**IBM Bayesian Optimization FAQ**

**Version 0.6**

Table of Contents

[What is “Bayesian optimization” and who is Thomas Bayes? 1](#_Toc21588077)

[If this math is well known, this isn’t new technology is it ? 2](#_Toc21588078)

[Does Bayesian optimization make applications run faster? 2](#_Toc21588079)

[Then if Bayesian optimization doesn’t accelerate an application – where is the benefit ? 2](#_Toc21588080)

[So, I have to keep and maintain my HPC environment to use Bayesian Optimization? 3](#_Toc21588081)

[I thought Bayesian Optimization was only for hyper-parameter optimization – not true ? 3](#_Toc21588082)

[Is IBM Bayesian Optimization Accelerator a machine learning product ? 4](#_Toc21588083)

[What are the alternatives to Bayesian optimization in use today ? 4](#_Toc21588084)

[What is different about IBM’s Bayesian Optimization Accelerator compared to other open source products on the market today ? 8](#_Toc21588085)

[Is there a comparison of IBM BOA vs. other offerings? 8](#_Toc21588086)

[How difficult is it to apply IBM Bayesian optimization to my design flow? 9](#_Toc21588087)

[How much faster will my design flow run if I use the Bayesian Optimization Accelerator? 9](#_Toc21588088)

[What industries and use cases has IBM applied IBM Bayesian Optimization Accelerator to with documented success? 10](#_Toc21588089)

[What execution architectures does IBM BOA support? 10](#_Toc21588090)

[Does it leverage accelerators like GPUs and is there an extra charge for those? 10](#_Toc21588091)

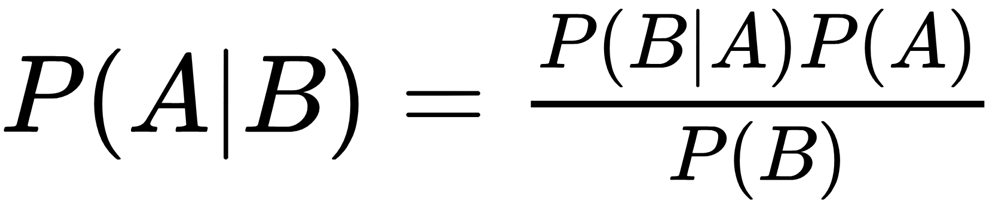
[What is the cost model for using IBM BOA? 10](#_Toc21588092)

[Is there a cloud service associated with IBM BOA ? 11](#_Toc21588093)

# What is “Bayesian optimization” and who is Thomas Bayes?

Thomas Bayes (1701 – 1761) was a young contemporary of Issac Newton 1642 – 1727), a philosopher, Presbyterian minister, and statistician.

Bayesian optimization is an application of a wider topic called “Bayesian statistics” all based on Thomas Bayes equation (sometimes referred to as Bayes Law, Bayes theorem, or Bayes rule).



At the highest level Bayesian optimization describes the likelihood of an outcome, based on data that is already known to be associated to the outcome. More concretively, the equations tell us how to calculate the probability of something happening if we have some data about a population where that something occurs.

If this math is well known, this isn’t new technology is it ?

Yes and no. The equation has been around for a long time – so that certainly isn’t new technology. However, our ability to apply this math to real world problems (high dimensionality) has been out of reach because the computation load goes up as the fourth power of the number of design variables.

As such, Bayesian optimization has been relegated to problems where the number of variables being explored was up to a couple dozen – after which the computational load was too high.

The scaling, and associated computational load, issue was a focus of the IBM UK Research team which set out to create an implementation of Bayesian Optimization that could be applied to several hundred to a few thousand design variables.

# Does Bayesian optimization make applications run faster?

No. This is the most common misconception of Bayesian Optimization performance claims. Any objective data acquisition , or simulation run executed in an HPC environmenthappens as it always has happened – no faster, no slower.

# Then if Bayesian optimization doesn’t accelerate an application – where is the benefit ?

Bayesian optimization provides a benefit in the case where an organization runs many simulations to explore a design space. Bayesian optimization is the most efficient way to explore a multi-dimensional design space to identify an optimal (minimum or maximum) design point within that space.

Relative to other design space exploration or search methods, a client using Bayesian optimization will achieve benefits from having (1) the best design inside the design space (results) with (2) the fewest number of simulations required to achieve the optimal result. Since the outcome/result is achieved with the fewest simulations, the user’s experience is that the design cycle time is drastically reduced.

# So, I have to keep and maintain my HPC environment to use Bayesian Optimization?

Yes you do.

As stated above, the individual simulations you run today will continue to run how and where they run when using Bayesian optimization. However, instead of a person modifying parameters and rerunning – or running “brute force” methods to survey a design space, Bayesian optimization of any kind will need to communicate with the HPC environment and make those parameter modifications automatically.



