# SELECTION OF OUTPUT MODE FOR BAYESIANOPTIMIZATION IN NOISY PROBLEMS

A PREPRINT

**Chenxi Li***
Department of Information Systems and Management Engineering
Southern University of Science and Technology
Shenzhen, China
12333226@mail.sustech.edu.cn

April 11, 2024

**ABSTRACT**

This paper investigates how to select optimal result outputs within noisy problem domains in Bayesian optimization. In noise-free problems, we can directly use the optimal observation point as the output result of the optimization process. However, in noisy environments, where all observations are subject to noise, the direct utilization of this result output may be inappropriate. Recognizing that the surrogate model can furnish the predicted mean at any point within the search space, this enables the adoption of predictive output methods. For instance, one can choose to output the point with the best predicted mean among all observed points or the point with the best predicted mean across the entire search space. Predictive output methods are likely more suitable than directly outputting observed values in high-noise problems. Nevertheless, the choice of the output method is not solely contingent on the magnitude of the noise; factors such as the response surface properties of the objective function and the dimensionality of the search space also exert an influence. This article conducts numerical experiments using the BBOB function group to explore Bayesian optimization problems with noise, aiming to offer insightful recommendations for the selection of output methods.

***Keywords*** Bayesian optimization · Gaussian process · blackbox optimization · noisy problems · BBOB

## 1 Introduction

Bayesian optimization stands out as an advanced optimization framework within the realm of black-box optimization. Its distinctive feature lies in its ability to derive solutions close to the global optimal value with a reduced number of objective function evaluations. This efficiency makes it a widely adopted approach in diverse domains, including drug research and development, as well as cost-effective optimization challenges such as the adjustment of neural network parameters and other objective function evaluations [Shahriari et al., 2016].The algorithm is a model-based sequential optimization. At each iteration, Bayesian optimization will formulate a strategy that balances exploration and development based on the currently available information to select the next evaluation point. Specifically, Bayesian optimization mainly consists of two parts: the surrogate model and the acquisition function. The surrogate model generally uses Gaussian process [Rasmussen and Williams, 2005] to model the objective function based on existing observation points and their observation value information. The acquisition function can then use the established model to select the location of the next observation point, thereby achieving the best possible optimization results with a smaller number of observations.

Bayesian optimization has demonstrated its efficacy in achieving favorable optimization outcomes, particularly in problems characterized by low dimensionality, absence of noise, and expensive black-box optimization problems.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

While Bayesian optimization remains applicable in the noisey objective function, certain adjustments to the algorithm settings may be necessary. One of the most intuitive problems is that when there is noise in the objective function, the observation value corresponding to the observation point will no longer be true. This issue can impact the optimization effectiveness of Bayesian optimization in two key aspects:acquisition function and optput results. Presently the most commonly used acquisition function is the EI [Jones et al., 1998] which based on the optimal value improvement expectation. That is, by calculating the expected improvement of a certain point relative to the current optimal value, the point with the largest expectation improvement is selected as the next observation point. When the noise is large, the current optimal value may be seriously disturbed, thus affecting the point selection strategy of EI, making it tend to over-explore.Smilarly, noise also affects our final output, as the minimum observations in the case of noise are not accurate. An intuitive solution to these problems is to replace the observed optimal value with the predicted mean optimal value of the surrogate model. Here the predicted mean is an inference of the surrogate model about the objective function based on the existing information. In noisy cases, especialy large noise, such predictions tend to be more reliable than direct observations. The main purpose of this article is to explore the conditions under which we should trust the observations, and under what conditions we should trust the predictions. There doesn't seem to be a detailed discussion of this in the current article, so this article hopes to explore this issue in depth through experiments. In order to make the experimental conclusions more representative, the test questions in this paper choose to use the mature test benchmark environment "BBOB", which contains 24 noise-free test functions. We modified it to be in a noisy form as a benchmark question for the test.The detail of BBOB functions can be found in [Hansen et al., 2010]

This article is divided into the following sections: The first part is a detailed introduction to Bayesian optimization, includeing surrogate model and acquisition function. The second part is an introduction to BBOB benchmark functions and experimental settings. In the third part, we discussed the use of observations and predictions in a noise-free environment value as the difference in optimization effect as the result output. The fourth part is a noisy experiment.

## 2 Bayesian optimization

Assume f (x) is an unknown target black-box function, Consider the optimization problem:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x) \tag{1}$$

where X is the disign space assumed compact. The objective function f(x) has no explicit expression, for any $x_0 \in \mathcal{X}$(observation point), the corresponding objective function value $f(x_0)$ can be calculated(observation value).Bayesian optimization frameworks commonly employ Gaussian processes (see Section 2.1) as surrogate models to construct the posterior distribution of the objective function, providing a high degree of flexibility. With Gaussian process modeling, the model can yield the predicted mean and uncertainty for any point in the search space. Leveraging these predicted values and uncertainties, we can design acquisition functions (see Section 2.2) to balance exploration and exploitation, thereby selecting the next observation point. The specific algorithm of Bayesian optimization is outlined in **Algorithm1**.

Note that **Algorithm1** provides the standard Bayesian optimization algorithm under the assumption of a noise-free condition. In the presence of noise,outputting the point corresponding to the minimum observed value as the result might not be appropriate. Two alternative predictive output methods may be considered: outputting the point with the smallest predicted mean among the observation points or outputting the point with the smallest predicted mean across the entire design space as predicted by the surrogate model. In the subsequent experimental section, we will analyze and compare the optimization performance of these three output methods.

