Vilnius university
Institute of Data Science and
Digital Technologies
LITHUANIA

INFORMATICS (N009)

BAYESIAN GLOBAL OPTIMIZATION OF BLACK-BOX FUNCTIONS

Saulius Tautvaišas

October 2020

Technical Report DMSTI-DS-N009-20-04

VU Institute of Data Science and Digital Technologies, Akademijos str. 4, Vilnius
LT-08412, Lithuania
www.mii.lt

Abstract

The main goal of global optimization is to find the best solution of optimization problems which could have many other local optima solutions. It is common that objective functions in global optimization problems are expensive to evaluate, non-convex and has no derivative. These functions are usually called black-box functions. Bayesian optimization has recently emerged as a popular approach for optimizing expensive black-box functions.

This work describes how Bayesian optimization works, the key components, including Gaussian process regression and common acquisition functions. We present main limitations of Bayesian limitations and extensions to high-dimensional global optimization problems. Furthermore, we review some new ideas emerging from meta-learning applied to Bayesian optimization. Finally, we present some other methods used in global optimization.

Keywords: Bayesian optimization, Global optimization, Gaussian processes, Meta-learning

Contents

1 Introduction 4
2 Global Optimization Problem 4
3 Bayesian Global Optimization 5
3.1 Gaussian Process 6
3.1.1 GP kernels 6
3.1.2 Learning kernel parameters 7
3.2 Acquisition functions 8
3.2.1 Expected Improvement 8
3.2.2 GP-UCP 9
3.2.3 Thompson Sampling 9
3.2.4 The information-based acquisition functions 9
3.2.4.1 Entropy search 10
3.2.4.2 Predictive Entropy Search 10
3.3 Limitations of Bayesian Optimization 10
3.4 Extensions in Bayesian Optimization 11
3.4.1 High dimensional Bayesian optimization 11
3.4.2 Meta learning 12
3.4.2.1 Neural Processes 12
3.4.2.2 Meta-acquisition function 13
4 Related Global Optimization Models 13
4.1 Classical Global Optimization Models 13
4.1.1 Nelder–Mead algorithm 13
4.1.2 Genetic Algorithm 14
4.1.3 Ant Colony Optimization Algorithm 14
4.1.4 Simulated Annealing Algorithm 14
4.2 Lipschitz Global Optimization Models 15
4.2.1 Direct Method 15
4.2.1.1 Partitioning Hyper-rectangles 15
4.2.1.2 Potentially Optimal Hyper-rectangles 16
4.2.2 Simplicial Lipschitz optimization 17
5 Conclusions and Future work 18
6 References 19

1 Introduction

Global optimization is concerned with the computation and characterization of global minima (or maxima) of nonlinear functions. Global optimization problems are widespread in the mathematical modelling of real-world systems for a very broad range of applications (Horst & Pardalos, 2013).

Recently Bayesian optimization (BO) has become extremely popular for tuning hyperparameters in machine learning algorithms (Frazier, 2018), designing engineering systems (Forrester et al., 2008; Candelieri et al., 2018), reinforcement learning (Shahriari et al., 2016), automatic configuration (Thornton et al., 2013) or in chemical engineering when selecting candidates for high-throughput screening (Hernández-Lobato et al., 2017; Griffiths & Hernández-Lobato, 2020).

Despite the above-mentioned successes, optimization in high-dimension problems with large among of data is still challenging. There has been a series of work addressing these issues for high-dimensional problems which showed very promising results (Kandasamy et al., 2015; Ziyu Wang et al., 2016; Gardner et al., 2017; Mutný & Krause, 2018; Rolland et al., 2018; Zi Wang et al., 2018; Munteanu et al., 2019; Binois et al., 2020). Furthermore, new emerging ideas from meta learning have been applied to Bayesian optimization which allows to transfer knowledge from one task to another and can also help to overcome some limitations in Bayesian Optimization.

2 Global Optimization Problem

Global optimization problem could be described as finding global maximizer (minimizer) of an unknown continuous function $f : \mathcal{X} \to \mathbb{R}$ defined on a compact subset $\mathcal{X} \subseteq \mathbb{R}^D$. We consider the following global optimization problem:

$$x^* = \operatorname*{argmax}_{x \in \mathcal{X}} f(x),$$

The function f is an objective function and $\mathcal{X}$ is called the feasible set. Alternatively, $\mathcal{X}$ is called search space or domain (Pintér, 1996). The objective function f is called black-box if it satisfies one or more of the following criteria: it does not have a closed-form expression, is expensive to evaluate and does not have easily available derivatives.

We can only get black-box function f value points by querying its function values at arbitrary $x \in \mathcal{X}$.

3 Bayesian Global Optimization

Bayesian optimization (Mockus et al., 1978; Brochu et al., 2010; Frazier, 2018) is a methodology for performing global optimization of black-box functions that are noisy and expensive to evaluate. Given small number of observed objective function inputs and corresponding outputs, Bayesian optimization iteratively develops a global statistical model of the objective function, which could provide an estimate of uncertainty about objective function and can be used to balance trade-off between exploration and exploitation. The statistical model consists of a prior distribution that captures our assumptions about the behaviour of unknown objective function and data generation mechanism (Shahriari et al., 2016). During each optimization iteration a posterior distribution is computed by conditioning on the previous evaluations of the objective function. This model is also called probabilistic surrogate model because it approximates the original objective function and can be queried efficiently at lower computational cost.

