July 2023

Bayesian optimization with additive kernels for the calibration of simulation models to perform cost-effectiveness analysis

David GÓMEZ-GUILLÉN [a,b,1], Mireia DÍAZ [b,c] Josep Lluís ARCOS [d] and Jesus CERQUIDES [d]

[a] *Universitat Autònoma de Barcelona (UAB)*
[b] *Institut Català d'Oncologia (ICO) - Institut d'Investigació Biomèdica de Bellvitge (IDIBELL)*
[c] *Consortium for Biomedical Research in Epidemiology and Public Health - CIBERESP. Carlos III Institute of Health*
[d] *Institut d'Investigació en Intel·ligència Artificial - Consell Superior d'Investigacions Científiques (IIIA-CSIC)*
ORCiD ID: David Gómez-Guillén https://orcid.org/0000-0003-1787-6482, Mireia Díaz https://orcid.org/0000-0001-9360-4548, Josep Lluís Arcos https://orcid.org/0000-0001-7751-1210, Jesus Cerquides https://orcid.org/0000-0002-3752-644X

Abstract. The use of mathematical simulation models of diseases in economic evaluation is an essential and common tool in medicine aimed at guiding decision-making in health. Cost-effectiveness analyses are a type of economic evaluation that assess the balance between long-term health benefits and the economic sustainability of different health interventions using this type of models. One critical aspect of these models is the accurate representation of the disease's natural history, which requires a set of parameters such as probabilities and disease burden rates. While these parameters can be obtained from scientific literature, they often need calibration to fit the model's expected outcomes. However, the calibration process can be computationally expensive and traditional optimization methods can be time-consuming due to relatively simple heuristics that may not even guarantee feasible solutions. In this work, we investigate the use of Bayesian optimization to enhance the calibration process by leveraging domain-specific knowledge and exploiting inherent structural properties in the solution space. Specifically, we examine the effect of additive kernel decomposition and constraint handling for efficient search. Our preliminary results show that this improved Bayesian optimization procedure asymptotically improves the calibration process, leading to faster convergence and better solutions for larger simulation models.

Keywords. bayesian optimization, gaussian processes, additive kernels, constrained optimization, simulation models, cost-effectiveness models, cancer research

[1] Corresponding Author: David Gómez-Guillén, dgomez_ext@iconcologia.net

1. Introduction

Healthcare interventions are increasingly evaluated based on their cost-effectiveness due to usual budgetary constraints, ensuring equitable and efficient distribution of healthcare services. These constraints mean that not all available and effective interventions can be included in health plans. In many countries, it has become standard policy to assess the costs of new healthcare interventions in relation to their expected benefits before implementing them. Cost-effectiveness analysis (CEA) using mathematical simulation models is a crucial tool in this context, enabling us to assess the value of healthcare interventions and determine which ones offer the best value for money in the long term[1]. By comparing the costs and benefits of alternative interventions, policymakers and healthcare providers can prioritize strategies and allocate resources to achieve the maximum health benefits for the population. Ultimately, the goal of healthcare is to improve health outcomes, and CEA plays a vital role in achieving this objective[2].

CEA usually relies on simulation models that mimic disease processes to project the effects of different medical strategies on health outcomes over time[3]. There are different types of models but some of the most common simulate the traversal of a group of individuals through different health states (figure 1). These models can generate various outcomes, but they always produce two critical measures: the average cost and the average life expectancy, usually measured in Quality-Adjusted Life Years (QALYs)[4].