Traditional x86 HPC Cluster



***Shared Filesystem***

***Service Nodes***

*(Scheduling, Provisioning, GUIs, etc)*

BOA Cluster

***AC922***

***IC922 (Mihawk)***



**New Input**

**Sim Output**

# I thought Bayesian Optimization was only for hyper-parameter optimization – not true ?

Not true at all.

Hyper-parameter optimization (HPO) is a single use case of a parameter space defined by the hyper-parameters, and the response (objective function) is the quality rating of the machine learning model on the testing dataset.

Bayesian Optimization often gets used for tuning machine learning models because a simple formulation can be used due to the relatively small number of hyper-parameters. However, in a deep learning case where the number of neurons can be in the 1000’s or 10’s of 1000’s, a generic Bayesian approach becomes computationally intractable and another method, or an enhanced Bayesian method must be used to approach that problem.

# Is IBM Bayesian Optimization Accelerator a machine learning product ?

Yes and no. The product is not (currently) part of the Watson brand, and therefore is not part of the AI product family within IBM. However, if the question refers to the mathematics being done within the product – then BOA is definitely a machine learning product.

Like many machine learning tools, BOA automatically builds a model of the objective function (referred to as a “response model”) using Gaussian processes, and it updates this model after each objective function data acquisition (simulation). BOA then uses the response model to calculate the optimal place in the parameter space to run the next data acquisition.

# What are the alternatives to Bayesian optimization in use today ?

Generally there are three major methods, and other methods which are combinations of these three methods.

Method 1: Grid search

Method 2: Random search

Method 3: Gradient search

Pictures are worth 1000 words, so let’s take an example of a simple case with one design variable. For simplicity, let’s say that we’re designing the angle of a rear spoiler wing, where we want maximum downforce. The range of allowable angles are [0:-14] (units are degrees).

In the real world, where a client might be working on this problem, they would use a computational fluid dynamics simulator (e.g. openFOAM, Fluent, Star-CCM+, etc).

Please note: *In the real world, the shape of the design response (downforce, in orange) is very complex and is unknown. It is shown here for illustration purposes only.*

**Method 1: Grid search**

Angle of attack (deg)

-2 -4 -6 -8 -10 -12

Downforce (kN)

The grid search typically divides the domain into evenly spaced sampling points within and along the edges of the domain (shown here as open green circles). At each one of these points, the simulator is executed, and the objective function (downforce) is calculated (shown as green circles filled with black).

After all the grid points have been simulated, the objective function values are sorted, and the optimal design point picked as the maximum (in this case) or minimum value.

There are two major weaknesses of this approach:

1. There is no guarantee that the minimum or maximum value will fall on one of the grid points. As such, a true optimum will be missed
2. The search is incomplete until all the grid points are simulated – potentially wasting a lot of time and resources (compute cycles, software license time, user time).

The method is also non-adaptive, so once set, the spacing between grid points does not change.

Therefore, to a degree, the user is in a difficult position – they must set a grid spacing before they start their search, but the spacing needs to be small enough to capture the character of the objective function, but not so small as to waste tremendous resource.

The best spacing within the grid in a dimension is inversely proportional to the absolute value of the gradient of the objective in that dimension. Or more simply – the grid spacing needs to be smaller when the objective function changes quickly, and by implication denser grid spacing equates to additional simulations, which takes more time and resource.

**Method 2: Random search**

Angle of attack (deg)

-2 -4 -6 -8 -10 -12

Downforce (kN)

The random search uses a random number generator for each parameter dimension (shown here as open green circles) and relies on covering the domain with enough points to characterize the response shape. At each one of these points, the simulator is executed, and the objective function (downforce) is calculated (shown as green circles filled with black).

Typically a user chooses a number of points to randomly generate, and once all those points have been simulated, the objective function values are sorted, and the optimal design point picked as the maximum (in this case) or minimum value.

Random search has the advantage that there is no prohibition against having widely and densely spaced points in the same search, and therefore can capture high gradient situations in the response – however capturing such features regularly can only be done by running large numbers of points. This is because of the main disadvantage, which is that to a degree, fidelity with the response characterized by random search depends on luck.

There weaknesses of this approach are the same as grid search:

1. There is no guarantee that the minimum or maximum value will fall on one of the grid points. As such, a true optimum will be missed
2. The search is incomplete until all the grid points are simulated – potentially wasting a lot of time and resources (compute cycles, software license time, user time).

**Method 3: Gradient search**

Angle of attack (deg)

-2 -4 -6 -8 -10 -12

Downforce (kN)

The gradient search is the most sophisticated search of the three fundamental types discussed here. Generally it starts with a random point in the design space, but then it numerically calculates the gradient of the response (by running multiple points with very small perturbations in the parameter values - shown here as open blue circles). Once the gradient has been calculated (points used for calculating gradient shown as blue circles filled with black), an “educated” guess can be made as to the distance and direction to search for the optimum (shown here as open green circles). Once this new point is simulated, the process of re-calculating the gradients is performed, and a new guess is calculated and simulated. This loop stops when the gradient becomes within a tolerance near zero, or the maximum number of simulations is reached.

