Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

MODIFICATION OF BAYESIAN OPTIMIZATION FOR EFFICIENT CALIBRATION OF SIMULATION MODELS

Daiki Kiribuchi
Masashi Tomita
Takeichiro Nishikawa

Research & Development Center
Toshiba Corporation
1, Komukai-Toshiba-cho
Saiwai-ku, Kawasaki-shi
Kanagawa 212-8582, JAPAN

Satoru Yokota
Ryota Narasaki
Soh Koike

Institute of Memory Technology Research & Development
Kioxia Corporation
580-1, Horikawa-cho
Saiwai-ku, Kawasaki-shi
Kanagawa 212-8520, JAPAN

ABSTRACT

Simulation models contain many parameters that must be adjusted (calibrated) in advance to reduce the error between simulations and experimental results. Bayesian optimization is often applied to minimize error after only a few simulations. However, Bayesian optimization uses only error information, ignoring information on other simulation results. In this paper, we improve Bayesian optimization by utilizing both and show that other simulation results effectively reduce the dimensionality of the parameter space. In an evaluation using actual semiconductor simulation results, the proposed method reduces the number of simulations by 50% compared with random search and conventional Bayesian optimization.

1 INTRODUCTION

Simulators are created and used in various fields to establish improvement guidelines (Xu et al. 2015). For example, simulations of hospital stay lengths (Zeng et al. 2012), container terminal processes (Bruzzone et al. 2012), factory operation scheduling (Hsieh et al. 2007), and semiconductor products (Kukita et al. 2018) have been proposed in studies examining improvement guidelines.

Simulation models contain many parameters. To simulate an actual experiment well, we must adjust these parameters in advance so that simulation results are compatible with experimental results. In particular, calibration is required to reduce the error between simulations and experiments.

Many parameters and long simulation calculation times make reducing error time-consuming. Reducing the time required for calibration would thus be effective for obtaining improvement guidelines. We regard this error minimization problem as a black-box optimization. Bayesian optimization is often applied when evaluating an objective (black-box) function takes time, as the case in Shahriari et al. (2016). Bayesian optimization assumes a Gaussian process as a prior for a black-box function. In this paper, we propose a method for improving calibration efficiency by modifying the conventional Bayesian optimization.

Our problem is partially different from conventional Bayesian optimization, which uses only information on the error to be minimized. In contrast, we can also obtain information on simulation results. For example, when running the simulation of hospital stay lengths (Zeng et al. 2012), we can obtain simulation results such as patient stay lengths and wait times, along with the error between the experimental and simulation results. This additional information can be applied to improve efficiency.

Furthermore, we must adjust many parameters to reduce the error in this setting. This requires scaling Bayesian optimization to high-dimensional parameter spaces. Bayesian optimization is generally most

successful when input dimensions are low and does not provide good results in higher dimensions (Shahriari et al. 2016). Various methods for improving conventional Bayesian optimization have been proposed (Chen et al. 2012; Kandasamy et al. 2015; Wang et al. 2016; Li et al. 2017), but these methods also use only information about the objective function.

We propose a method for efficiently solving high-dimensional problems that predicts additional simulation results by a regression method and predicts the objective function value using prediction values for simulation results as explanatory variables. This method can utilize information on both the error and other simulation results.

In conventional Bayesian optimization, error is estimated from parameters by Gaussian process regression. In the proposed method, we first predict simulation results from parameters by the regression model and then estimate error by Gaussian process regression from the predicted simulation results. If there are fewer dimensions of the predicted simulation results than there are parameters, the proposed method can reduce the input dimensions for Gaussian process regression compared with conventional Bayesian optimization. This dimensionality reduction may reduce error in only a few simulations.