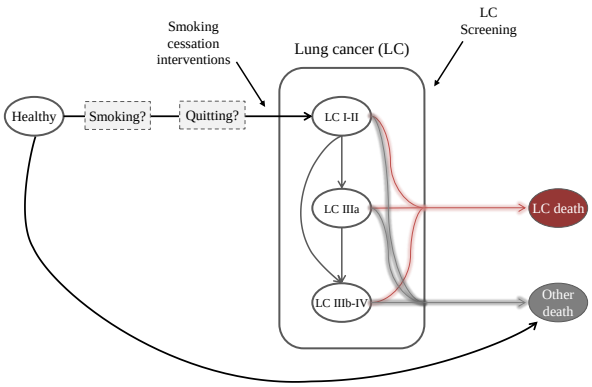

Figure 1. Lung cancer markov model state diagram

In order to execute the simulations, input parameters are required to describe the disease process, such as probabilities, hazard ratios, or disease burden rates extracted from scientific literature. Due to the inherent uncertainty of these values, it is often necessary to calibrate the model before proceeding with the analysis. Calibration consists in adjusting the input parameters until the resulting output approximates a target value identified in the scientific literature, such as disease incidence, prevalence, or mortality. This optimization process can be especially taxing for complex models, and may necessitate the use of advanced techniques to efficiently explore the solution space.

Moreover, calibrations can be highly dimensional optimization problems with many arbitrary constraints between parameters, dictated by the specific medical domain. In this work, we explore the challenges associated with calibrating simulation models and propose methods to overcome them. Our research provides valuable insight into novel ways

to calibrate these simulation models more efficiently using Bayesian Optimization (BO). We also investigate the circumstances under which it outperforms common methods used in the field.

2. Background

2.1. Bayesian Optimization

BO is a powerful technique for optimizing expensive, black-box functions that are difficult or time-consuming to evaluate[5]. The key advantage of this method is that it can find the global optimum of the function with relatively few evaluations, even in high-dimensional spaces. This is because the method actively seeks out the most promising areas of the input space to evaluate next, rather than simply evaluating points at random or using simple heuristics. The downside is that the method can be computationally expensive, especially for functions with a large number of input variables or when the surrogate model is complex.

BO has been successfully applied to a wide range of optimization problems in machine learning, including hyperparameter tuning, experimental design, and automatic algorithm configuration. It can help to quickly identify good values of the hyperparameters or experimental conditions, without having to exhaustively search the entire parameter space.

2.2. Gaussian Processes

Gaussian Processes (GPs) are non-parametric regression models that represent each observation as a random variable drawn from a normal distribution $f(x) \sim \mathcal{N}(\mu(x), k(x,x))$ [6]. The covariance function or kernel is the mechanism to give a GP its expressive power, and its choice will heavily depend on the kind of function we aim to model[7]. The squared exponential (SE) kernel $k(x,x') = \sigma^2 e^{-\frac{||x-x'||^2}{2l^2}}$ is a popular choice, despite significant drawbacks such as its locality and sensitivity to the curse of dimensionality[8]. In general, modeling complex high dimensional functions using a single kernel can be computationally expensive using local kernels, making this a first class research problem (e.g. [12][13]). Additive kernel decomposition[9] addresses this problem by breaking down the kernel into a sum of simpler kernels, each of which captures a different aspect of the relationship between the input variables. This approach can capture both local and global interactions between the input variables. Additionally, additive kernel decomposition can improve the interpretability of the model, as each kernel term can be associated with a specific interaction term. This can help users understand which orders of interaction are important for the current optimization problem.

Additive kernels suffer from the non-identifiability problem: the kernel hyperparameters are not uniquely identifiable from the observed data, which can lead to challenges in model selection and interpretation. To ensure a unique decomposition, Lu et al[11] proposed an extension of additive kernels by including an extra constant kernel $\tilde{k}_{add_0}(x,x')$ with an additional variance hyperparameter σ_0^2 and an orthogonality constraint to generate Orthogonal Additive Kernels (OAK)[11]. Assuming a normal input distribution $x_i \sim \mathcal{N}(\mu_i, \delta_i^2)$, the following constrained base kernel is derived:

$$
\begin{aligned}
k_{add_{OAK}}(x,x') &= \sum_{i=0}^{D} \sigma_i^2 \tilde{k}_{add_i}(x_i,x_i') \\
\tilde{k}_{add_j}(x,x') &= \sum_{1\le i_1<i_2<\ldots<i_j\le D} \left[\prod_{d=1}^{j} \tilde{k}_{i_d}(x_{i_d},x_{i_d}')\right] \\
\tilde{k}_i(x,x') &= e^{\frac{(x_i-x_i')^2}{2l_i^2}} - \frac{l_i\sqrt{l_i^2+2\delta_i^2}}{l_i^2+\delta_i^2} e^{-\frac{(x_i-\mu_i)^2+(x_i'-\mu_i)^2}{2(l_i^2+\delta_i^2)}}
\end{aligned}
\tag{1}
$$