This method has the advantage that it’s possible it could converge to an optimal result very quickly if the design space is smooth and the number of design variables is small – however it too has some weaknesses

1. The method is vulnerable to local minima or maxima where the gradients go do zero in a local region within the parameter space.
2. Further this method can get very expensive (meaning it takes more time and resource) because at each point, the gradient calculation cost is 2N where N is the number of design parameters.

**Other methods**

Other methods which are simply combinations of the above search methods include but are not limited to:

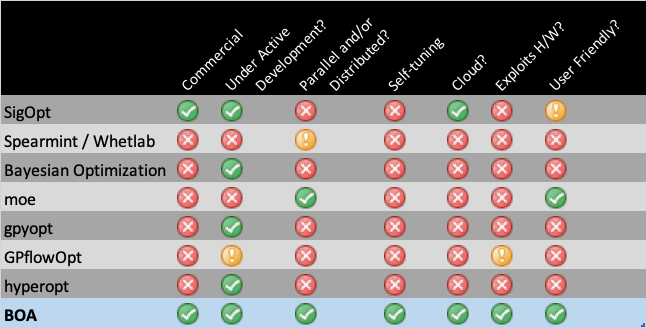
* Genetic algorithm search (grid search combined with random search)
* Mesh refinement search (successive levels of grid search with different spacing)
* Biased search (grid search with more/less densely spaced sampling points)

# What is different about IBM’s Bayesian Optimization Accelerator compared to other open source products on the market today ?

We can summarize some significant differences between IBM BOA and the standard approaches taken by other formulations:

1. Parallelization – We have made a large effort to parallelize the Bayesian method and selectively use a compression / reduction method so that we can scale to high dimensional problems.
2. Vectorization – some of the Bayesian calculations can now be accelerated by leveraging computational accelerators like GPUs
3. Explainability – In real time, BOA can provide explanations for each execution decision
4. Heterogeneous Agnosticism - The Bayesian Accelerator treats this challenge through the use of a platform, language, simulator, and discipline agnostic API interface

# Is there a comparison of IBM BOA vs. other offerings?



# How difficult is it to apply IBM Bayesian optimization to my design flow?

That depends a lot on the simulator being used, the complexity of the constraints and response (objective function) and how to calculate each.

IBM BOA uses a set of plugins called “Interface functions” which are available via a publicly accessible GitHub repository and community. These interface functions are application (simulator) specific.

They perform three functions:

1. Read output data from the simulator for the purpose of calculating the response
2. Create new input data for execution of the next simulation
3. Submit simulation jobs to the workload management system in the HPC

IBM Systems lab services also offers a service to create interface functions for a client application, but because of the range in complexity these engagements can be as short as 3 days to 2 weeks of consulting time.

# How much faster will my design flow run if I use the Bayesian Optimization Accelerator?

The speedup that Bayesian Optimization Accelerator can offer depends considerably upon the methods being used currently. We have seen a wide range of speedups from a factor of 3x faster in the Unilever computational chemistry case, to +100x speedup in the case of the IBM electronic design for signal integrity in the Power10 chip.

# What industries and use cases has IBM applied IBM Bayesian Optimization Accelerator to with documented success?

Currently (October 2019) IBM’s Bayesian Optimization Accelerator prototype has been applied to the following industries and use cases:

|  |  |  |
| --- | --- | --- |
| Industry | Use Case | Result Summary |
| Manufacturing | Lubricant mixing | 1/3 the simulations required to achieve 4 orders of magnitude better resolution |
| Drug Discovery | Malaria drug toxicity | Reduced a 20k compound database to 200 compounds being tested |
| Electronic Design | Signal Integrity | Minimized the number of simulations required from +1100 to 9 |
| Automotive | Truck Design / F1 wing design | Performed a design optimization in 24 hours that took a person nearly 3 weeks |

# What execution architectures does IBM BOA support?

IBM BOA supports the AC922 (Dual socket Power9) server with 2,4 or 6 (water cooled) GPUs. For large BOA installations we support small (up to 16 servers) clusters where IC922 machines do the management and communication functions, and the AC922s do the BOA calculations.

# Does it leverage accelerators like GPUs and is there an extra charge for those?

Yes it does use GPUs. We use a combination of TensorFlow and PyTorch to make use of GPU acceleration for various methods.

There is no extra charge for using GPUs, nor any additional charges for upgrading a server with additional GPUs.

# What is the cost model for using IBM BOA?

IBM BOA uses a subscription model and will be charged on a per-server per-year basis.

# Is there a cloud service associated with IBM BOA ?

Not today. This is something we are considering for the roadmap of the product, but we feel that the most important thing is to get BOA into the market today and get HPC users leveraging the technology. Once it is established as a quantum leap forward in design space search, then we will explore other delivery methods, the primary alternative being a cloud service.