To select the next query point, Bayesian optimization uses an acquisition function α, which measures how promising are each point in the search space $\mathcal{X}$ if it were to be evaluated next, based on assumptions about the objective function in our statistical model $\mathcal{M}$. The main goal is to find the next query point $\hat{x}$ which maximizes the acquisition function and use it for objective function f evaluation. The main steps of Bayesian optimization algorithm are illustrated in Algorithm 1.

Algorithm 1 Bayesian optimization

```
**Inputs**: objective $f$, acquisition function $\alpha$, search space $\mathcal{X}$, model $\mathcal{M}$, initial design $\mathcal{D}$
**repeat:**
    Fit the model $\mathcal{M}$ to the data $\mathcal{D}$
    Maximize the acquisition function: $\hat{x} = arg\ max_{x \in \mathcal{X}}\ \alpha(x, \mathcal{M})$
    Evaluate the function: $\hat{y} = f(\hat{x})$
    Add the new data to the data set: $\mathcal{D} = \mathcal{D} \cup \{(\hat{x}, \hat{y})\}$
**until** termination condition is met
**Output**: the recommendation $x^* = \arg max_{x \in \mathcal{X}}\ \mathbb{E}_{\mathcal{M}}[f(x)]$
```

After we select the most promising next query point $\hat{x}$, we evaluate the objective function f, and we add the new observation to the data set $\mathcal{D}$ and then begin the next iteration. Optimization loop is terminated when a maximum elapsed time for the entire optimization procedure or a maximum number of function evaluations is reached. Bayesian optimization is well suited for the problems when we can optimize acquisition function much more efficiently and easier than the original optimization problem.

3.1 Gaussian Process

Gaussian processes (GPs) are the most popular priors distribution used for modelling the function f in Bayesian optimization (Snoek et al., 2012; Shahriari et al., 2016; Frazier, 2018). They define distributions over functions where any finite set of function values has a multivariate Gaussian distribution (Rasmussen & Williams, 2018). A Gaussian process $GP(\mu, \kappa)$ is fully specified by a mean function $\mu(\cdot)$ and covariance (kernel) function $\kappa(\cdot,\ \cdot)$. Let f be a function sampled from $GP(0, \kappa)$. Given observations $\mathcal{D}_n = \{(x_t,\ y_t)\}_{t=1}^{n}$ where $y_t \sim \mathcal{N}(f(x_t),\ \sigma)$, we obtain the posterior mean and variance of the function as

$$\mu_n(x) = \kappa_n(x)^T(\mathrm{K}_n + \sigma^2\mathrm{I})^{-1}y_n$$

$$\sigma_n^2(x) = \kappa(x, x) - \kappa_n(x)^T(\mathrm{K}_n + \sigma^2\mathrm{I})^{-1}\kappa(x)$$

via the kernel matrix $\mathrm{K}_n = \left[\kappa(x_i,\ x_j)\right]_{x_i, x_j \in D_n}$ and $\kappa_n(x) = [\kappa(x_i,\ x)]_{x_i \in \mathcal{D}_n)}$.

Despite the fact that GP provide flexible, broadly applicable function estimators, the $O(n^3)$ computation of the inverse $(\mathrm{K}_n + \sigma^2\mathrm{I})^{-1}$ can is a major bottleneck as n grows for posterior function value predictions.

At each iteration of Bayesian optimization, one has to re-compute the predictive mean and variance. These two quantities are used to determine the next iteration x_{n+1} based on the belief about f given $\mathcal{D}_n$, a sampling strategy is defined in terms of an acquisition function.

3.1.1 GP kernels

Kernel function is the critical ingredient in defining Gaussian process prior. It encodes our assumptions about the objective function we wish to learn. The kernel function is a positive definite function and defines nearness or similarity between two input pairs x and x'. Kernel functions can be classified into stationary, dot-product or non-stationary functions. Stationary kernel functions are the most popular and widely used, so we restrict ourselves only to this family of functions. A stationary kernel is one whose value depends on x and x' only through their difference $x - x'$, which means that the structure of the function is translation invariant (Rasmussen & Williams, 2018).

The two most widely used kernel function are the squared exponential kernel and the Matérn 5/2 kernel. The squared exponential is infinitely differentiable, which means that the GP with this covariance function has mean square derivatives of all orders and is very smooth. It has the form

$$\kappa_{SE}(x, x') = \exp(-\frac{|x - x'|^2}{2\ell^2})$$

where ℓ defining the characteristic length-scale.

Squared exponential kernel puts a strong smoothness assumption on objective function, which is unrealistic for modelling many physical processes. For this reason, it is recommended to use Matérn 5/2 kernel function which is only two times mean square differentiable. It has the following form:

$$\kappa_{M52}(x, x') = \left(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\right) exp\left(-\frac{\sqrt{5}r}{\ell}\right)$$

where $r = |x - x'|$ and ℓ is the characteristic length-scale.

