskas, A., and J. Calvin. 2019. "Bi-objective Decision Making in Global Optimization based on Statistical Models". *Journal of Global Optimization* 74(4):599–609.

AUTHOR BIOGRAPHIES

ANTONIO CANDELIERI, Ph.D., is Assistant Professor for the Department of Economics, Management and Statistics at the University of Milano-Bicocca, Italy. His research activities are focused on Machine Learning and Bayesian Optimization. His email address is antonio.candelieri@unimib.it. His website is https://www.unimib.it/antonio-candelieri.