One important advantage of these additive kernels is that we can interpret the σ_i^2 as the contribution of each individual order to the total kernel. Since many problems often rely on a few low-order interactions, we can truncate the higher orders and limit the computational cost while retaining most of the information present in the full decomposition. To achieve this, OAK kernels can be useful in accurately identifying each contribution and providing an accurate representation on the actual composition on the function.

3. Methodology

In this section we will describe the simulation model and the optimization methods we will use to calibrate it. These include BO and the rest of techniques that will be compared.

3.1. Description of the Simulation Model

We use a lung cancer model presented in a published cost-effectiveness analysis[14] as a fast benchmark for BO on simulation models. This Markov-based microsimulation model simulates a cohort's progression through six different health states from 35 to 79 years of age, in monthly intervals. The transition probabilities used in the model were age-specific, with distinct values for each 5-year age group (i.e. 35-39, 40-44, ..., 75-79). The state diagram for this model is pictured in figure 1.

Certain inherent constraints, such as ensuring that the sum of the probabilities in each row equals one or that certain probabilities are zero, were imposed on the matrices. This allowed the number of parameters to be optimized per age group to be reduced from 36 to 11. From the nine age groups, each one with a set of 11 parameters, only the first few of these were calibrated in this study. As a result, the problem was simplified to the calibration of 11 parameters, rather than the original $11 \cdot 9 = 99$ parameters associated with the full simulation. Furthermore, this model was designed to be computationally inexpensive, taking less than 10ms to simulate. By introducing arbitrary delays in the model we can observe the relationship of optimization times and model simulation times for different optimization methods.

The calibration target for the model was defined as the weighted sum of the euclidean distances between the observed and expected outputs of interest, namely lung cancer incidence (45%), lung cancer mortality (45%) and mortality from other causes (10%), computed for each age group.

3.2. Optimization Methods

In cost-effectiveness analysis, it is common to have an initial estimate of a good solution based on approximate values found in the scientific literature. For all optimization experiments conducted, a solution space of plus or minus $\pm 50\%$ was considered for each input variable, centered around this initial value.

First, we used different optimization methods to illustrate the performance differences between regular BO and classical methods. For this purpose we used python implementations of commonly used methods: a hill-climbing technique (Nelder-Mead[2][15]), metaheuristics (Simulated Annealing (SA)[3][16] and Particle Swarm Optimization[4][17]), and BO with GPs[5]. The default hyperparameter values were used for these methods, except for Particle Swarm Optimization, where the number of particles was set to 1,000 times the number of age groups calibrated.

For a second, distinct set of experiments we developed a new BO implementation with GPs from scratch using the R programming language. This implementation was used as a rapid prototyping environment to evaluate different enhancements to the optimization process for our specific domain, without being concerned by execution time at this stage. This implementation uses the Expected Improvement acquisition function, with Particle Swarm Optimization to search for the maximum. Finally, both the SE and the OAK kernels were implemented. Other runtime optimizations such as GPU use are beyond the scope of this work.

Note that the methods implemented in python were used exclusively for a fair execution time comparison between optimization methods in a common python runtime environment, while the methods implemented in R were used only for a fair error comparison among BO alternatives, without considering execution time.

3.3. Hyperparameter tuning

Before starting the BO procedure we learn the lengthscales $l_1, ..., l_D$ and the variances $\sigma_0^1, \sigma_1^2, ..., \sigma_n^2$ of the OAK kernel in a two stage process. This approach allows us to break down a potentially complex hyperparameter tuning task for high-dimensional problems into low-dimensional, manageable problems.

In the first stage, lengthscales are found by maximizing the marginal likelihood for each dimension separately. This approach is followed due to the implicit assumption that our simulation models have a strong additive component of order 1 and that a linear combination of one-dimensional kernels can be a reasonable approximation. These optimization subtasks are D simple univariate convex problems: for small and large lengthscales the kernel overfits and underfits, respectively, producing low-likelihood models. Each optimum is a unique value between these two extremes, quick to find using simple binary search.