3.1.2 Learning kernel parameters

In many practical applications, it may not be easy to specify all aspects of the kernel function. While some properties such as stationarity of the covariance function may be easy to determine from the context, it is more difficult to have an information about other properties, such as the value of free hyperparameters. The mismatch of hyperparameters and the data can lead to very poor performance.

For the squared exponential kernel function hyperparameters play the role of characteristic length-scales which defines how far length-scale needs to move along a particular axis in input space for the function values to become uncorrelated. If the length scales in kernel function are set very large, then the GP prior will not be able to capture the higher variations in the objective function and if the length scales are set very small, the GP might fail to generalize. The kernel hyperparameters can be learned from the data by maximizing the marginal likelihood of the GP. The marginal likelihood of a GP is given by:

$$\log p(y|X, \theta) = -\frac{1}{2} y^T K_y^{-1} y - \frac{1}{2} log|K_y| - \frac{n}{2} log 2\pi$$

Where $K_y = \mathrm{K}_f + \sigma_n^2 \mathrm{I}$ is the covariance matrix for the noisy targets y and K_f is the covariance matrix for the noise-free latent f. We can explicitly write the marginal likelihood conditioned on the hyperparameters and then perform maximum likelihood estimation respect to the hyperparameters.

The complexity of computing the marginal likelihood is dominated by the need to invert the K_y^{-1} matrix, which requires $O(n^3)$ time for inversion of an n by

n matrix. Once inverted mattrix is known, the computation of the derivatives with respect to hyperparameters requires only time $O(n^2)$ per hyperparameter (Rasmussen & Williams, 2018).

3.2 Acquisition functions

Acquisition function is used to guide the search for finding the maximum of objective function. Usually, high acquisition function values correspond to potentially high values of the objective function. Maximizing the acquisition function is used to select the next point at which to evaluate the function. The choice of acquisition function is nontrivial. Each works well for certain classes of functions, and it is often difficult or impossible to know which will perform best on an unknown function (Brochu et al., 2010).

3.2.1 Expected Improvement

The expected improvement (EI) acquisition function (Mockus et al., 1978; Donald R. Jones et al., 1998) is one of the most popular acquisition function. It measures the expected improvement amount by which observing $f_{n+1}(x)$ leads to improvement over some target $f(x^+)$:

$$\alpha_{EI}(x|\, \mathcal{D}_n) = \mathbb{E}(max\{0, f_{n+1}(x) - f(x^+)\}|\mathcal{D}_n)$$

In this definition $x^+ = \operatorname{argmax}_{x \in \{x_{1:n}\}} f(x)$ is the element with the best objective value in the n steps of the optimization process. The next query point is found by:

$$x_{n+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \alpha_{EI}(x|\mathcal{D}_n)$$

This utility function is biased to selecting the points with high variance and points with high mean value (Donald R. Jones et al., 1998).

3.2.2 GP-UCP

The Gaussian Process Upper Confidence Bound (GP-UCB) acquisition function (Srinivas et al., 2010) is defined as follow:

$$\alpha_{UCB}(x) = \mu(x) + \kappa\sigma(x)$$

where κ is a constant, μ and σ are the posterior predictive marginal GP mean and variance. GP-UCB acquisition function implicitly balance the exploration-exploitation trade-off and prefers the points with high posterior mean and variance.

3.2.3 Thompson Sampling

Thompson sampling (TS) (Thompson, 1933) is also commonly known as randomized probability matching is a randomized acquisition strategy which was introduced in 1933 and recently attracted renewed interest in multi-armed bandits problems.

In the bandit setting this strategy samples a reward function from the posterior and selects the arm with the highest simulated reward, while in the GP context this strategy corresponds to sampling the objective function from the GP posterior and then finding the maximum of that sample. TS can be formulated as acquisition function as:

$$\alpha_{TS}(x, \mathcal{D}_n) = f^{(n)}(x)$$
$$f^{(n)}(x) \sim GP(\mu, \kappa | \mathcal{D}_n)$$

Empirical evaluations show good performance which, however, seems to deteriorate in high dimensional problems, likely due to aggressive exploration (Shahriari et al., 2016).

3.2.4 The information-based acquisition functions

The information-based acquisition functions seek to maximize the expected information gain about the solution to the global optimization problem. This is achieved by considering the posterior distribution over the location of the solution given the data $p_*(x| D_n)$. Two most popular information-based acquisition functions are entropy search (ES) and predictive entropy search (PES).

3.2.4.1 Entropy search

The goal of Entropy Search (ES) acquisition function is to reduce the uncertainty in the location x^* by selecting the point which is expected to cause the largest reduction in entropy of the distribution $p_*(x|\,\mathcal{D}_n)$ (Hennig & Schuler, 2012). The acquisition function for ES can be expressed formally as:

$$\alpha_{ES}(x) = H(x^*|\,\mathcal{D}_n) - \mathbb{E}_{y\,|\,\mathcal{D}_n,x}\, H(x^*|\,\mathcal{D}_n \cup \{(x,y)\})$$

where $H(x^*|\,\mathcal{D}_n)$ denotes the differential entropy of the posterior distribution $p_*(x|\,\mathcal{D}_n)$ and the expectation is over the distribution of the random variable y. This function is not tractable for continuous search spaces and so approximations must be made. Recent work uses a discretization of the search space to obtain a smooth approximation $p_*(x|\,\mathcal{D}_n)$ and its expected information gain (Shahriari et al., 2016).