To our knowledge, two Bayesian optimization methods using additional information have been proposed (Astudillo et al. 2019; Uhrenholt et al. 2019), but these methods must predict all additional information to calculate error by Gaussian process regression. When the dimension of this additional information is large, the calculation of Gaussian process regression for all dimensions is impractically time-consuming. Further, these methods do not scale to optimization problems with high-dimensional parameters. We propose a method that does not need to predict all additional information, and can predict additional information with a regression method having shorter calculation times than does Gaussian process regression, making it more practical. Even with high-dimensional parameters, dimensionality reduction by our method efficiently reduces error.

To confirm the efficiency of the proposed method, we performed evaluation experiments using actual simulation results. As described below, the results confirmed that the proposed method reduces the number of simulations required for calibration compared with random search and conventional Bayesian optimization.

2 PROBLEM SETTING

In this section, we generalize the problem setting described in the previous section to present it as a mathematical problem. Here, we consider minimizing an objective function $f(x)$ on a set $\mathcal{X} \subset \mathbb{R}^D$ as

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

where $f(x)$ is the true objective function as obtained from simulations (or experiments) with parameters $x \in \mathcal{X}$. We can observe a noise-added objective function for the parameters x as

$$y = f(x) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma^2),$$

where $\mathcal{N}(0, \sigma^2)$ is a normal distribution with mean 0 and variance σ^2. The goal of this problem is to find x such that the objective function $f(x)$ is minimized when observing y for parameters x only N times. That is, assuming that x_N^* is the estimated optimal solution x^* after N times, we successively determine $x_1, \cdots, x_N$ so that a simple regret $r_N = f(x_N^*) - f(x^*)$ becomes smaller.

The above is the conventional problem setting for Bayesian optimization. Our proposed method differs in that we assume we can obtain both the objective function value y and other outputs $z \in \mathbb{R}^d$ as the simulation results for the parameters x as

$$z = g(x) + \delta, \delta \sim \mathcal{N}(0, \Sigma^2),$$

where g is $\mathbb{R}^D \to \mathbb{R}^d$. In calibration for the hospital simulation, for example, the objective function value y is the error to be minimized, and outputs z are patient stay periods, waiting times, and the like. Our setting requires no additional cost to simulate outputs z, because z must be simulated to compute the objective function.

We also assume that the objective function $f(x)$ can be decomposed into or approximated by $h(g(x))$, and that h is $\mathbb{R}^d \to \mathbb{R}$. This reduces the dimensionality of the parameter space if d (the dimension of $g(x)$) is smaller than D (the dimension of x). For example, we can consider $h(z)$ as an element sum of vector z like the additive structure (Kandasamy et al. 2015), as

$$h(z) = \sum_{i=1}^{d} z_i = \sum_{i=1}^{d} g_i(x),$$

where z_i is the i-th element value of vector z, and g_i is the i-th output function from the multi-output function g. We can define any other function instead of this, such as an element product of z and the norm of z. In the above problem setting, we propose a method that sequentially determines $x_1, \cdots, x_N$ so as to minimize the simple regret.

3 RELATED WORK

In this section, we briefly describe Bayesian optimization (Shahriari et al. 2016), which is the basis of the proposed method, and random embedding Bayesian optimization (REMBO) (Wang et al. 2016), which is a related method.

Algorithm 1 shows the Bayesian optimization algorithm. Bayesian optimization assumes a Gaussian process as the prior for the objective function. The Bayesian optimization algorithm first calculates the posterior of the objective function from data. Next, it calculates an acquisition function $\alpha_n(x)$ using the derived posterior. Although many acquisition functions have been proposed such as UCB (Shahriari et al. 2016), we use expected improvement (EI) in this paper. We calculate the parameters x_{n+1} maximizing the acquisition function. Finally, we observe the objective function value y_{n+1} for the maximum input parameters x_{n+1} and add these to the data. Bayesian optimization automatically repeats the above procedure.

Algorithm 1: Bayesian optimization.