In the second stage, the marginal likelihood is maximized for the whole set of variances, given the previously found lengthscales. This is a $(D+1)$-dimensional optimization subtask, solved using Nelder-Mead[15].

[2] `https://docs.scipy.org/doc/scipy/reference/optimize.minimize-neldermead.html`

[3] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.dual_annealing.html`

[4] `https://pyswarms.readthedocs.io/en/latest/`

[5] `https://secondmind-labs.github.io/trieste/1.1.2/index.html`

4. Results

Our interest in these experiments are about the comparative performance of BO versus the rest of methods used in the calibration of simulation models. We can see the relationship between simulation time and calibration time for each method in figure 2. For very fast models the inference overhead of BO dominates and other methods are able to calibrate faster by simulating the model many times. However, as the simulation time increases, the Bayesian method efficient approach in number of function evaluations results in faster calibration times. Specifically, for a model with 11 parameters and simulation times of less than 250ms, we observed the Bayesian approach outperform the alternative methods.

In contrast, as the dimensionality of our problem grows, the Bayesian method overhead increased significantly, as shown in the y-intercept of figure 2. The calibration times for the other methods also increased but, overall, the simulation time threshold where the Bayesian approach outperforms the other methods increases exponentially with the number of parameters, from 0.2 seconds to 0.35, 0.95 and 3.25 seconds. The bottom left plot in figure 2 projects that Bayesian calibration of all 99 parameters would be the best technique when each simulation takes approximately 5 minutes of computation.

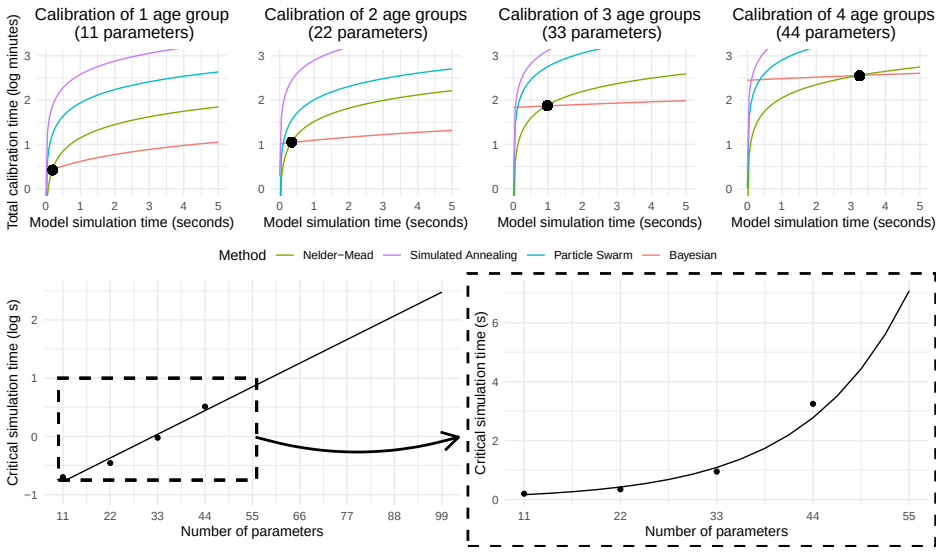

Figure 2. Total calibration time in log scale against model simulation time required to attain similar levels of error. The bottom left figure shows the exponential trend (in log scale) in the necessary simulation time before the bayesian method becomes the fastest method, as a function of the number of parameters. The bottom right figure is a zoomed-in plot of the same figure removing the log scale. The methods used in this comparison were all implemented in python.

In any case, the focus of our research in this work is the number of evaluations, where we see a sharp drop in error when using BO to achieve a similar level of accuracy compared to other methods, as shown in figure 3. Although each iteration requires a significant amount of time due to the Bayesian inference step, this overhead will become less relevant as the size of the model increases.