**Algorithm 1:** Bayesian optimization

**Input:** number of initial point $n_0$, number of max iteration $n$

Randomly get initial data $D_{1:n_0}$ and update GP model;

**for** $t = 1, 2, ...n$ **do**

  Find $x_t$ by optimizing the acquisition function over the GP $x_t = \arg\max_x u(x|D_{1:n_0+n})$;

  Sample the objective function: $y_t = f(x_t) + \epsilon_t$;

  Augment the data $D_{1:n_0+n} = D_{1:n_0+n-1}, (x_t, y_t)$ and update GP model;

**end**

**Result:** $(x_{t^*}, y_{t^*}) \in D_{1:n_0+n}, t^* = \arg\max y_{1:n_0+n}$

### 2.1 Gaussian process

Gaussian process is a nonparametric model that is fully characterized by its prior mean function $m(x) : \mathcal{X} \to \mathbb{R}$ and positive definite kernel functions(Can also be thought of as a covariance function)$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. For any finite set of points within the search space $x_{1:n}$, define $f_i := f(x_i)$ as the corresponding value of the objective function at $x_i$, $y_i := f(x_i) + \epsilon_i$ is the corresponding noisy observation at $x_i$.In the Gaussian process model, we assume that $\mathbf{f} := f_{1:n}$ obeys a joint Gaussian distribution, and that $\mathbf{y} := y_{1:n}$ obeys a normal distribution given $\mathbf{f}$, which is mathematically expressed as follows:

$$\mathbf{f} \mid \mathcal{X} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}) \tag{2}$$

$$\mathbf{y} \mid \mathbf{f}, \sigma^2 \sim \mathcal{N}(\mathbf{f}, \sigma^2\mathbf{I}]) \tag{3}$$

The peculiarity of this model is that we treat the objective function $f(x)$ as a random variable, and equation(2) represents the prior distribution of the variable $f(x)$. where $textbfm$ represents the prior mean function, and $K_{i,j} := k(x_i, x_j)$ is the covariance matrix. After obtaining the observation data $mathcalD_n = (x_i, y_i)_{i=1}^n$, $f(x)$ given the observed data $mathcalD_n$, for the new value to be predicted $f(x^*)$, we have the following joint distribution:

$$\begin{bmatrix} \mathbf{y} \\ f(x^*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{m}, \begin{bmatrix} \mathbf{K} + \sigma^2\mathbf{I} & \mathbf{k}(x^*) \\ \mathbf{k}(x^*)^T & k(x^*, x^*) \end{bmatrix}\right)$$

Using the conditional probability distribution formula of normal distribution, it is easy to deduce the mean and variance functions of $f(x^*)$ as follows:

$$\mathbf{m}_n(x^*) = \mathbf{m}(x^*) + \mathbf{k}(x^*)^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}(\mathbf{x})) \tag{4}$$

$$\sigma_n^2(x^*) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}) \tag{5}$$

The $\mathbf{k}(x^*)$ in the formula is the vector of covariance between $x^*$ and $\mathbf{x}_{1:n}$. Using the Gaussian process model, we can obtain the predicted mean and predicted variance for each point in the search space, which provides an important basis for the design of the acquisition function.

### 2.2 Expected improvement

BO has many choices of acquisition fuction, including Expected improvement[Jones et al., 1998], GP upper confiddences bound[Srinivas et al., 2009], Thompson sampling[Thompson, 1933], and Entropy search methods[Hennig and Schuler, 2012]. In this article, we will focus on the most commonly used method EI.The main idea of EI is to select the point with the greatest improvement expectation compared to the current optimal value as the next observation point. Specifically, we first define the improvement function **I**(improvement function):

$$I(\mathbf{x}, v, \theta) := (v - f_n^*)\mathbb{I}(v > f_n^*) \tag{6}$$

where $v \sim \mathcal{N}(m_n(\mathbf{x}), \sigma^2(\mathbf{x}))$, $\mathbb{I}$ is indicator function, $\theta$ is the hyperparameter set $f_n^*$ is the current optimal value, $m_n(\mathbf{x}), \sigma_n(\mathbf{x})$ are respectively the values returned by the Gaussian process model for the input value **x** Predicted mean and predicted variance. Take expectation for $v$ to get the acquisition function EI:

$$u(\mathbf{x}; D_{1:n}) := \mathbb{E}[I(\mathbf{x}, v, \theta)] = (m_n(\mathbf{x}) - f_n^*)\Phi\Big(\frac{m_n(\mathbf{x}) - f_n^*}{\sigma(\mathbf{x})}\Big) + \sigma_n(\mathbf{x})\phi\Big(\frac{m_n(\mathbf{x}) - f_n^*}{\sigma_n(\mathbf{x})}\Big) \tag{7}$$

Where $\Phi$ and $\phi$ are the cumulative distribution (CDF) and probability density distribution (PDF) of the standard normal distribution respectively.

When there is noise in the objective function, the current optimal value $f_n^*$ is not the real value. When the noise is large, the value of $f_n^*$ may be more extreme, causing EI to mistakenly believe that the The vast majority of points have minimal improvement expectations, making EI overly biased towards exploration rather than exploitation. One of the most intuitive solutions is to use the minimum predicted mean $m_n^*$ of the surrogate model in the observed points. Replace $f_n^*$. This actually improves EI from optimizing the optimal observation value with noise to optimizing the optimal predicted value of the surrogate model, thereby improving the algorithm's robustness to noise. We record the improved EI as EIm, and its expression is as follows:

$$u'(\mathbf{x}; D_{1:n}) := (m_n(\mathbf{x}) - m_n^*)\Phi\Big(\frac{m_n(\mathbf{x}) - m_n^*}{\sigma(\mathbf{x})}\Big) + \sigma_n(\mathbf{x})\phi\Big(\frac{m_n(\mathbf{x}) - m_n^*}{\sigma_n(\mathbf{x})}\Big) \tag{8}$$

$\mu^+$ is the current optimal value, $\mu_n(\mathbf{x}), \sigma_n(\mathbf{x})$ are respectively returned by the Gaussian process model for the input value $\mathbf{x}$ The predicted mean and predicted variance.In fact, there are many other variants of EI designed to deal with noisy problems, such as Augmented expected improvement(AEI)[Huang et al., 2006], the reinterpolation procedure(RI)[Forrester et al., 2006], Expected quantile improvement(EQI)[Picheny et al., 2012] and so on. The behavior of these methods has been detailed discussed in [Picheny et al., 2013]. The experiments in this article will not involve too many variations of EI, since there is no direct relationship between the output method of the final result and the point selection method of the acquisition function.