1: for $n = 1, 2, \cdots$ do:
2: Calculate an acquisition function $\alpha_n(x)$ for inputs x based on data $\{x_i, y_i\}_{i=1}^{n}$.
3: Calculate parameters x_{n+1} maximizing the acquisition function

$$x_{n+1} = \underset{x \in X}{\operatorname{argmax}}\, \alpha_n(x).$$

4: Calculate the objective function value y_{n+1} for input parameters x_{n+1}.
5: end for.

As described above, we cannot obtain good results by conventional Bayesian optimization with high-dimensional parameters, but various methods for improving Bayesian optimization have been proposed. REMBO (Wang et al. 2016) is the method most relevant to this paper. REMBO reduces dimensions of the explanatory variables by linear transformation, and is the same as Bayesian optimization after the linear transformation. Algorithm 2 shows the REMBO algorithm. Note that z in Algorithm 2 is unrelated to the z in our setting. We randomly select matrix A to linearly transform the input parameters. Wang et al. (2016) demonstrated that REMBO is effective with higher dimensional parameters.

Algorithm 2: REMBO.

1: Generate a random matrix $A \in \mathbb{R}^{D\times d}$.
2: Determine a set $\mathcal{Z} \subset \mathbb{R}^d$.
3: for $n = 1, 2, \cdots$ do:
4: Calculate an acquisition function $\alpha_n(z)$ for z based on data $\{z_i, y_i\}_{i=1}^n$.
5: Calculate point z_{n+1} maximizing the acquisition function

$$z_{n+1} = \underset{z\in\mathcal{Z}}{\operatorname{argmax}}\, \alpha_n(z).$$

6: Calculate the objective function value y_{n+1} for input parameters $x_{n+1} = Az_{n+1}$.
7: end for.

4 PROPOSED METHOD

For the above-described problem setting, this section describes a method by which we can obtain outputs other than the objective function. In our problem setting, it is necessary to reduce the input dimension as REMBO does because the dimension D of the parameters is large. In addition, we can obtain both the objective function value y and other outputs z. Because Bayesian optimization and REMBO use only the objective function value y, we may reduce the necessary number of simulations by also using other outputs z.

We propose a modified Bayesian optimization method that uses output z to reduce parameter dimensions. The proposed method predicts output z from parameter x by a regression method, then predicts the objective function value y using the prediction result $\hat{z}$ of the regression as explanatory variables. Next, it calculates an acquisition function $\alpha_n(\hat{z})$ using the derived posterior of the objective function. We use expected improvement (EI) in this paper. (See Shahriari et al. (2016) for an explicit representation of EI.) Algorithm 3 shows the proposed method.

Algorithm 3: Proposed method.

1: for $n = 1, 2, \cdots$ do:
2: Based on data $\{x_i, z_i\}_{i=1}^n$, calculate a regression model $\phi_n(x)$ of z for input parameters x.
3: Calculate the regression result $\hat{z}_i = \phi_n(x_i)$ for input data $\{x_i\}_{i=1}^n$.
4: Calculate an acquisition function $\alpha_n(\hat{z})$ for $\hat{z}$ based on data $\{\hat{z}_i, y_i\}_{i=1}^n$.
5: Calculate parameters x_{n+1} maximizing the acquisition function

$$x_{n+1} = \underset{x\in\mathcal{X}}{\operatorname{argmax}}\, \alpha_n\big(\phi_n(x)\big).$$

6: Calculate output z_{n+1} and objective function value y_{n+1} for input parameters x_{n+1}.
7: end for.

When calibrating a queuing model, for example, x is a control parameter, outputs z are waiting times, and the objective function value y is the error of the waiting times $\|z - z_{obs}\|$, where z_{obs} is the observed waiting times. The proposed method predicts z from x, then calculates the posterior of the error y from pairs of predicted waiting times $\hat{z}$. Next, it calculates an acquisition function $\alpha_n(\hat{z})$ using the derived posterior, then the control parameter x_{n+1} maximizing the acquisition function. Finally, we observe the waiting times z_{n+1} and the error y_{n+1} for the maximum input parameters x_{n+1} and add these to the data.

