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

July 2, 2024

ABSTRACT

This paper explores the method of Bayesian optimization for selecting the final output result in noisy optimization problems. For noise-free problems, the best observation can be directly output as the final optimization result. However, in the presence of noise, each observation is affected by the noise and it may not be appropriate to output the best observation directly. Considering that the surrogate models in Bayesian optimization are able to provide predicted mean for any point in the search space, and for noisy objective function, the predictions of the model tend to be more plausible than the actual observations. This inspired us to use a result output method based on the predicted values of thesurrogate models. For example, assuming that we want to find the minimum of the objective function, we can output the point with the lowest predicted mean among all observed points, or the point with the lowest predicted mean in the entire search space. In the case of high noise levels, these methods may be more appropriate than using the observations directly. However, the choice of output method depends not only on the noise level, but may also depend on factors such as the response surface properties and the dimensionality of the search space. In this paper, we will conduct numerical experiments using the BBOB function set to investigate the advantages and disadvantages of each output method for Bayesian optimization in different noise environments, and provide informative suggestions for the choice of output methods.

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

1 Introduction

Bayesian optimization stands out as an advanced optimization framework for black-box optimization. The advantage of this method is the ability to approximate the global optimum with fewer objective functions to evaluate. 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. In particular, Bayesian optimization is primarily made up of two parts: the surrogate model and the acquisition function. The surrogate model generally uses Gaussian process [Rasmussen and Williams, 2005] for simulating the objective function based on current observation points. The acquisition function can then use the established model to select the location of the next observation point, thereby optimizing outcomes more efficiently with fewer observations.

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

Bayesian optimization demonstrates its efficiency in many scenarios, especially in low-dimensional, noise-free and expensive black-box optimization problems. When the objective function is noisy, Bayesian optimization faces a critical problem in that the observations corresponding to the observation points are no longer accurate. This problem may affect the effectiveness of Bayesian optimization in two ways: the acquisition function and the output results. Presently the most commonly used acquisition function is EI [Jones et al., 1998] which based on the optimal value improvement expectation. That is, by calculating the expected improvement of a specific point relative to the current optimal value, the point with the largest expectation improvement is selected as the next observation point. In cases of high noise level, the current optimal value may be seriously disturbed, thus affecting the point selection strategy of EI, making it tend to over-explore. Smilarly, noise level also affects our final output results, as the minimum observation in this case is also not accurate. An intuitive solution to these problems is to replace the observed optimum with the optimum of the predicted mean derived by the agent model. Here the predicted mean is the agent model's extrapolation of the objective function based on the available information. In noisy situations, these predictions are often more reliable than direct observations. The main purpose of the research in this paper is to explore the circumstances under which we should trust predicted values more than observed values. To the best of our knowledge, this does not seem to have been discussed in detail in current articles, so this paper attempts to explore it through numerical experiments. In order to make the experimental conclusions more representative, the test problemss in this paper choose to use the advanced test benchmark environment "BBOB", inclusive of 24 noise-free test functions. We modified it to be in a noisy form as a benchmark question for the numerical experiments. The detail of BBOB functions can be found in [Hansen et al., 2010]

The structure of this article is segmented as follows: 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 may 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 throughout the entire design space as predicted by the surrogate model. The following experimental segment will focus on examining and contrasting the optimization efficacy of these 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 expressed as follows:

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

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

Note here we treat the objective function $f(x)$ as a random variable, and equation(2) represents the prior distribution of the variable $f(x)$. where $\mathbf{m}$ represents the prior mean function, and $K_{i,j} := k(x_i, x_j)$ is the covariance matrix. After obtaining the observation data $\mathcal{D}_n = (x_i, y_i)_{i=1}^n$, $f(x)$ given the observed data $\mathcal{D}_n$, 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 $\mathbf{x}_{1:n}$. By employing 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 a variety of acquisition fuction options, 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]. This paper concentrates 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, θ 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 Φ and ϕ 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 inaccurate. In situations of high noise level, the value of f_n^* may be more extreme, causing EI to mistakenly believe that the most 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 among the observed points to 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:

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

μ^+ is the current optimal value, $\mu_n(\mathbf{x}), \sigma_n(\mathbf{x})$ are the predicted mean and predicted variance for the input value $\mathbf{x}$. 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 functions in this article are selected from the 24 test functions of the BBOB noiseless test function group, and noise is added as our test benchmark problems. These functions are designated as F1-F24. In the original BBOB text's configurations, these functions are categorized 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). Every function within the BBOB test function group is capable of acquiring novel test functions via alterations like shifting or rotating. 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:** In the experiment, the dimensions of the function are segmented into 2 and 4 dimensions. For 2-dimensional problems, the total budget for a single experiment is 100 observation points under noise-free conditions, and escalates to 300 points in noisy conditions. 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:

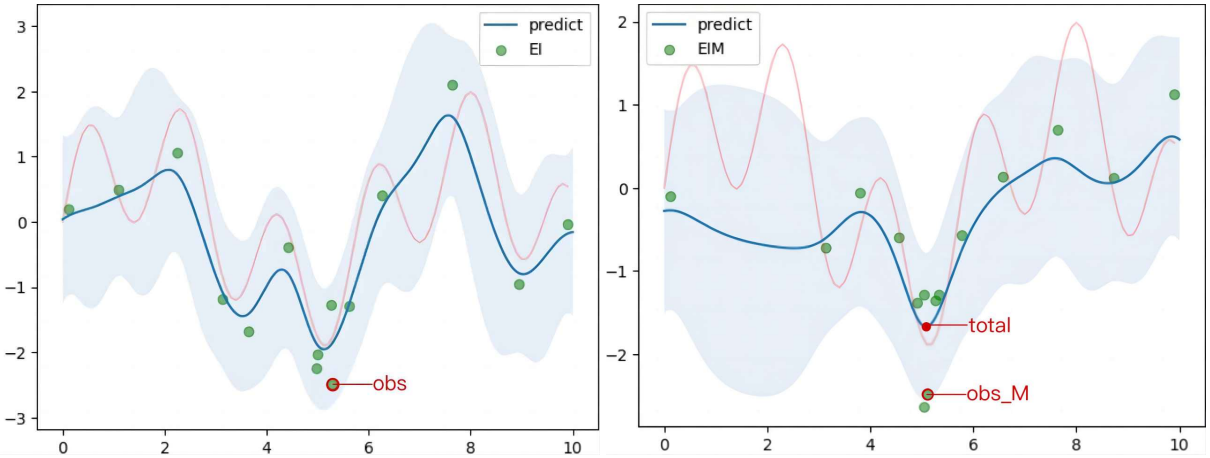
$$loss = \frac{y_a - y_{opt}}{y_{opt}} \times 100\% \tag{9}$$

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 the experimental results for the noiseless problem. In the noise-free problem, the methods **obs**,**obs_M** are exactly the same since the observations of the observation points are all accurate. We only need to compare the **obs** and **total** methods and analyze whether the predictive output method can output better points as the final result. Of course, it makes sense to consider **obs** as the most reasonable output method under noise-free conditions. The main purpose of our noise-free experiments is to investigate the specific gap between the optimization efficiency of these two output methods under noise-free conditions. Table 2 details the specific differences between these two output methods:

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**,**obs_M** relative to the global optimal value of each test function. The third columns is the relative loss rates of these two output methods. For two-dimensional problems, the predictive method **total** is significantly less efficient in optimization than **obs** when applied to the functions F2, F10, F11, and F12. A review of Table 1 reveals a common feature of these functions: their conditioning levels are significantly high, exceeding 10^6. The standard deviation of each of these functions is very large, resulting in high noise levels. The high noise level can easily mask the original structural characteristics of the objective function, making it difficult to build a suitable model for the Gaussian process to predict. By examining Table 2, it can be seen that the loss rate of **obs** is likewise relatively high on these four functions compared to the others, which further suggests that the surrogate model struggles due to high conditioning levels. Therefore, it can be

concluded that for noiseless problems, low accuracy in surrogate model modeling can greatly increase the loss rate of predictive output methods, in which case observational output method **obs** is more plausible. For simpler functions, the results in the third column show that the loss ratio of **total** relative to **obs** is mostly above 10%. For more complex functions such as F10, F12, etc., the gap between the optimization results will increase further. Therefore, for noise-free problems, we should indeed directly use the observation-based Bayesian optimization result output method (The loss rate of **total** 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 these two output methods begins to decrease on many functions.This will be discussed later in the article.

4.2 Experimental results for the 2D noise problem

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 accurate.(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.