In subsequent experiments, we will use EI for the output method based on observed values; and use EIm for the output method based on predicted values.

## 3 Test functions and experimental settings

### 3.1 BBOB benchmark functions

The test function in this article is selected from the 24 test functions of the BBOB noiseless test function group, and noise is added as our test benchmark problem. We number these functions F1-F24. According to the settings in the original BBOB text, these functions are divided into 5 types: separable functions (F1-F5), Functions with low or moderate conditioning (F6-F9),Functions with high conditioning and unimodal (F10-F14), Multi-modal functions with adequate global structure (F15-F19), Multi-modal functions with weak global structure (F20-F24). All functions in the BBOB test function group can obtain new test functions through changes such as translation or rotation. In the subsequent experiments of this article, we fixed these 24 noise-free functions in advance. The optimal values, standard deviations and simple descriptions of these functions are shown in Table 1. More detailed information about them can be found in[Hansen et al., 2010].

| 2Dfunctions | optimal value | std | comment |
|---|---|---|---|
| F1 | 79.48 | 12.57 | sphere function,unimodal,presumably the most easy continuous domain ssearch problem |
| F2 | 66.95 | 10044455.67 | Globally quadratic and ill-conditioned(about $10^6$) function with smooth local irregularities.Conditioning is about $10^6$ |
| F3 | 77.66 | 418.57 | Highly multimodal function with a comparatively regular structure for the placement of the optima. |
| F4 | 77.66 | 171.86 | Highly multimodal function with a structured but highly asymmetric placement of the optima. |
| F5 | 66.71 | 29.02 | Purely linear function,solution is on the domain boundary |
| F6 | 65.87 | 237396.11 | Unimodal,highly asymmetric function, |
| F7 | 92.94 | 431.55 | unimodal, non-separable, conditioning is about 100.The function consists of many plateaus of different sizes. |
| F8 | 98.62 | 46480.79 | (Rosenbrock function) in larger dimensions the function has a local optimum with an attraction volume of about 25% |
| F9 | 65.61 | 21715.05 | rotated version of the previously defined f8. |
| F10 | 59.13 | 9634378.45 | rotated version of the previously defined f2. |
| F11 | 76.27 | 21241083.28 | A single direction in search space is a 1000 times more sensitive than all others.Conditioning is about $10^6$ |
| F12 | 56.61 | 9607260659 | conditioning is about $10^6$, rotated, unimodal |
| F13 | 68.42 | 449.99 | Resembles f12 with a non-differentiable bottom of valley |
| F14 | 77.31 | 41.04 | The sensitivies of the $z_i$-variables become more and more different when approaching the optimum |
| F15 | 70.03 | 521.74 | Prototypical highly multimodal function which has originally a very regular and symmetric structure for the placement of the optima. |
| F16 | 71.35 | 79.07 | Highly rugged and moderately repetitive landscape, where the global optimum is not unique. |
| F17 | 69.83 | 19.00 | A highly multimodal function where frequency and amplitude of the modulation vary.Conditioning is low |
| F18 | 119.54 | 308.17 | Moderately ill-conditioned counterpart to f17 |
| F19 | 71.69 | 74.72 | Resembling the Rosenbrock function in a highly multimodal way. |
| F20 | 71.29 | 45818.02 | The most prominent 2D minima are located comparatively close to the corners of the unpenalized search area. |
| F21 | 124.08 | 12.21 | The function consists of 101 optima with position and height being unrelated and randomly chosen. |
| F22 | 51.57 | 24.05 | The function consists of 21 optima with position and height being unrelated and randomly chosen.Conditioning is about 1000 |
| F23 | 85.39 | 20.12 | Highly rugged and highly repetitive function with more than 10D global optima. |
| F24 | 93.30 | 18.18 | Highly multimodal function with two funnels. |

Table 1: BBOB functions

### 3.2 Experimeental settings

- **Dimension and budget:** The function dimensions in the experiment are divided into 2 dimensions and 4 dimensions. For two-dimensional problems, the total budget for a single experiment is 100 observation points under noise-free conditions, and increases to 300 observation points when there is noise. For four-dimensional problems, the total budget is 200 under no-noise conditions and increases to 400 under noisy conditions.
- **Initialization:** Following the result of [Bossek et al., 2020], we set the initial number of points for Gaussian process modeling to $10\%$ of the total budget, and use Latin hypercube sampling to randomly select points in the search area.

- **loss rate** Denote $y_a$ as the true value of the result obtained by the optimization algorithm, $y_{opt}$ as the true optimal value. We define the loss rate:

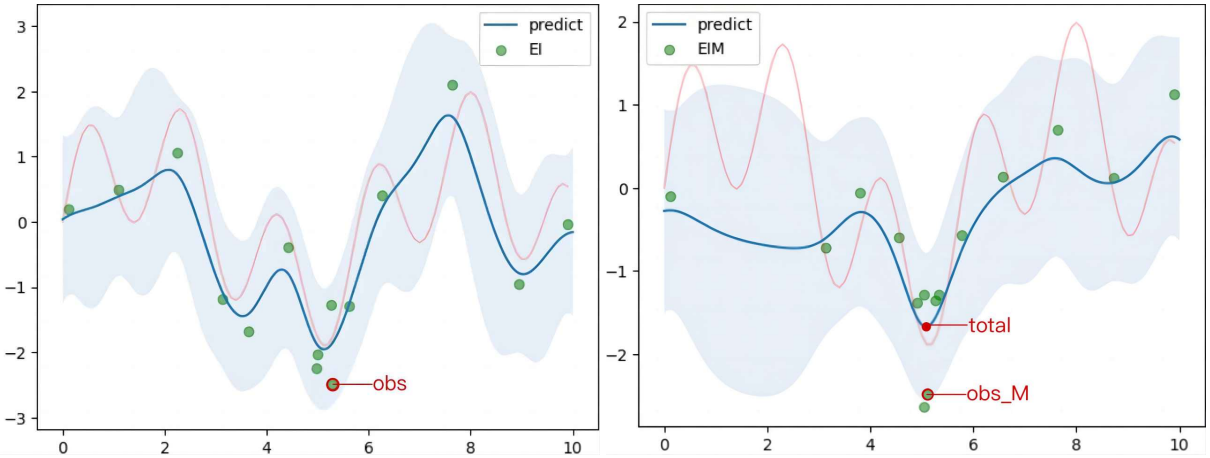
$$loss = \frac{y_a - y_{opt}}{y_{opt}} \times 100\% \tag{9}$$

as the optimization performance evaluation index of the algorithm. Note that the minimum values of the 24 functions we tested are all positive and do not approach 0, so the loss rate is well defined. Similarly, we can also define the relative loss rate between the two algorithms:

$$relloss = \frac{y_1 - y_2}{y_2} \times 100\% \tag{10}$$

where $y_1$ is the real value of the result obtained by optimization algorithm 1 and $y_2$ is the real value of the result obtained by optimization algorithm 2. If the relative loss rate is negative, it means that the optimal result of Algorithm 1 is better than Algorithm 2.

- **Randomness processing:** In order to make the results more general, we will use different random seeds to repeat 30 times for each set of experiments and then average the results. The randomness in the Bayesian optimization process is mainly reflected in the initial point position and the internal optimization of the acquisition function.
- **Noise settings:** The noise added to the test function in the experiment of this article is all Gaussian white noise, and its mean value is 0. The noise level is expressed in terms of the proportion of the function standard deviation($5\%$ for small noise, $20\%$ for moderate, $50\%$ for extremely noisy).
- **Acquisition function and output results:**Two forms of acquisition functions will be used in the experiment: EI optimized based on observed values and EIm optimized based on predicted values. As mentioned earlier, We will discuss three output methods:
**1.**Directly output the minimum observation value(Abbreviated as **obs**).
**2.**Output observation point with the minimum predicted mean.(Abbreviated as **obs_M**).
**3.**Output the point with the minimum predicted mean over the total design space.(Abbreviated as **total**).
The choice of output method will align with the selection of acquisition function. If the output method based on observed values (obs) is employed, then the Expected Improvement (EI) acquisition function will be used. Conversely, if the output method based on predicted values (obs_M and total) is employed, then we use EIM.




(a) Using EI for output method based on observation (b) Using EIM for output method based on prediction

Figure 1: The above two figures show the results of Bayesian optimization for a one-dimensional noise function. The pale red curve is the true function curve, and the blue one is the curve of the predicted mean function of the GP. The green dot is the observation point selected by the acquisition function. The acquisition function used in the left figure is EI, which corresponds to the output method 'obs', which will directly output the point corresponding to the smallest observation. The acquisition function for the right figure is EIM, which shows two outputs based on the predicted values. Method 'total' corresponds to the minimum value of the predicted mean function (blue curve), and 'obs_M' is the point with the largest predicted mean (blue curve) corresponding to the coordinates of all observed points (green dots).

# 4 Experimental results and analysis.

## 4.1 Noise-free experimental results

We first focus on experimental results for the noiseless problem. In the case of no noise, since the observed values of the observation points are all real values, the sequences $obs, obs\_M$ are exactly the same. In the same way, $output1 and output2$ are also completely consistent. We only need to compare the sequence $obs, total\_M$ to analyze whether the predictive sequence can contain better points, and whether $output2$ can output the better points as the final result. Of course, it is common sense to judge that $output1$ is the most reasonable output method under noise-free conditions. Our noise-free experiment primarily aims to explore the optimization effectiveness and observational output when using the predictive output method in noiseless conditions. The distinct disparities between the methods are detailed in Table 2:

| 2D | obs | total | total vs obs | 4D | obs | total | total vs obs |
|---|---|---|---|---|---|---|---|
| F1 | 0.00% | 0.00% | 0.00% | F1 | 0.00% | 0.00% | 0.00% |
| F2 | 5.71% | 59.70% | 50.71% | F2 | 6589.66% | 7861.45% | 585.31% |
| F3 | 10.22% | 27.81% | 16.80% | F3 | 26.95% | 56.09% | 23.06% |
| F4 | 5.01% | 18.19% | 12.62% | F4 | 29.90% | 63.32% | 26.56% |
| F5 | 0.00% | 0.00% | 0.00% | F5 | 0.00% | 0.00% | 0.00% |
| F6 | 1.52% | 3.45% | 1.89% | F6 | 31.23% | 105.48% | 58.40% |
| F7 | 0.08% | 0.18% | 0.10% | F7 | 0.05% | 0.09% | 0.05% |
| F8 | 0.07% | 1.75% | 1.68% | F8 | 4.33% | 5.16% | 0.84% |
| F9 | 0.12% | 3.66% | 3.54% | F9 | 5.76% | 11.17% | 5.25% |
| F10 | 15.30% | 1282.36% | 1133.40% | F10 | 454.52% | 4275.66% | 1253.11% |
| F11 | 8.50% | 719.52% | 656.81% | F11 | 22.84% | 2774.84% | 2255.78% |
| F12 | 1645.98% | 2907234.96% | 1209546.08% | F12 | 526280.42% | 27772809.81% | 11673.63% |
| F13 | 0.96% | 1.78% | 0.81% | F13 | 27.39% | 12.98% | -11.03% |
| F14 | 0.01% | 0.02% | 0.02% | F14 | 0.01% | 0.03% | 0.01% |
| F15 | 5.26% | 23.30% | 17.33% | F15 | 22.63% | 64.67% | 34.49% |
| F16 | 0.74% | 26.55% | 25.71% | F16 | 3.20% | 14.22% | 10.55% |
| F17 | 0.52% | 4.43% | 3.89% | F17 | 0.91% | 1.20% | 0.30% |
| F18 | 0.76% | 11.46% | 10.62% | F18 | 2.40% | 3.50% | 1.08% |
| F19 | 0.25% | 10.95% | 10.68% | F19 | 2.90% | 14.37% | 11.13% |
| F20 | 1.64% | 5.11% | 3.41% | F20 | 2.78% | 6.00% | 3.13% |
| F21 | 0.07% | 0.14% | 0.07% | F21 | 0.70% | 0.84% | 0.13% |
| F22 | 0.46% | 0.75% | 0.29% | F22 | 3.62% | 3.87% | 0.24% |
| F23 | 4.48% | 50.64% | 44.24% | F23 | 3.10% | 24.36% | 20.63% |
| F24 | 5.11% | 27.33% | 21.20% | F24 | 20.30% | 49.25% | 24.22% |