3.2.4.2 Predictive Entropy Search

Predictive Entropy Search (PES) acquisition function strategy is to select the next point from the search space which maximizes the expected reduction in the negative differential entropy of $p_*(x|\,\mathcal{D}_n)$ (Hernández-Lobato et al., 2014).

$$\alpha_{PES}(x) = H[p(x^*|\,\mathcal{D}_n)] - \mathbb{E}_{y\,|\,\mathcal{D}_n,x}\,[H[p(x^*|\,\mathcal{D}_n \cup \{(x,y)\})]]$$

where $H[p(x)] = -\int p(x)\log p(x)\,dx$ represents the differential entropy of its argument and the expectation above is taken with respect to the posterior predictive distribution of y given x. The exact evaluation of this equation is not feasible in practice. However after making few simplifying assumptions the expectation can be approximated via Monte Carlo with Thompson samples (Shahriari et al., 2016; Frazier, 2018).

3.3 Limitations of Bayesian Optimization

There are several known limitations and challenges in Bayesian optimization. Even though Bayesian optimization typically works well in low-dimensional search spaces, optimization in high-dimension problems become very challenging, because complexity grow exponentially with the dimensionality of the search space. Another challenging area is computational complexity of Gaussian process, which is cubic in the number of data points and to ensure that a global optimum is found when dimensionality increases more data points are needed to have good coverage of search space which becomes a bottleneck. Furthermore, maximizing acquisition function when search space dimensionality is high can become challenging for commonly used global optimisation heuristics, because it could require computation exponential in dimension. Finally, available kernels are usually restricted in their functional form and an additional optimisation procedure is required to identify the most suitable kernel, as well as its hyperparameters, for any given task (Frazier, 2018; Rasmussen & Williams, 2018; Kim et al., 2019).

3.4 Extensions in Bayesian Optimization

3.4.1 High dimensional Bayesian optimization

There has been a series of work addressing BO in high-dimensional (Kandasamy et al., 2015; Ziyu Wang et al., 2016; Gardner et al., 2017; Mutný & Krause, 2018; Rolland et al., 2018; Zi Wang et al., 2018; Munteanu et al., 2019; Binois et al., 2020). Some of these authors tries to find and exploit potential additive structure in the objective function (Kandasamy et al., 2015; Gardner et al., 2017; Zi Wang et al., 2018). These methods typically rely on training a large number of GP and therefore do not scale to large evaluation budgets. Other methods exist that rely on a mapping between the high-dimensional space and an unknown low-dimensional subspace to scale to large numbers of observations (Ziyu Wang et al., 2016; Munteanu et al., 2019). The BOCK algorithm of Oh et al. (Oh et al., 2018) uses a cylindrical transformation of the search space to achieve scalability to high dimensions. Ensemble Bayesian optimization (EBO) (Zi Wang et al., 2018) uses an ensemble of additive GPs together with a batch acquisition function to scale BO to tens of thousands of observations and high-dimensional spaces. (Eriksson et al., 2019) abandoned a global surrogate and instead maintained several local models that move towards better solutions. Their TurBO algorithm applies a bandit approach to allocate samples efficiently between these local searches. Recently, (Munteanu et al., 2019) have proposed the general HeSBO framework that extends GP-based BO algorithms to high-dimensional problems using a novel subspace embedding that overcomes the limitations of the Gaussian projections used in (Ziyu Wang et al., 2016; Binois et al., 2020).

3.4.2 Meta learning

Meta-learning is an important and active field of research and recently started attracting more and more attention in Bayesian optimization. These methods try to achieve optimal data-efficiency by transferring knowledge across many different tasks. This is usually done by incorporating the information obtained from previously seen tasks into the optimization process. Many practical applications in optimizations are repeated numerous times in similar settings, so using information from previous runs could allow to achieve global optimal in fewer steps.

3.4.2.1 Neural Processes

Bayesian Optimization based on Gaussian process regression scales cubically with the respect the number of evaluations. Many available kernels for GP requires strong prior assumption about the objective function which limits their functional form. Also, finding the most suitable kernel for the given problem and optimizing its the hyperparameters is not an easy task. Neural Processes (NPs) (Gamelo et al., 2018; Kim et al., 2019) was introduced to overcome these limitations by the standard GP regression. NP is a neural network-based formulation that learns an approximation of a stochastic process by modelling a distribution over regression functions with prediction complexity linear in the size of observed context set. Furthermore, the model overcomes many functional design restrictions by learning an implicit kernel from the data directly.