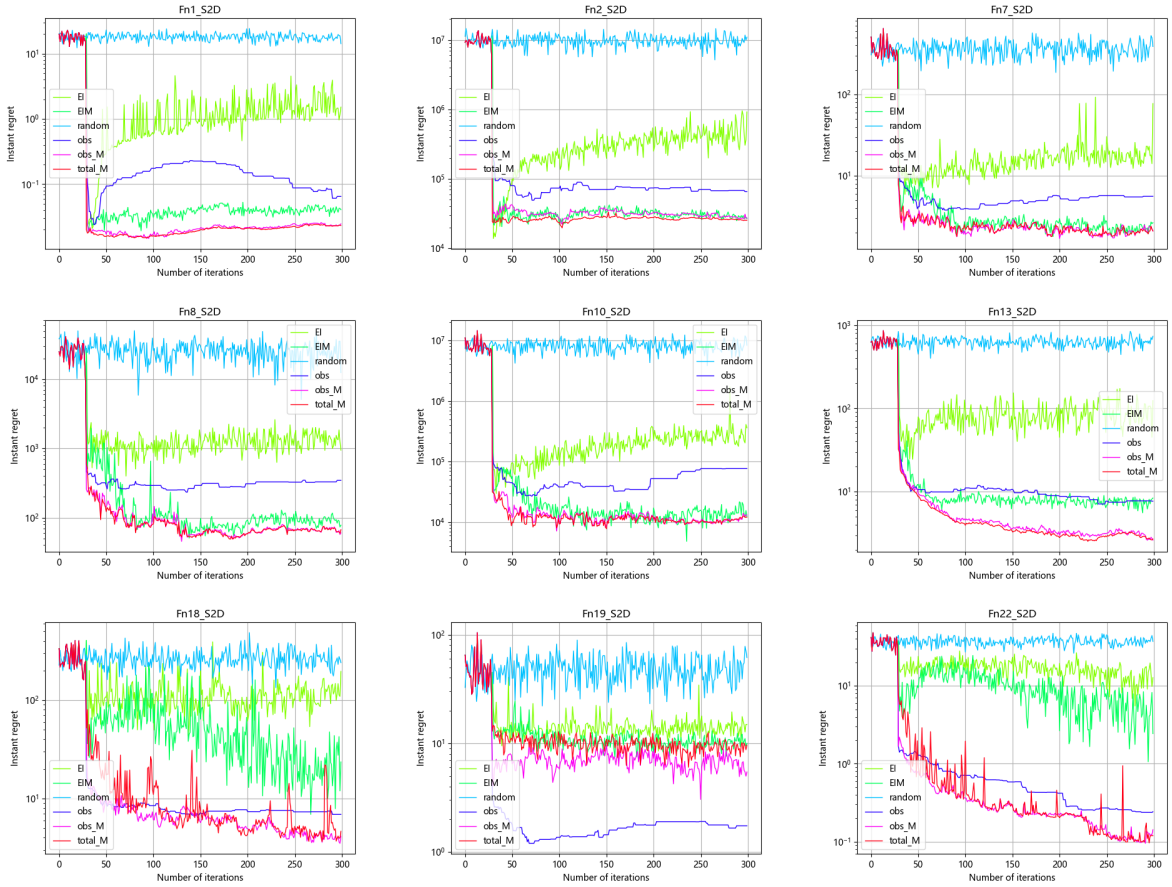


Figure 2: 2D optimization results(small noise)

It should be noted that the points found by the three acquisition functions(EI, EIM, random) described above at each iteration are used as the observation points of BO at the corresponding rounds, while the points found by the three output methods(**obs**, **obs_M**, **total**) at each iteration are used as the output results of BO at the corresponding rounds. The meaning of these two is essentially different. We draw the curves of the three acquisition functions mainly to observe the difference of the point selection strategies of the three acquisition functions in different optimization problems; while the real interest of this paper is the difference of the three output methods, and the curves of the three acquisition functions are not the main research object.

In the two-dimensional small noise optimization result figures, 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.