Table 2: Noiseless results

The first two columns in the table are the loss rate of the output methods $obs, total$ relative to the global optimal value of each function. The third columns is the relative loss rates of sequence $total$ and sequence $obs$ respectively. For 2D problems, It can be seen intuitively that the optimization performance of the predictive $total$ on functions F2, F10, F11, and F12 is very poor. Looking back at Table 1, we can find that these functions all have a common feature: the conditioning is extremely high (both exceed $10^6$). The corresponding standard deviations of these functions also have large values, which makes the corresponding noise values also large. Excessive noise can easily cover the original structural characteristics of the function, making the modeling of Gaussian process extremely difficult and unable to achieve a good prediction effect. In fact, observing Table 2, we can find that the loss rate of $obs$ corresponding to these four functions is also relatively large, which shows that ordinary Bayesian optimization will also be affected by low modeling accuracy when facing this type of problem. Therefore, we can conclude that for noise-free problems, if the modeling accuracy of the Gaussian process is not good, it is unacceptable to use predictive output methods. For simpler functions, judging from the results in the third column, the loss rate of $total$ relative to $obs$ is mostly above $10\%$. For more complex functions F10, F12, etc., the optimization results are even more unacceptable. Therefore, for noise-free problems, we should indeed directly use the observation-based Bayesian optimization result output method (The loss rate of $output3$ relative to $obs$ in all test functions is greater than 0). For 4D problems, the conclution remains same, however, after increasing the dimension, we find that the relative loss rate of A and B begins to decrease on many functions.This will be discussed later in the article.

### 4.2 Experimental results with noise problems

In the previous section, we briefly compared the optimization results of the observative and predictive outputs on 24 noise-free functions. In this section, we will pick out some functions and add noise to them as new optimization problems. In order to ensure the consistency of the conclusions, we will use EI as the acquisition function for the observable output method (obs), and EIM for the predictive output method (obs_M, total). It was noted that in the noisy problems, obs is no longer exactly the same as obs_M, as the observed value will no longer be true.(See Section 3.2 for details on noise settings). We will use the Instant Regret as the vertical axis of the figures, and the number of iterations of BO as the horizontal axis to generate curves for analysis. For example, in each iteration of BO, EI selects a point as the next observation. We subtract the optimal value of the function from the value of the real function corresponding to the selected point to get the instantant regret value of EI at the corresponding round. EIM can do the same for a similar curve, and for comparison, we have added a curve corresponding to the acquisition function (denoted random) completely choose points randomly. For our three output methods (obs, obs_M, total), they will also choose a point as the output result of the round in each iteration of BO, so we can also obtain the instantant regret curves of the three methods in the same way for analysis and comparison.