Bayesian optimization directly predicts the objective function value y from parameters x by Gaussian process regression. However, the proposed method predicts output z from parameters x, and predicts the

objective function value y by Gaussian process regression from a predicted value $\hat{z}$. The proposed method thus predicts y in two stages from x, using information about z. If $d < D$, we can regard the proposed method as a Bayesian optimization that reduces the parameter dimensions.

We can use methods such as linear regression, lasso regression, or random forest regression to predict z from input x, and we can use cross-validation to determine hyperparameters for each regression method every iteration. The applied regression method depends on the problem. Here, we compare the proposed method with various regression methods in numerical experiments, described below.

In Algorithm 3, the acquisition function $\alpha_n(\hat{z})$, which is a function of $\hat{z}$, is regarded as a function $\alpha_n(\phi_n(x))$ with input x, and is maximized with respect to x. We can also use a simple method that calculates the next input $x_{n+1} = \phi_n^{-1}(z_{n+1})$, where z_{n+1} is the maximum value of the acquisition function $\alpha_n(\hat{z})$ with respect to $\hat{z}$ (i.e., $z_{n+1} = \mathrm{argmax}_{\hat{z}}\, \alpha_n(\hat{z})$). However, compared with this simple method, Algorithm 3 does not need to calculate the inverse transformation of the regression. Algorithm 3 can therefore use regression methods in which computation of the inverse transformation is difficult. In this sense, Algorithm 3 is better than the simple method. If we can calculate the derivative of the function $\phi_n(x)$ (like linear regression and lasso regression), we maximize $\alpha_n(\phi_n(x))$ by multi-start L-BFGS. Otherwise, we apply an evolutionary strategy to calculate the maximum of $\alpha_n(\phi_n(x))$. Such optimizers are common in Bayesian optimization (Shahriari et al. 2016).

Consider the case where linear regression is used as the regression model for z in the proposed method. If $A_n \in \mathbb{R}^{d\times D}$ is a coefficient of the linear regression, then the explicit expression of the regression model is $\phi_n(x) = A_n x$. The proposed method predicts the objective function value using the predicted value $\hat{z} = A_n x$ as an explanatory variable. This is very similar to the REMBO method, but the proposed method is considered better than REMBO in this problem setting in two ways.

First, REMBO uses random linear transformation, but the proposed method determines the linear transformation using information about the output z. This transformation using information is better than REMBO at predicting the objective function value y, and the proposed method effectively reduces the number of simulations required to obtain good parameters.

Second, while REMBO continues to use fixed transformations regardless of time, the proposed method continuously improves the transformation matrix each time data is added, thereby improving accuracy of the two-step prediction in the proposed method at each step.

5 NUMERICAL EXPERIMENTS

5.1 Numerical Experiments with an Actual Problem

We performed numerical experiments using a simulation of a semiconductor fabrication problem. We used four methods for calibrating a simulation model (Ichikawa et al. 2010) actually used in semiconductor process development, comparing performance between the proposed and other methods. Using existing simulation results, we compared the average number of selections to find the sample with smallest error.

We used a simulation that reproduces a shape after semiconductor processing. This simulation considers neutral diffusive transport, ion etching, and sputtered molecules on all surfaces. We used an in-house SiO_2 etching model (Ichikawa et al. 2010) for surface reactions. Figure 2 in Koike et al. (2019) shows an image of the shape obtained by this simulation. This calibration aims at making the shape in the simulation results as close as possible to an experimental shape after actual processing.

The simulation model contains many parameters having physical meanings, 26 of which must be calibrated ($D = 26$). Calibrated parameters include ion flux, neutral flux, etching yield, and reflection probability. We define the objective function y as the difference between the actual shape after processing and the shape obtained by simulation. We want to minimize this objective function. As the output z, we use eight shape lengths obtained by simulation ($d = 8$). These eight lengths characterize the shape.

Table 1: Average number of selections for each method.