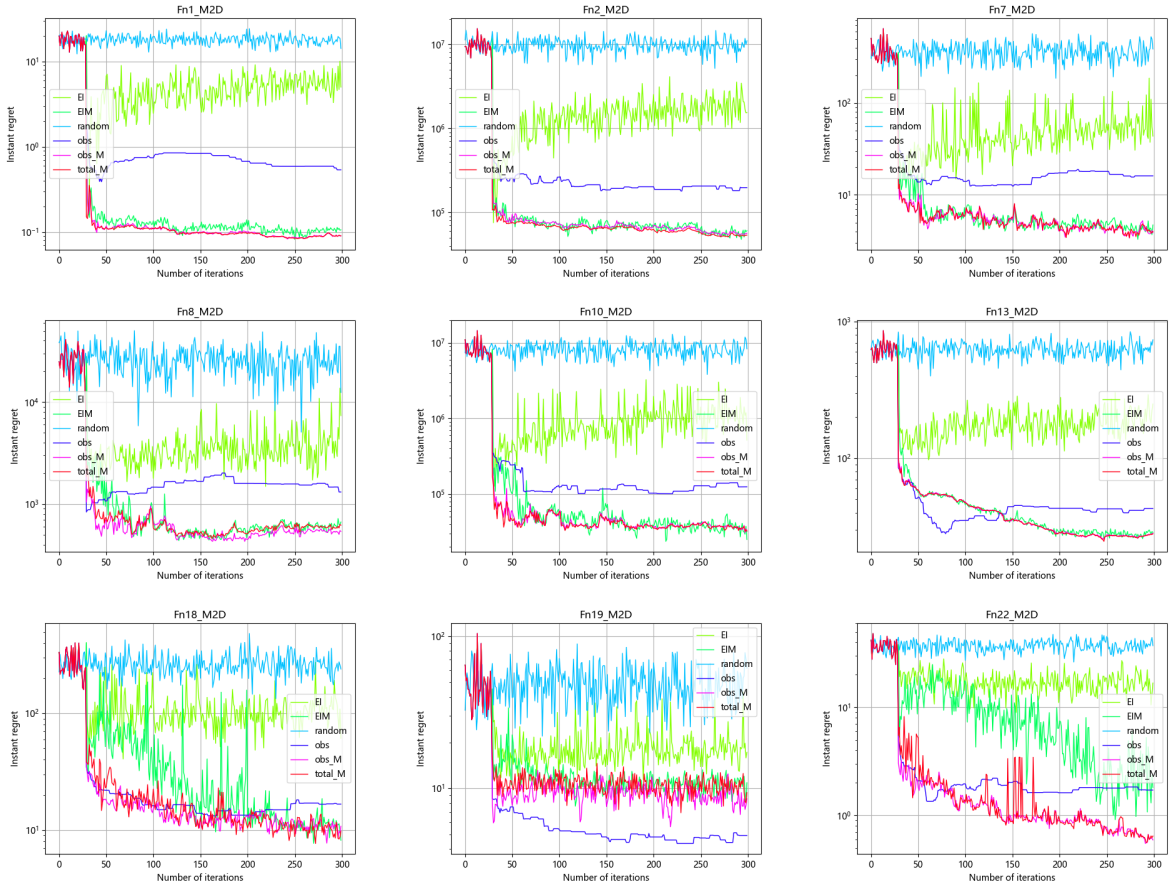


Figure 3: 2D optimization results(Medium noise)

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 optimization of the test function Fn19 , we find that obs has a clear advantage over other output methods. Further observation of the image corresponding to Fn19 reveals that the curves of EI, EIM, **obs_M**, and **total** are basically clustered at the same height, which actually indicates that the GP model models the function 19 poorly, the exploitation ability of the acquisition function is weak, and the prediction accuracy of the model is low. In this case, the results of the predictive output method are unreliable. Therefore, predictive output should be used with caution for highly complex problems with low modeling accuracy like Function 19. For general noise problems with good modeling accuracy, it is more reasonable to use predictive output methods (**obs_M**).

Figure3 illustrates the optimization results for moderate noise levels. It can be seen that with increased noise, the gap between the observational output method **obs** and the predictive output method widens further for most of the test functions. Overall, the conclusions for the choice of output methods are generally consistent with those under small noise.That is, for most problems with small noise, the use of the **obs_M** output method is the best choice. The use of **obs** only needs to be considered if one is convinced that BO's surrogate has low predict accuracy, but in fact replacing the model may be the better choice in such cases. If we increase the noise level further (see Appendix B for details), the above conclusions still hold true, and when noise level is too high, the advantage of **obs** in Fn19 even no longer exists.