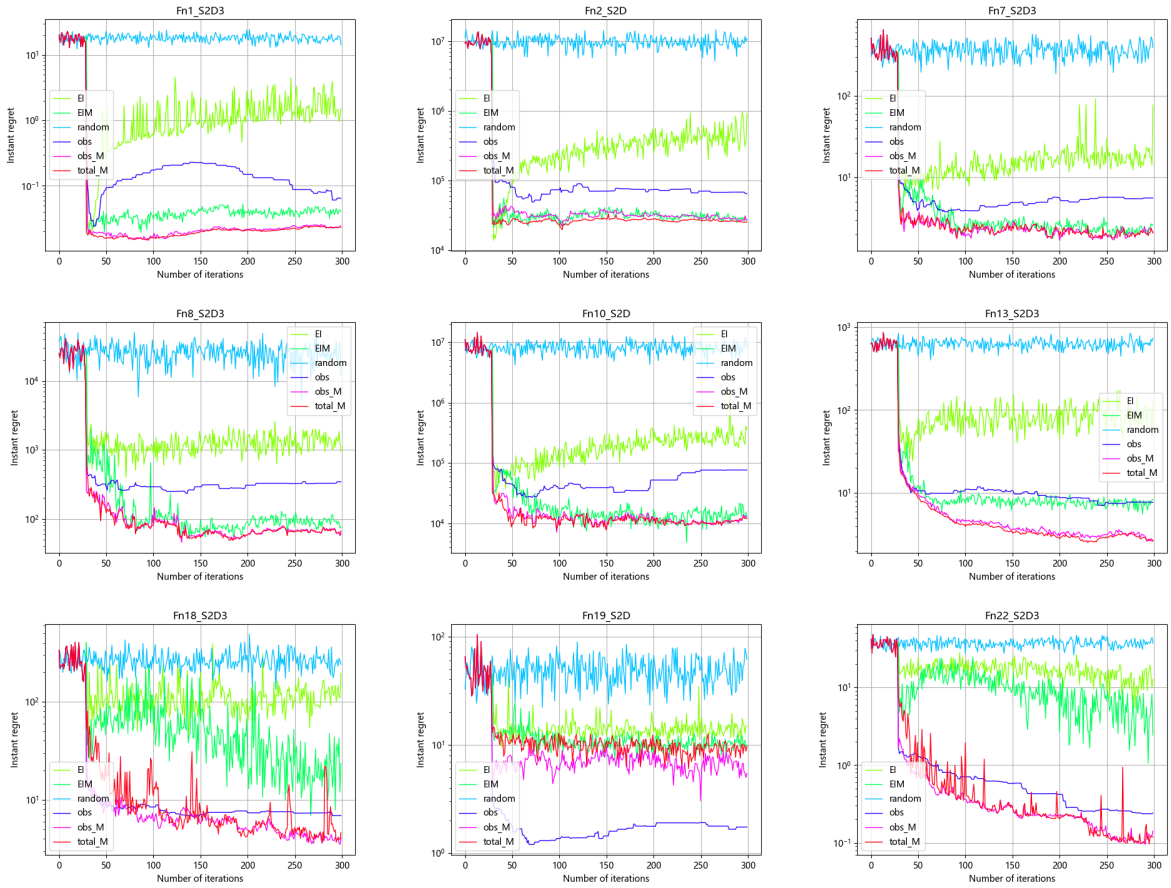



Figure 2: 2D optimization results(small noise)

In the two-dimensional small noise optimization result image, we can see that the curve of EI is always above the curve of EIM, which essentially reflects the difference between the two acquisition functions in terms of point selection strategy: EI is more inclined to explore and EIM is more inclined to development. There is no superiority or disadvantage between these two strategies, for different optimization problems they will have different optimization results. For the three output methods, since we all choose the output point after the budget is exhausted as the final output result, we can directly judge the quality of their output results by observing the average regret corresponding to the end of the curve.

Let's start with two predictive output methods. From Figure2, there is little difference between obs_M and total in the final optimization results, but the output of total shows great instability in some functions. For example, for the function Fn18, the output of total in turns 280-290 fluctuates greatly. Considering that our budget is actually arbitrary, if the budget happens to be between 280-290, the final output returned by total will be very bad. Compared with total, the output value of obs_M is more stable, so for two-dimensional problems with small noise, we recommend using obs_M instead of total if you want to use a predictive output method.

As for the observative output method obs, we can see from Figure2 that the final output result of obs is much worse than obs_M on most of the test functions, and its curve will always be above the curve of obs_M after fewer rounds of iterations. This means that for any arbitrary given budget, the final output of obs is worse than obs_M. However, in terms of the optimization of the test function Fn19, we found that obs has obvious advantages over other output methods. Further observation of the corresponding images of Fn19 shows that the curves of EI, EIM, obs_M, and total are basically clustered at the same height, which actually means that the GP model has a very poor modeling effect on function 19, the exploitation ability of the acquisition function is weak, and the prediction accuracy of the model is very low. In this case, the predictive output gives unreliable results. Therefore, for highly complex, low-accuracy modeling problems like Function 19, predictive output should be used with caution. For general noisy problems with good modeling accuracy, it is more reasonable to use predictive output method obs_M.

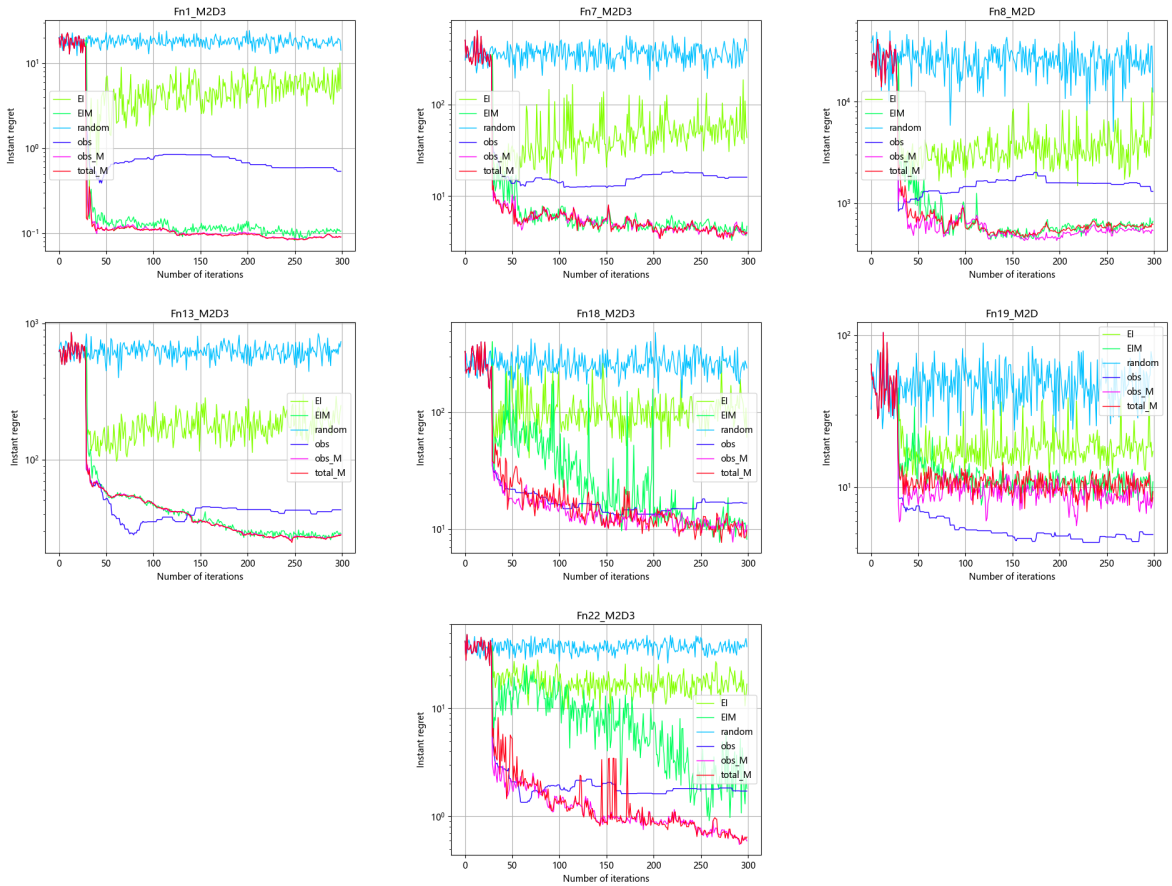



Figure 3: 2D optimization results(Medium noise)