NP model maps an input $\boldsymbol{x}_i \in \mathbb{R}^{d_x}$ to an output $\boldsymbol{y}_i \in \mathbb{R}^{d_y}$ and defines a conditional distribution of target $(\boldsymbol{x}_c, \boldsymbol{y}_c) = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i \in C}$ given an arbitrary number of observed contexts $(\boldsymbol{x}_T, \boldsymbol{y}_T) = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i \in T}$ in a way that is invariant to ordering of the contexts and ordering of the targets. This distribution could be modeled as:

$$p(\boldsymbol{y}_T | \boldsymbol{x}_T, \boldsymbol{x}_C, \boldsymbol{y}_C) = \int p(\boldsymbol{y}_T | \boldsymbol{x}_T, \boldsymbol{r}_C, \boldsymbol{z}) \; q(\boldsymbol{z} | s_C) \; d\boldsymbol{z}$$

With $r_c = r(x_C, y_C)$ where r is a deterministic function that aggregates $(\boldsymbol{x}_c, \boldsymbol{y}_c)$ into a finite dimensional representation with permutation invariance in C. A global latent variable $\boldsymbol{z}$ represents uncertainty in the predictions of $\boldsymbol{y}_T$ for a given observed $(\boldsymbol{x}_c, \boldsymbol{y}_c)$. It is modelled by factorized Gaussian and parameterized by $s_c = s(\boldsymbol{x}_C, \boldsymbol{y}_C)$.

The likelihood $p(\boldsymbol{y}_T | \boldsymbol{x}_T, \boldsymbol{r}_C, \boldsymbol{z})$ is called *decoder* and q, r, s forms an *encoder*. The parameters of the encoder and decoder are learned by maximizing the following ELBO.

Neural processes were applied to Bayesian optimization on 1-D function using Thompson sampling. The results showed that this approach does not reach the optimal performance compared to standard GP approach as NP samples were noisier than those of a GP. However, NP were much faster to evaluate since only a forward pass through the network is needed. This difference in computational speed gets even more noticeable as the dimensionality of the problem and the number of necessary function evaluations increases (Gamelo et al., 2018).

3.4.2.2 Meta-acquisition function

Neural Acquisition Function (NAF) (Volpp et al., 2019) is a flexible meta-learning approach which allows to directly to incorporate the prior knowledge from previous runs and other related tasks into the optimization strategy of Bayesian optimization. The model replaces traditional acquisition function with neural network while retaining all other elements from standard Bayesian Optimization framework. Using reinforcement learning to meta-train an acquisition function the proposed method learns to extract implicit structural information and to exploit it for improved data-efficiency.

The experiments showed that this method was able to outperform the existing methods and was broadly applicable to a wide range of practical problems were source data was abundant or scares. The resulting neural AFs can represent search strategies which go far beyond the abilities of current approaches.

4 Related Global Optimization Models

4.1 Classical Global Optimization Models

4.1.1 Nelder–Mead algorithm

Nelder-Mead (NM) method is one of the best known optimization algorithms originally designed for solving convex non-differentiable unconstrained nonlinear optimization problems (Nelder & Mead, 1965). It belongs to direct search family methods which uses only the objective function values for searching the optimal solution.

There are many modifications of the NM method proposed in the literature in since the method was first developed (Kolda et al., 2003). In recent years NM is used for solving non-convex and non-differentiable optimization problems (Dražić et al., 2016). Also, many other algorithms were developed based on hybridization this algorithm with other algorithms (Chelouah & Siarry, 2003) or based on similar ideas (Eriksson et al., 2019).

4.1.2 Genetic Algorithm

Genetic algorithms (GA) were first popularized by (Holland, 1975) are stochastic optimization algorithms inspired by the principles of natural evolution. They can often outperform conventional optimization methods when applied to difficult real-word optimization problems. Many different evolutionary algorithm based strategies have been developed recently to find the global minimum for nonlinear programming problems (Pham & Yang, 1993; Andrzej & Stanislaw, 2006; Toledo et al., 2014)

4.1.3 Ant Colony Optimization Algorithm

Ant colony optimization is a classical approach to solve combinatorial optimization problems, which were introduced by (Dorigo, 1992). The main idea of this algorithm is the indirect communication among the individuals of a colony of agents, called ants. The method is based on the principle how ants search for food and find their way back to the nest.

This algorithm was successfully applied to global optimization problem (Toksari, 2006) and when compared with other global optimization algorithms showed noticeable performance improvement.

4.1.4 Simulated Annealing Algorithm

Simulated annealing (SA) is a stochastic method for global optimization. This algorithm originated from the analogy between the physical annealing process and the problem of finding minimal solutions for discrete minimization problems. This algorithm was first developed for combinatorial minimization problems (Kirkpatrick et al., 1983) and later modified to apply for continuous global optimization functions

(Dekkers & Aarts, 1991). SA algorithms have been successfully applied to a wide variety of real engineering problems (Henderson et al., 2006).