4.3 Experimental results for the 4D noise problem

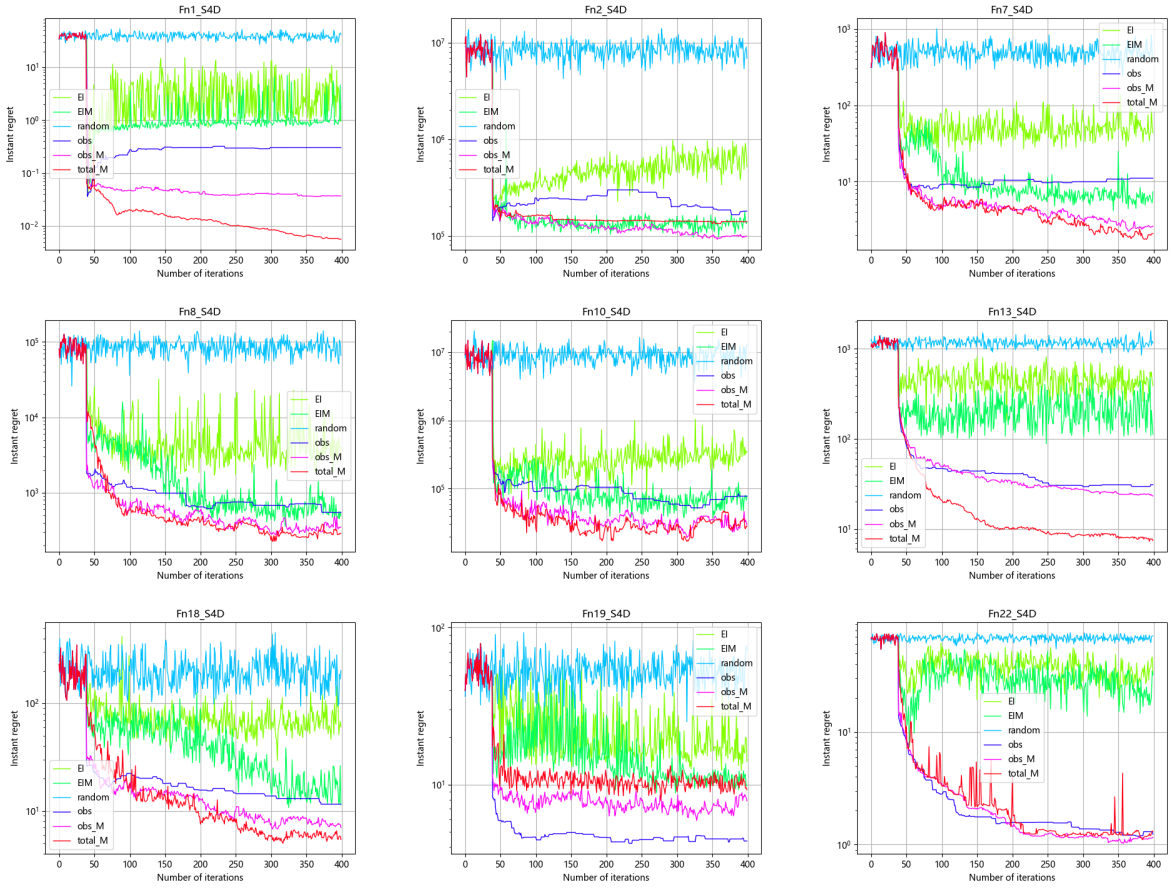


Figure 4: 4D optimization results(small noise)

In this section we increase the dimensionality of the optimisation problem from 2 to 4, which will allow the search space to expand geometrically. Figure 4 shows the optimisation results on a low noise level problem. Compared to the 2-dimensional image, we can clearly notice that the output method **total** starts to become competitive. For example, on the function Fn1, which is the simplest form and the least difficult to optimise, textbftotal outputs far better results than the other output methods. The same is true on function Fn13. There is also a small advantage over **obs_M** on functions Fn7 and Fn18. This is mainly due to the fact that **obs_M** can only select among the points that have been

observed, and even though we have increased the maximum number of iterations to 400 for the four-dimensional problem, 400 points is still too few for a four-dimensional search space. And, as we discussed earlier, both for EI and EIM, these acquisition functions selected points are not used exclusively for exploitation; they are also required to undertake certain exploration tasks. This leads to a further reduction in the proportion of these 400 points that contain higher points. In the case of **total**, on the other hand, the 400 points it selects are fully focused on exploitation and are not limited to observed points. This makes it more likely to find local optimal value points. Whereas for higher dimensional problems, too large a search space can make it extremely difficult to find the global optimal solution, in which case finding the local optimal solution may instead be a better choice. However, it is worth noting that raising the dimensionality does not solve the problem of the instability of **total** output results. For example, on Fn22 **total** still has large fluctuations, which means that it is still possible that it may give an extremely bad point as an optimisation result for output in a particular experiment. There is no denying that **total** is competitive in slightly higher dimensional optimisation problems, but choosing it still carries the risk of instability. Therefore, we take a conservative view on the use of **total**, i.e., it should be used with caution. In contrast, the performance of **obs_M** remains relatively stable. It outperforms **obs** on all test problems except Fn19, and is not significantly weaker or even better than **total** on most problems. Taken together, **obs_M** continues to be the most reasonable choice in 4 dimensions with low noise levels. We similarly tried to continue to increase the noise level (see Appendix B for details) and the conclusions obtained remains the same.

5 Conclusion

In the previous section we conducted numerical experiments on optimisation test problems without noise and with different noise levels and dimensions. The main conclusions are summarised as follows:

- For the noiseless problem, directly using the observation-based output **obs** is the most reasonable choice.
- For the noisy problem, using the predictive-based output is superior to the observation-based output in most of the cases, even when the noise level is low.Considering that the output results of **total** are not stable, the use of **obs_M** is the most reasonable choice.
- For more complex problems, when the prediction accuracy of BO's surrogate model is low, the predictive-based output method should be used with caution. This is because the model's prediction of the objective function is less credible. In this case, the choice of observation-based output will often give better output results, but in practice it is meaningless. If we can be sure that the model is less predictive, then we should consider replacing the model rather than discussing what output to use.
- When the optimisation problem is of high dimensionality, the use of **total** can be considered, but it still carries the risk of unstable output results.