Methods	Random	BO	REMBO	2norm BO	BO_Lasso
Actual problem	56.5±4.7	45.6±4.1	44.1±4.5	-	**23.0±3.2**
Actual problem (with smaller spaces)	57.8±4.7	55.5±4.5	61.6±4.7	-	**7.0±1.1**
Linear test problem	77.4±4.0	48.7±3.6	62.5±4.3	19.0±3.5	**0.83±0.04**
Nonlinear test problem	77.4±4.0	80.9±3.9	76.4±4.0	83.0±3.8	**63.1±3.9**

Calculation times for one simulation were about 2 h in this experiment. We evaluated the methods using 600 samples of simulation results with 600 parameters uniformly selected from the search space. In this experiment, we can regard observations of the objective function as having no noise ($\sigma^2 = 0$).

To imitate a realistic problem setting, 200 of 600 samples are first randomly selected and used as initial data. Next, each method sequentially selects 100 samples from among the remaining samples. This trial is repeated 100 times by each method.

We compared the number of samples selected until the optimal sample having the smallest objective function value among the 600 samples was selected. When calculating the average number of selections, the number of selections is set to 0 when the initial data include the optimal parameter, and to 100 when the sequentially selected 100 samples do not include the optimal sample. Note that the average index calculated by this procedure estimates a smaller number of selections when the method has poor performance, because 100 samples selected by a poorly performing method often do not include the optimum sample.

We first compared average indexes between the proposed and conventional methods. Table 1 shows the average number of selections under the proposed method using lasso regression (BO_Lasso) and three conventional methods using random search (Random), Bayesian optimization (BO), and REMBO as benchmarks. We cannot use the methods of (Astudillo et al. 2019; Uhrenholt et al. 2019) in an actual setting, because these methods would need to predict all of the thousands of possible shape lengths, which would take too much time.

Table 1 shows that the conventional methods, Bayesian optimization and REMBO, reduced the number of selections by only about 20% compared with random search, while the proposed method reduced the number of selections by about 60%. The proposed method is also more effective than the conventional methods, reducing the number of selections by about 50%. Under REMBO, the number of dimensions is set to $d = 8$ with the proposed method. There is a big difference in performance between REMBO and the proposed method, even using linear transformation with the same number of dimensions, suggesting that using the transformation matrix obtained from regression is better than using a random matrix.

Figure 1 shows the average minimum observed objective function value for each selection under each of the above methods. Comparing BO with REMBO in Figure 1, REMBO has smaller minimum values in the first half, while BO has higher performance in the second half. The proposed method always provides smallest minimum values and has extremely high performance.

To confirm which regression method should be used with the proposed method, we applied multiple regression methods to compare results of the proposed method. We also compared the proposed method with BO_Lasso(fix), which first reduces dimension by lasso regression, then applies conventional Bayesian optimization to the new variable, which has fewer dimensions. Table 2 shows the average number of selections for the proposed method using linear regression (BO_LR), lasso regression (BO_Lasso), random forest regression (BO_RF), BO_Lasso(fix), and REMBO.

Table 2 shows that the proposed method using lasso regression has the smallest average number of selections. The proposed method using lasso regression therefore has the highest performance in this problem setting. The proposed method using lasso regression is also more effective than the method that first reduces dimension, reducing the number of selections by 34%.