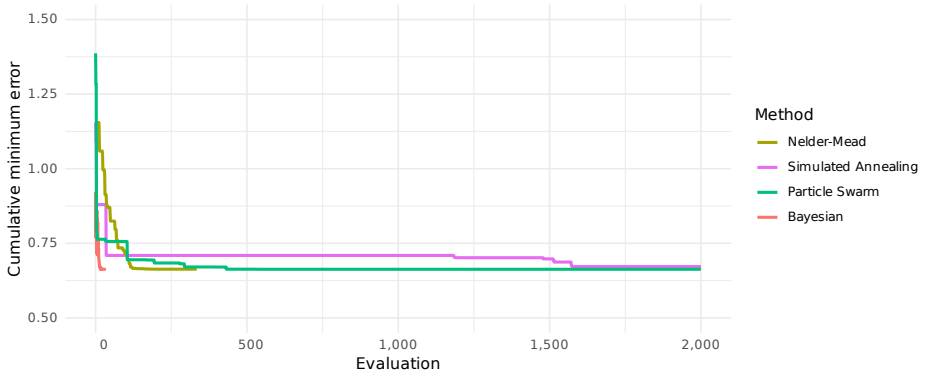

Figure 3. Time series of the lung cancer model calibration error (using 1 age group). The error is plotted against the number of evaluations by method.

While exploring the results of BO with OAK kernels we found that one of the variables had very significant explanatory power by itself, which could produce misleading results in the comparison. To address this issue, we introduced a third univariate SE kernel that considers only this variable. Figure 4 shows the average progression of the error during the optimization process for the three kernels and their interquartile range for a sample of 30 random executions. The univariate SE kernel shows a lower average error and lower spread than the full SE kernel, due to the reduction in dimensionality of the problem that allows for an easier exploration of the solution space, with barely any information loss. However, the OAK kernel under a normality assumption for the inputs is able to efficiently search the full 11-dimensional space to reach even better average results than the univariate SE kernel, while reducing the dispersion as the optimization progresses.

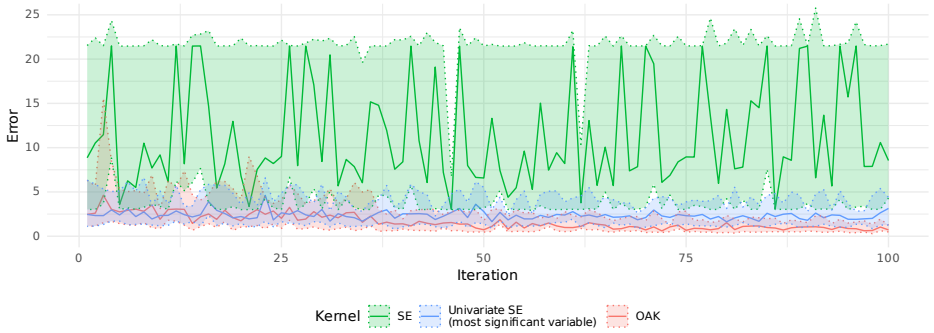

Figure 4. Time series of the median BO error with its interquartile range as the shaded area. We used three different kernels: the SE kernel (blue), the univariate SE kernel using only the most significant variable (green) and the OAK kernel under a normality assumption for the inputs (red). The methods used in this comparison were all implemented in R.

5. Discussion

BO is currently considered a state-of-the-art optimization method in various domains that involve costly function evaluations. Even though this result is already well known in the literature, an important research question would be determining the threshold at which the cost of the functions justifies the use of BO. Our findings show that simulation models with execution times of only a few seconds can be expensive enough to warrant the use of BO. In the cases when the function evaluations are not as costly, there are two other critical points in each BO iteration that must be taken into account: the surrogate model regression and the acquisition function optimization.

Regarding the former, SE kernels suffer greatly from the curse of dimensionality. In high-dimensional problems, the number of observations to explore the solution space quickly increases. This renders the regression unfeasible when trying to invert large Gram matrices to calculate the posterior predictive distribution. To address this issue, OAK kernels can reduce the number of necessary observations, mitigate the effects of an expanding Gram matrix and enhance the efficiency of the search. The observed scaling issues in figure 2, resulting from increasing dimensionality, justify the need for high-dimensional improvements such as OAK kernels. Nevertheless, our experiments showed that the asymptotic behaviour of BO persisted, making it the optimal choice for sufficiently large models.