It is also worth mentioning that the form of the noise can likewise influence the choice of output methods. Our experiments above only discuss homogeneous noise, and the conclusions may change when the noise becomes non-homogeneous. We briefly discuss this scenario in Appendix A and give the corresponding experimental results.

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:

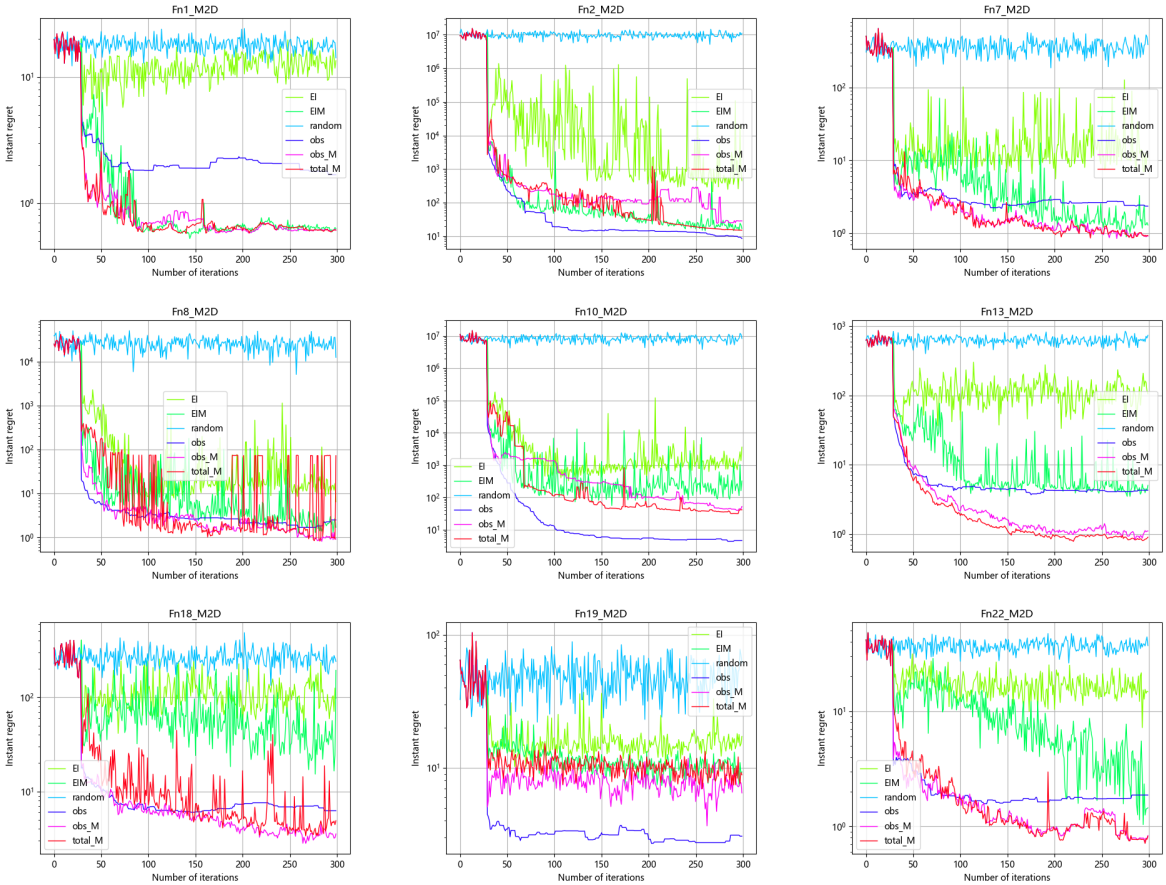


Figure 5: 2D optimization results(Non-homogeneous noise)

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. Fn2, 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:

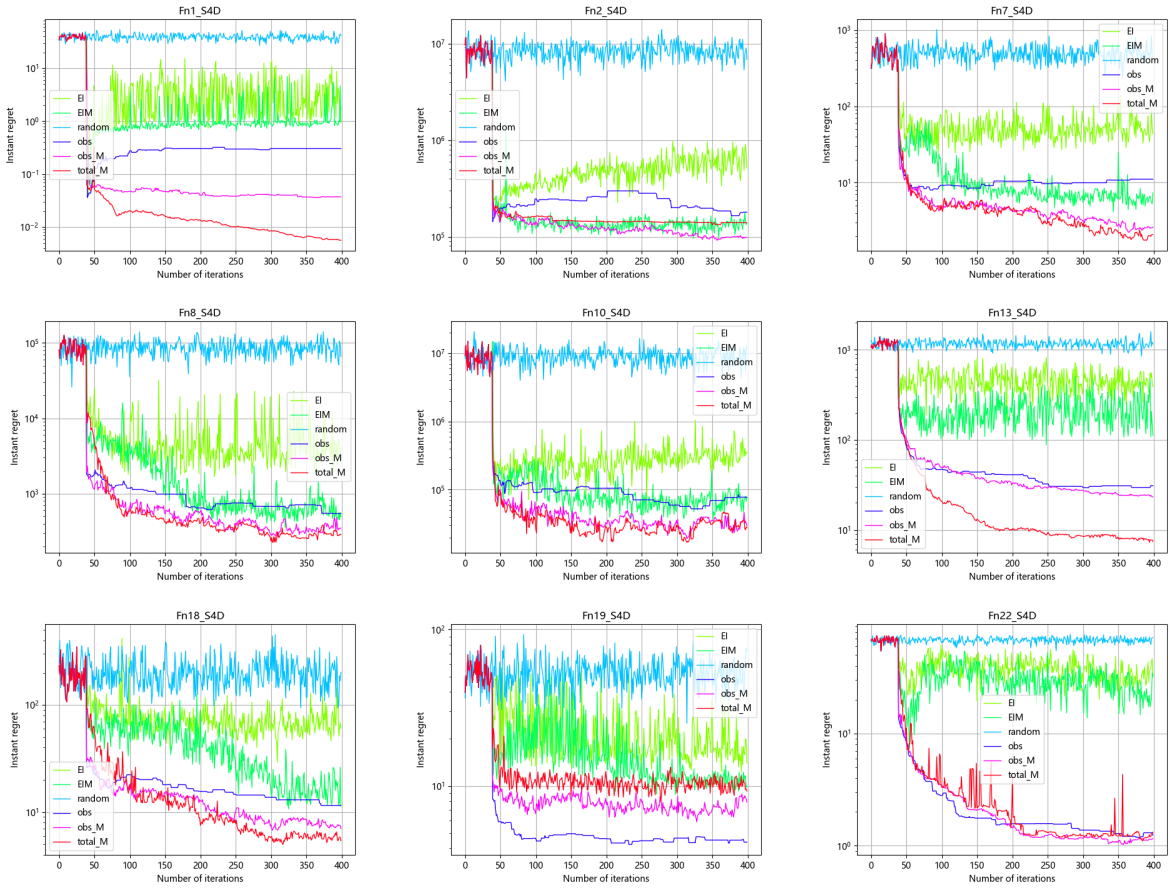


Figure 6: 4D optimization results(homogeneous noise)

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.