4.2 Lipschitz Global Optimization Models

4.2.1 Direct Method

The DIRECT optimization algorithm (D. R. Jones et al., 1993; Deng & Ferris, 2007; Donald R. Jones, 2008) which stands for *di*viding *rect*angles, is a global optimization method first motivated by Lipschitz optimization, which has proven to be effective in a wide range of application domains. The algorithm works by iteratively dividing large hyperrectangles in search domain into the smaller ones. Each hyperrectangle in the decomposition is characterized by the objective function value at the center location. During each iteration, a set of potentially optimal hyperrectangles are selected for further divisions. An example of such partition in 2 dimensional space is illustrated in Figure 1.

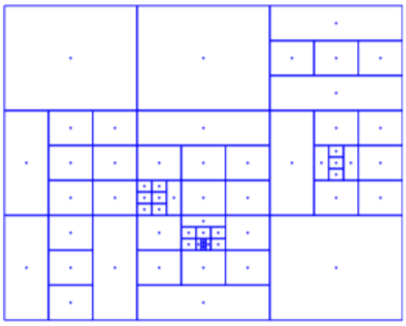

Figure 1: The DIRECT optimization algorithm

4.2.1.1 Partitioning Hyper-rectangles

For each hyperrectangle, let $\mathcal{D}$ be the coordinate directions corresponding to the largest side lengths, δ be one third of the largest length, and c be the center point. The function will explore the objective values at the points $c \pm \delta e_i$, for all $e_i \in \mathcal{D}$, where e_i is the ith unit vector. The hyperrectangle will be trisected along the dimensions in $\mathcal{D}$, first along dimensions whose objective values are better. The procedure continues until each point $c \pm \delta e_i$ occupies a single hyperrectangle. Two possible partitioning scheme in a 2-dimensional case are illustrated in Figure 2.

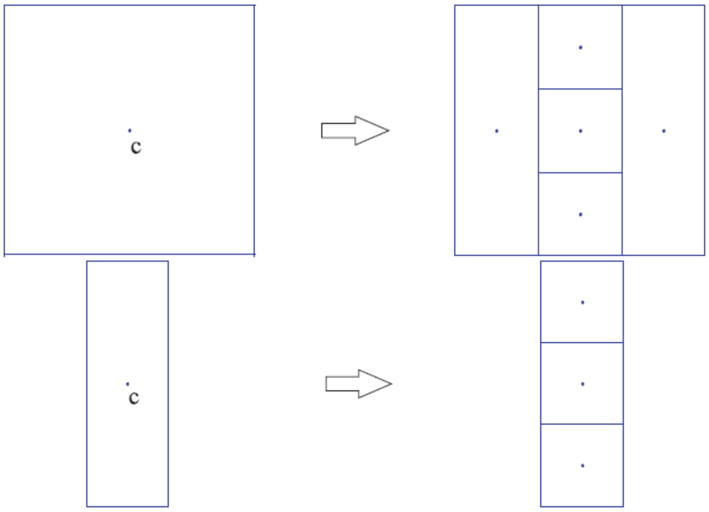

Figure 2: Partitioning hyperrectangles

Since we only perform trisections, the length of any side in the unit hyperrectangle can possibly be $3^{-k}, k = 1, 2, \ldots$

4.2.1.2 Potentially Optimal Hyper-rectangles

Selection of potentially optimal hyperrectangles combines the purposes of both global and local searches. Let $\mathcal{H}$ be the index set of existing hyperrectangles. For each hyperrectangle $j \in \mathcal{H}$, we evaluate the function value at the center representing point $f(c_j)$ and note the size of the hyperrectangle α_j. The size α_j is computed as the distance from the center point to the corner point. A hyperrectangle $j \in \mathcal{H}$ is said to be potentially optimal if there exists a constant $\widetilde{K}$ such that:

$$f(c_j) - \widetilde{K}\alpha_j \le f(c_i) - \widetilde{K}\alpha_i, \forall i \in \mathcal{H},$$

$$f(c_j) - \widetilde{K}\alpha_j \le f_{min} - \varepsilon|f_{min}|.$$

In the above expressions, f_{min} is the lowest function value available and ε is a parameter that balances between global and local search. The parameter is typically nonsensitive and set as 0.0001. An equivalent interpretation of the process of selecting potential optimal rectangles is illustrated in Figure 3.

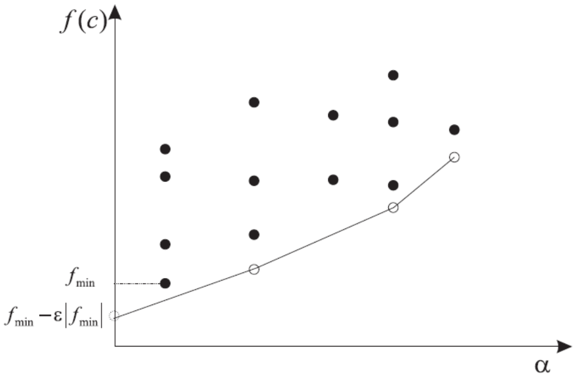

Figure 3: Identifying potentially optimal hyperrectangles

First hyperrectangles are sorted in groups according to the size α. Each hyperrectangle is plotted in the figure as a black dot in accordance with its center function value $f(c_j)$ and size α_j. Potentially optimal hyperrectangels are denoted as white dots on the lower convex hull of the grap in the figure above.