A remarkable insight about additive kernels can be found in the comparison made in figure 4. As explained in the results, we considered a simulation model with eleven parameters, where one of the parameters was found to have a significant impact in the overall error. From a domain view, this parameter corresponds to the probability of death from other causes, which in this simulation has a greater impact on a greater amount of people than the rest. If we don't use additive kernels the optimization process has to explore all eleven dimensions and it is incapable of reducing the error over 100 iterations (green line). If we focus on this significant parameter by itself (blue line) we can see that the exploration finds much better solutions, but for more complex problems it might be difficult to manually isolate the important variables. The additive kernel (red line) is able to automatically detect this fact and perform what could be viewed as some kind of variable selection, while at the same time managing to refine better solutions with the rest of parameters.

The optimization of the acquisition function is the initial bottleneck, where the number of observations is still small enough so that the surrogate model regression is not yet a problem. As the search space remains constant, this acquisition optimization doesn't become much more expensive as more data is observed. We used Particle Swarm Optimization as an easy way to take advantage of parallelism in this area, but other approaches mentioned in the next section are being considered as well[10].

Lu et al[11] mention that an interesting direction of work would be to extend OAK kernels to BO leveraging the inferred low-order representation. In our tests we show that, even with a straightforward application of OAK kernels on this simple example, a slight improvement over the SE kernel is noticeable. This improvement is expected to be more meaningful for complex models, where more structure can be leveraged. It is interesting to note that these results hold even though some assumptions of the model were not met. Specifically, hyperparameter tuning was performed with a dataset sampled from a uniform input distribution, while the constrained kernels were calculated assuming

normality in the input. Even if these distributions were consistent, we still would have the problem of determining the input distribution for the actual optimization process, which would be neither normal nor uniform.

6. Future Work and Conclusions

One particular aspect that we did not incorporate into this article is constraint handling. Simulation models can be highly constrained problems, and these constraints are another expression of the structure of the solution space. We have been able to manage arbitrary constraints successfully using additional surrogate models and a new Constrained Expected Improvement acquisition function, as introduced by Gardnet et al[18].

We also mentioned in the discussion the convenience of exploiting the parallelization potential of the different areas of the optimization process. For that purpose we use the Particle Swarm method for the optimization of the acquisition function, but other more sophisticated venues for parallelization include batched optimization[19], parallel acquisition functions[20] or GPU approaches[21] among others.

Our research group recognizes the importance of efficiently calibrating increasingly complex models, injecting relevant domain knowledge in the process. In this work we have shown that using OAK kernels in a BO setting allows faster calibration times under very common circumstances in the health economics field. This is the beginning of several enhancements that are being implemented to have the tools to work with more challenging models in the future.

7. Acknowledgement

This work was supported by a grant from the Instituto de Salud Carlos III-ISCIII (Spanish Government) co-funded by European Regional Development Fund, a way to build Europe (CIBERESP CB06/02/0073, PI19/01118), also with the support of the Secretariat for Universities and Research of the Department of Business and Knowledge of the Government of Catalonia. Grants to support the activities of research groups (SGR 2017–2021). Grant number 2021SGR1029. We thank the CERCA Programme and Generalitat de Catalunya for institutional support.

References

[1] Drummond MF, Sculpher MJ, Claxton K, Stoddart GL, Torrance GW, Drummond MF, et al. Methods for the Economic Evaluation of Health Care Programmes. Fourth Edition, Fourth Edition. Oxford, New York: Oxford University Press; 2015. 464 p. doi: 10.1002/(SICI)1099-176X(199903)2:1¡43::AID-MHP36¿3.0.CO;2-7

[2] Levin HM. Cost-Effectiveness Analysis: Methods and Applications. 2nd edition. Thousand Oaks, Calif: SAGE Publications, Inc; 2000.