B Other homogeneous noise experimental results

B.1 2-dimensional experiments

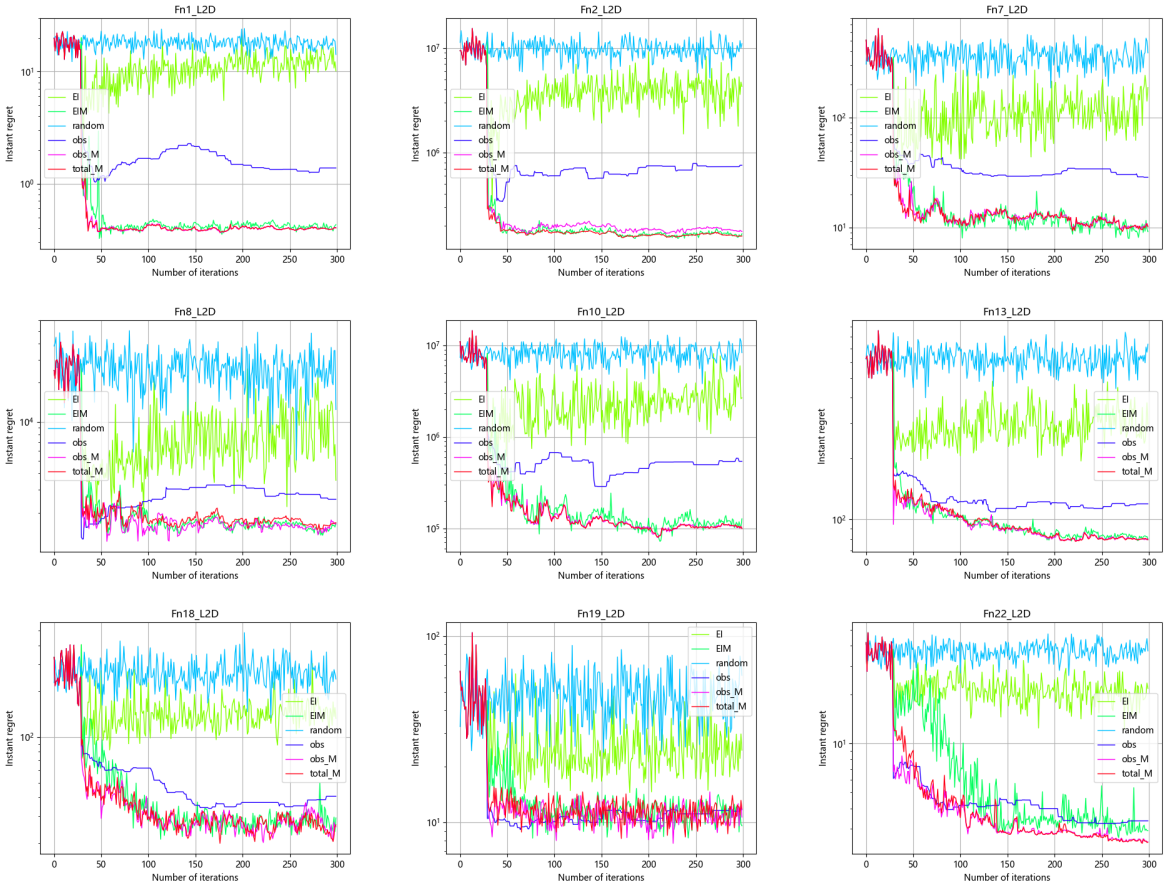


Figure 7: 2D optimization results(Large noise)

B.2 4-dimensional experiments

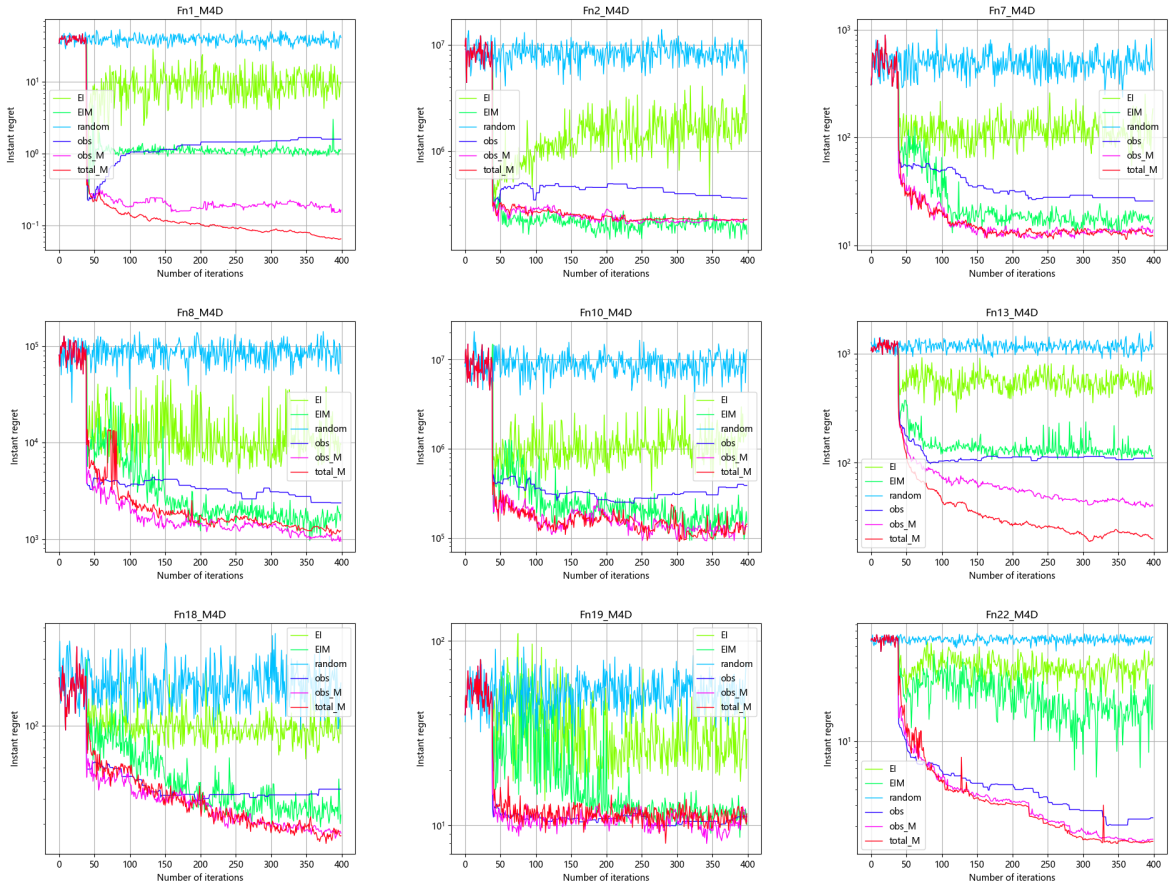


Figure 8: 4D optimization results(Medium noise)