The introduction of ε may result in exclusions of good hyperrectangles in the smaller size groups. Thus ε is considered as a balancing parameter between local and global search. As noted from the figure, the best hyperrectangle in the largest size group is always selected. The algorithm will eventually converge to the global optimum because the maximum size $max_j \alpha_j$ decreases to zero and the entire search space is thoroughly explored.

4.2.2 Simplicial Lipschitz optimization

Many research papers related to DIRECT algorithm use hyper-rectangles for search space partitioning. Using different strategies for search space partitioning could be more beneficial for some different type of problems. DISIMPL optimization algorithm (Paulavičius & Žilinskas, 2014a, 2014b) which means DIviding SIMPLices was proposed to use simplicial partitions for search space partitioning. This algorithm adopts similar ideas of DIRECT algorithm with comparable convergence properties.

The experiments showed that proposed DISIMPL algorithm had very competitive results to DIRECT algorithm for standard test problems and performed particularly well when the objective functions have symmetries and the numbers of local and global extremum points can be reduced by avoiding symmetries.

5 Conclusions and Future work

We have introduced Bayesian optimization and its main components, including Gaussian process regression, acquisition functions and the process of learning hyperparameters. We then discussed main challenges and limitations facing Bayesian optimization.

Many new research directions are focused on addressing the main limitation in Bayesian optimization. One of which is developing new methods that work well with high-dimensional optimization problems. We have reviewed the main literature and experiments in this research are, which showed very promising results. Also, new ideas coming from meta-learning field were successfully applied to Bayesian optimization which address scalability issues.

Finally, we have discussed other popular methods used in global optimization from classical to more recent ones. Many new algorithms use some ideas from these methods to develop hybrid and more efficient new optimization algorithms.

For future work, research in meta-learning and knowledge transfer domain, particularly in neural processes, seems to be very promising for high-dimensional problems. This approach could be used to replace Gaussian process which is used in most work on Bayesian optimization and could speed up optimization process. Furthermore, developing new acquisition functions and combining with meta-learning may provide substantial value in high dimensional problems.

6 References

Andrzej, O., & Stanislaw, K. (2006). Evolutionary Algorithms for Global Optimization. In J. D. Pintér (Ed.), *Global Optimization: Scientific and Engineering Case Studies* (pp. 267–300). Springer US. https://doi.org/10.1007/0-387-30927-6_12

Binois, M., Ginsbourger, D., & Roustant, O. (2020). On the choice of the low-dimensional domain for global optimization via random embeddings. *Journal of Global Optimization*. https://doi.org/10.1007/s10898-019-00839-1

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *ArXiv Preprint ArXiv:1012.2599*.

Candelieri, A., Perego, R., & Archetti, F. (2018). Bayesian optimization of pump operations in water distribution systems. *Journal of Global Optimization*. https://doi.org/10.1007/s10898-018-0641-2

Chelouah, R., & Siarry, P. (2003). Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. *European Journal of Operational Research*. https://doi.org/10.1016/S0377-2217(02)00401-0

Dekkers, A., & Aarts, E. (1991). Global optimization and simulated annealing. *Mathematical Programming*. https://doi.org/10.1007/BF01594945

Deng, G., & Ferris, M. C. (2007). Extension of the DIRECT optimization algorithm for noisy functions. *Proceedings - Winter Simulation Conference*. https://doi.org/10.1109/WSC.2007.4419640

Dorigo, M. (1992). Optimization, learning and natural algorithms. *PhD Thesis, Politecnico Di Milano*.

Dražić, M., Dražić, Z., Mladenović, N., Urošević, D., & Zhao, Q. H. (2016). Continuous variable neighbourhood search with modified Nelder-Mead for non-differentiable optimization. *IMA Journal of Management Mathematics*. https://doi.org/10.1093/imaman/dpu012

Eriksson, D., Pearce, M., Gardner, J. R., Turner, R., & Poloczek, M. (2019). Scalable Global Optimization via Local Bayesian Optimization. *Advances in Neural Information Processing Systems*, 5497–5508. http://arxiv.org/abs/1910.01739

Forrester, A. I. J., Sóbester, A., & Keane, A. J. (2008). Engineering Design via Surrogate Modelling. In *Engineering Design via Surrogate Modelling*. https://doi.org/10.1002/9780470770801

Frazier, P. I. (2018). Bayesian Optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*. https://doi.org/10.1287/educ.2018.0188

Gamelo, M., Rosenbaum, D., Maddison, C. J., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D. J., & Eslami, S. M. A. (2018). Conditional neural processes. *35th International Conference on Machine Learning, ICML 2018*.

Gardner, J. R., Guo, C., Weinberger, K. Q., Garnett, R., & Grosse, R. (2017). Discovering and exploiting additive structure for Bayesian optimization. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017*.

Griffiths, R. R., & Hernández-Lobato, J. M. (2020). Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*. https://doi.org/10.1039/c9sc04026a

Henderson, D., Jacobson, S. H., & Johnson, A. W. (2006). The Theory and Practice of Simulated Annealing. In *Handbook of Metaheuristics*. https://doi.org/10.1007/0-

306-48056-5_10

Hennig, P., & Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*.