[3] Gray, Alastair M., Philip M. Clarke, Jane L. Wolstenholme, Sarah Wordsworth, Alastair M. Gray, Philip M. Clarke, Jane L. Wolstenholme, Sarah Wordsworth. Applied Methods of Cost-effectiveness Analysis in Healthcare. Handbooks in Health Economic Evaluation. Oxford, New York: Oxford University Press, 2010. doi: 10.1093/pubmed/fds009

[4] Weinstein MC, Torrance G, McGuire A. QALYs: the basics. Value Health. 2009 Mar;12 Suppl 1:S5-9.

[5] Garnett R. Bayesian Optimization. Cambridge: Cambridge University Press; 2023. doi: doi:10.1017/9781108348973

[6] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. Cambridge, Mass: MIT Press; 2006. 248 p. (Adaptive computation and machine learning). doi: 10.7551/mitpress/3206.001.0001

[7] Duvenaud D, Lloyd J, Grosse R, Tenenbaum J, Zoubin G. Structure Discovery in Nonparametric Regression through Compositional Kernel Search. In: Proceedings of the 30th International Conference on Machine Learning. PMLR; 2013. p. 1166–74.

[8] Bengio Y. On the challenge of learning complex functions. Prog Brain Res. 2007;165:521–34. doi: 10.1016/S0079-6123(06)65033-4

[9] Duvenaud DK, Nickisch H, Rasmussen C. Additive Gaussian Processes. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2011.

[10] Wilson JT, Hutter F, Deisenroth MP. Maximizing acquisition functions for Bayesian optimization. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc.; 2018. p. 9906–17. (NIPS'18).

[11] Lu X, Boukouvalas A, Hensman J. Additive Gaussian Processes Revisited. In: Proceedings of the 39th International Conference on Machine Learning [Internet]. PMLR; 2022 [cited 2022 Oct 5]. p. 14358–83.

[12] Durrande N, Ginsbourger D, Roustant O. Additive covariance kernels for high-dimensional Gaussian process modeling. Annales de la Faculté de Sciences de Toulouse. 2012;Tome 21(numéro 3):481.

[13] Binois M, Wycoff N. A Survey on High-dimensional Gaussian Process Modeling with Application to Bayesian Optimization. ACM Trans Evol Learn Optim. 2022 Aug 16;2(2):8:1-8:26. doi: 10.1145/3545611

[14] Diaz M, Garcia M, Vidal C, Santiago A, Gnutti G, Gómez D, Trapero-Bertran M, Fu M; Lung Cancer Prevention LUCAPREV research group. Health and economic impact at a population level of both primary and secondary preventive lung cancer interventions: A model-based cost-effectiveness analysis. Lung Cancer. 2021 Sep;159:153-161. doi: 10.1016/j.lungcan.2021.06.027

[15] Nelder JA, Mead R. A Simplex Method for Function Minimization. The Computer Journal. 1965 Jan 1;7(4):308–13. doi: 10.1093/comjnl/7.4.308

[16] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by Simulated Annealing. Science. 1983 May 13;220(4598):671–80. doi: 10.1126/science.220.4598.671

[17] Bonyadi MR, Michalewicz Z. Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. Evolutionary Computation. 2017 Mar;25(1):1–54. doi: 10.1162/EVCO_r_00180

[18] Gardner JR, Kusner MJ, Xu Z, Weinberger KQ, Cunningham JP. Bayesian optimization with inequality constraints. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. Beijing, China: JMLR.org; 2014. p. II-937-II–945. (ICML'14).

[19] González J, Dai Z, Hennig P, Lawrence N. Batch Bayesian Optimization via Local Penalization. In: Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS) [Internet]. 2016. p. 648–57. (JMLR Workshop and Conference Proceedings; vol. 51).

[20] Wang J, Clark SC, Liu E, Frazier PI. Parallel Bayesian Global Optimization of Expensive Functions. Operations Research. 2020;68(6):1850–65. doi: 10.1287/opre.2019.1966

[21] Wang K, Pleiss G, Gardner J, Tyree S, Weinberger KQ, Wilson AG. Exact Gaussian Processes on a Million Data Points. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2019.