## References

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016. doi:10.1109/JPROC.2015.2494218.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi:10.7551/mitpress/3206.001.0001. URL `https://doi.org/10.7551/`

mitpress/3206.001.0001.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455492, dec 1998. ISSN 0925-5001. doi:10.1023/A:1008306431147. URL https://doi.org/10.1023/A:1008306431147.

Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. Real-parameter black-box optimization benchmarking bbob-2010 : Experimental setup. 2010. URL https://api.semanticscholar.org/CorpusID:260830254.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. *arXiv e-prints*, art. arXiv:0912.3995, December 2009. doi:10.48550/arXiv.0912.3995.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. ISSN 00063444. URL http://www.jstor.org/stable/2332286.

Philipp Hennig and Christian J. Schuler. Entropy search for information-efficient global optimization. *J. Mach. Learn. Res.*, 13(null):18091837, jun 2012. ISSN 1532-4435.

D. Huang, Theodore Allen, William Notz, and Ning Zheng. Global optimization of stochastic blackbox systems via sequential kriging meta-models. *Journal of Global Optimization*, 34:441–466, 03 2006. doi:10.1007/s10898-005-2454-3.

Alexander I. J. Forrester, Andy J. Keane, and Neil W. Bressloff. Design and analysis of "noisy" computer experiments. *AIAA Journal*, 44(10):2331–2339, 2006. doi:10.2514/1.20068. URL https://doi.org/10.2514/1.20068.

Victor Picheny, David Ginsbourger, Yann Richet, and Grégory Caplin. Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*, 55, 03 2012. doi:10.1080/00401706.2012.707580.

Victor Picheny, Tobias Wagner, and David Ginsbourger. A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48, 09 2013. doi:10.1007/s00158-013-0919-4.

Jakob Bossek, Carola Doerr, and Pascal Kerschke. Initial Design Strategies and their Effects on Sequential Model-Based Optimization. *arXiv e-prints*, art. arXiv:2003.13826, March 2020. doi:10.48550/arXiv.2003.13826.

# Appendix

## A Non-homogeneous noise experimental results

### A.1 2-dimensional experiments

The basic settings of the experiments are the same as those in 4.2, with the only difference being that the noise settings are no longer homogeneous variance, but are set as follows:

$$f_{GN}(f, \beta) = f \times exp(\beta \mathcal{N}(0, 1)) \tag{11}$$

where $mathcalN(0, 1)$ represents random sampling from the standard normal distribution, and $beta$ is the parameter that controls the magnitude of the noise, here we set it to 0.1. Note that the value of noise here will be related to the true value of the function. Also similar to 4.2, we give the following image of the experimental results:

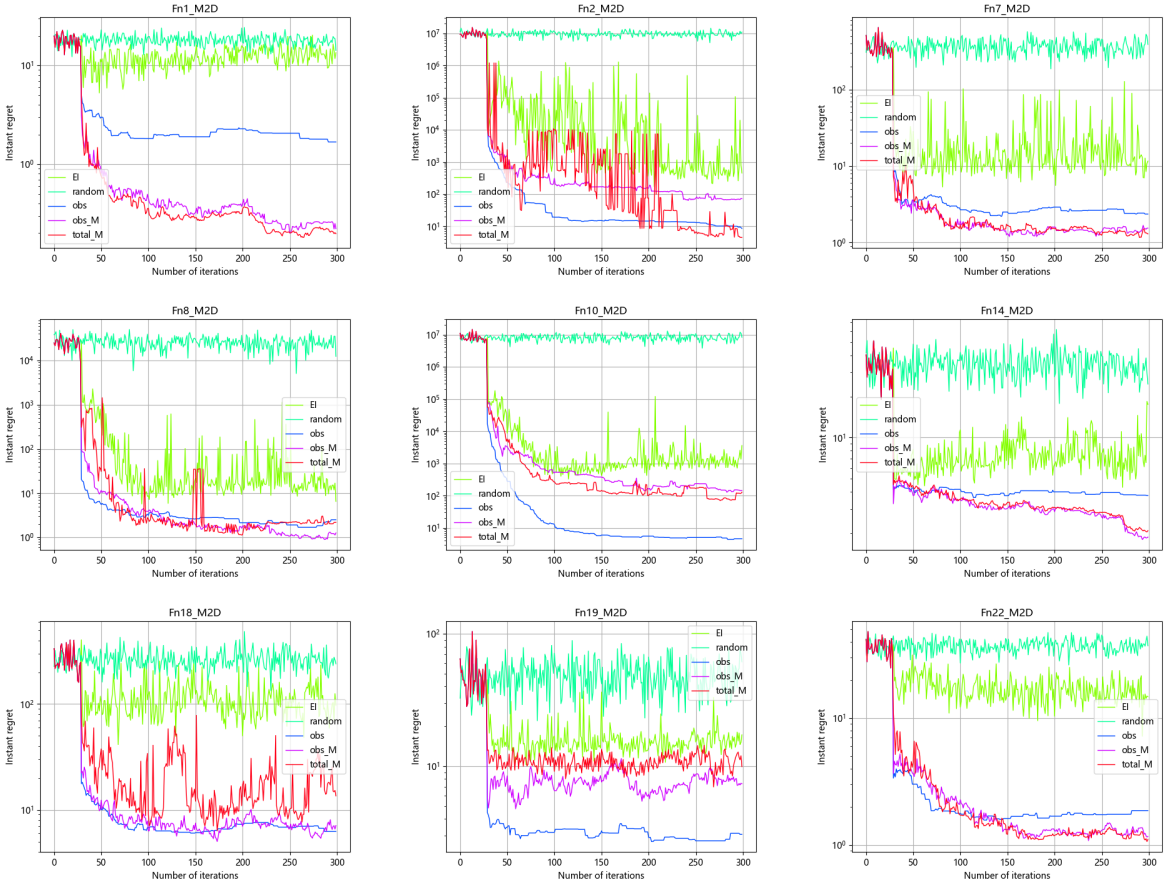



Figure 4: 2D optimization results(Non-homogeneous noise)

The results reflected in the figures are basically similar to those in 4.2, with the only difference being that for functions with large standard deviations (e.g., Fn10), the observative output methods in non-homogeneous noise is much better than the predictive output method. This is because when the noise value is positively correlated with the true value of the function, the point with the smaller value of the function corresponds to the less noise, so if we observe that the value of a point is small, it is most likely because the function value corresponding to that point is also small, rather than due to extreme noise. In this case, the observant output method is more reliable than the predictive output methods, even if noise is affected.

### A.2 4-dimensional experiments

Keeping the other settings unchanged, now we increase the dimension of the test function to 4 dimensions (the number of initial points and the total budget are increased to 40 and 400 respectively), and we get the following result:

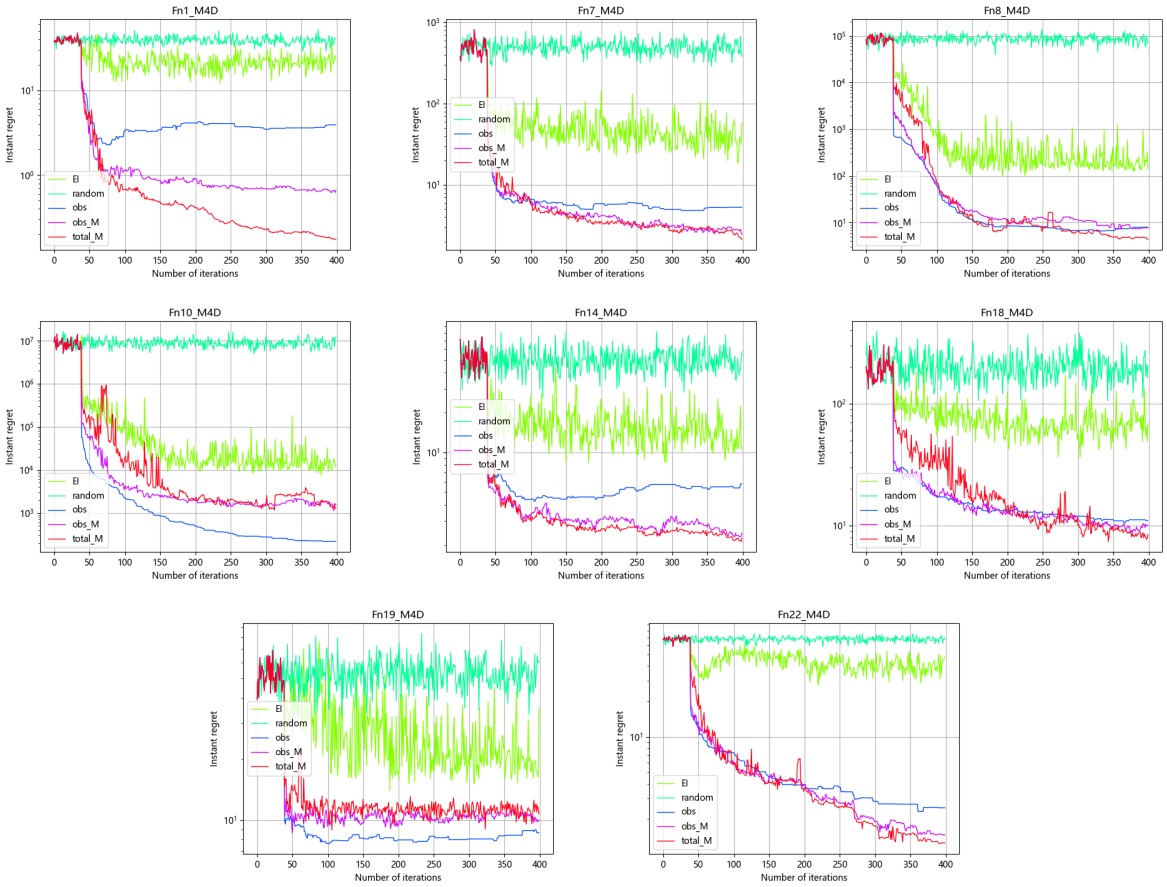



Figure 5: 2D optimization results(Non-homogeneous noise)

From the 4D image, we can draw conclusions that are basically the same as those of 2D. We noticed that after the dimension increased, total began to show a certain competitiveness compared with the other two output methods, especially in the simplest test function, Fn1, which was significantly ahead of the other two output methods. The characteristic of total is that it is a completely exploitation output method, although obs_M also output the maximum value of prediction mean, but its output is limited to the observations selected by the acquisition function EI. As we mentioned earlier, EI is an exploratory algorithm, and the size of the search space increases dramatically when the dimension is increased to 4D, so that the benefits of exploration in a limited budget may be much lower than that of exploitation. However, there is still some instability in the output result of total (for example, Fn18), and the final result is not significantly better than obs_M in most test functions, so we still do not recommend using the total output method under four-dimensional conditions.