Hernández-Lobato, J. M., Hoffman, M. W., & Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. *Advances in Neural Information Processing Systems*.

Hernández-Lobato, J. M., Requeima, J., Pyzer-Knapp, E. O., & Aspuru-Guzik, A. (2017). Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. *34th International Conference on Machine Learning, ICML 2017*.

Holland, J. H. (1975). Adaptation in natural and artificial systems, University of Michigan press. *Ann Arbor, MI*.

Horst, R., & Pardalos, P. M. (2013). *Handbook of global optimization* (Vol. 2). Springer Science & Business Media.

Jones, D. R., Perttunen, C. D., & Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*. https://doi.org/10.1007/BF00941892

Jones, Donald R. (2008). Direct Global Optimization Algorithm. In *Encyclopedia of Optimization*. https://doi.org/10.1007/978-0-387-74759-0_128

Jones, Donald R., Schonlau, M., & Welch, W. J. (1998). Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*. https://doi.org/10.1023/A:1008306431147

Kandasamy, K., Schneider, J., & Póczos, B. (2015). High dimensional Bayesian Optimisation and bandits via additive models. *32nd International Conference on Machine Learning, ICML 2015*.

Kim, H., Mnih, A., Schwarz, J., Garnelo, M., Eslami, A., Rosenbaum, D., Vinyals, O., & Teh, Y. W. (2019). Attentive neural processes. *7th International Conference on Learning Representations, ICLR 2019*.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*. https://doi.org/10.1126/science.220.4598.671

Kolda, T. G., Lewis, R. M., & Torczon, V. (2003). Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*. https://doi.org/10.1137/S003614450242889

Mockus, J., Tiesis, V., & Zilinskas, A. (1978). The application of Bayesian methods for seeking the extremum. In *Towards Global Optimisation*. https://doi.org/10.1007/978-94-009-0909-0_8

Munteanu, A., Nayebi, A., & Poloczek, M. (2019). A framework for Bayesian optimization in embedded subspaces. *36th International Conference on Machine Learning, ICML 2019*.

Mutný, M., & Krause, A. (2018). Efficient high dimensional Bayesian optimization with additivity and quadrature fourier features. *Advances in Neural Information Processing Systems*.

Nelder, J. A., & Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*. https://doi.org/10.1093/comjnl/7.4.308

Oh, C. Y., Gavves, E., & Welling, M. (2018). BOCK: Bayesian Optimization with Cylindrical Kernels. *35th International Conference on Machine Learning, ICML 2018*.

Paulavičius, R., & Žilinskas, J. (2014a). *Simplicial global optimization*. Springer.

Paulavičius, R., & Žilinskas, J. (2014b). Simplicial Lipschitz optimization without the Lipschitz constant. *Journal of Global Optimization*.

https://doi.org/10.1007/s10898-013-0089-3

Pham, D. T., & Yang, Y. (1993). Optimization of Multi-Modal Discrete Functions Using Genetic Algorithms. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *207*(1), 53–59. https://doi.org/10.1243/PIME_PROC_1993_207_159_02

Pintér, J. D. (1996). Global Optimization in Action. *Nonconvex Optimization and Its Applications*.

Rasmussen, C. E., & Williams, C. K. I. (2018). Gaussian Processes for Machine Learning. In *Gaussian Processes for Machine Learning*. https://doi.org/10.7551/mitpress/3206.001.0001

Rolland, P., Scarlett, J., Bogunovic, I., & Cevher, V. (2018). High-dimensional Bayesian optimization via additive models with overlapping groups. *International Conference on Artificial Intelligence and Statistics, AISTATS 2018*.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. In *Proceedings of the IEEE*. https://doi.org/10.1109/JPROC.2015.2494218

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*.

Srinivas, N., Krause, A., Kakade, S., & Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*. https://doi.org/10.1109/TIT.2011.2182033

Thompson, W. R. (1933). On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*. https://doi.org/10.2307/2332286

Thornton, C., Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2013). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2487575.2487629

Toksari, M. D. (2006). Ant colony optimization for finding the global minimum. *Applied Mathematics and Computation*. https://doi.org/10.1016/j.amc.2005.09.043

Toledo, C. F. M., Oliveira, L., & França, P. M. (2014). Global optimization using a genetic algorithm with hierarchically structured population. *Journal of Computational and Applied Mathematics*. https://doi.org/10.1016/j.cam.2013.11.008

Volpp, M., Fröhlich, L. P., Fischer, K., Doerr, A., Falkner, S., Hutter, F., & Daniel, C. (2019). Meta-Learning Acquisition Functions for Transfer Learning in Bayesian Optimization. *ArXiv Preprint ArXiv:1904.02642*.

Wang, Zi, Gehring, C., Kohli, P., & Jegelka, S. (2018). Batched large-scale bayesian optimization in high-dimensional spaces. *International Conference on Artificial Intelligence and Statistics, AISTATS 2018*.

Wang, Ziyu, Hutter, F., Zoghi, M., Matheson, D., & De Freitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*. https://doi.org/10.1613/jair.4806