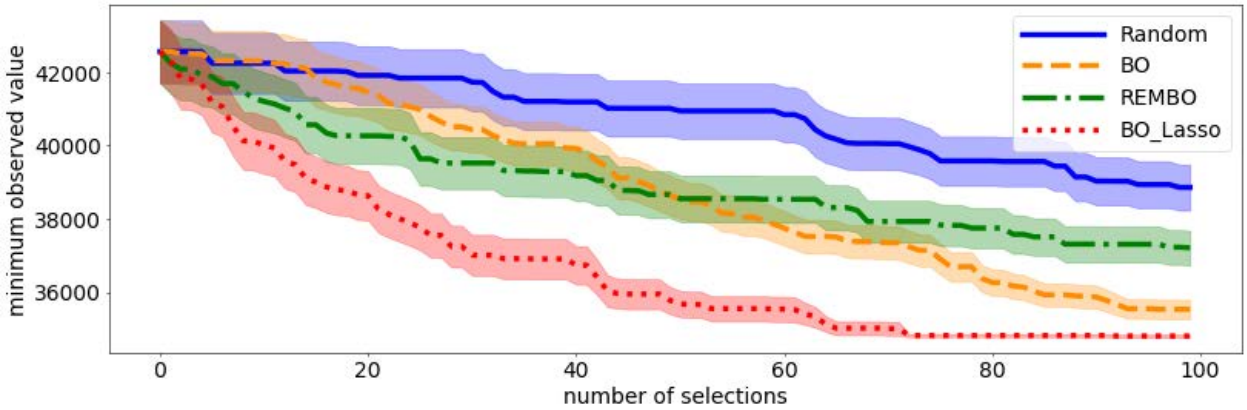

Figure 1: Average minimum observed values for each selection under each method. The horizontal axis shows the number of selections, and the vertical axis shows the average minimum values of the objective function observed up to the selection, with colored areas representing the standard deviation of the minimum observed objective function value for each method.

Table 2: Average number of selections for the proposed method using each regression method.

Methods	REMBO	BO_LR	BO_Lasso	BO_RF	BO_Lasso(fix)
Actual problem	44.1±4.5	28.1±3.5	**23.0±3.2**	36.1±4.0	34.6±4.2
Linear test problem	62.5±4.3	**0.83±0.04**	**0.83±0.04**	26.74±2.8	**0.83±0.04**
Nonlinear test problem	76.4±4.0	61.8±4.1	**60.2±3.9**	73.3±4.0	67.6±4.1

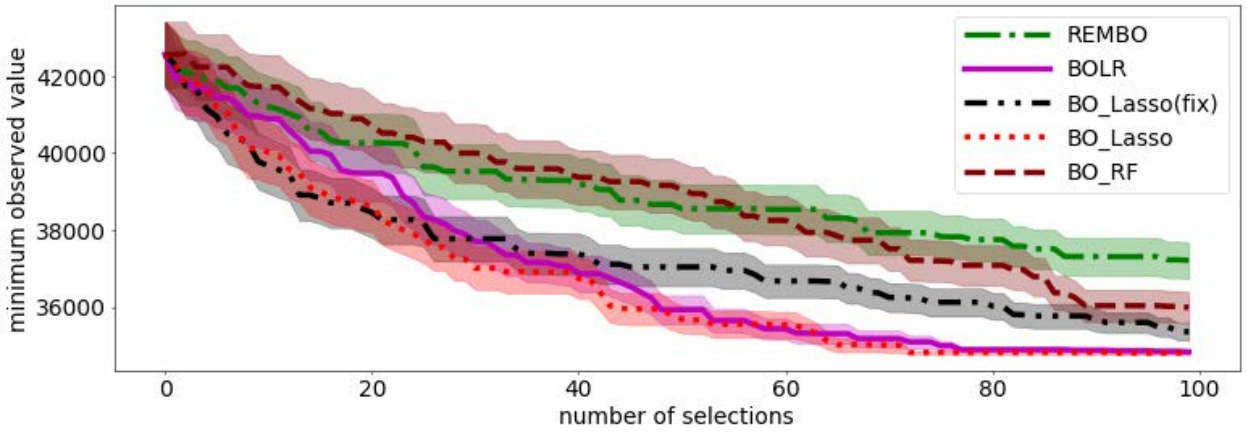

Figure 2: Average minimum observed values for each selection under the proposed method using each regression method.

Figure 2 shows the average minimum value of objective function observations for each selection with the proposed method using each of the above regression methods. Comparing the proposed method using random forest regression and REMBO, REMBO is better in the first half, but the proposed method has smaller minimum values in the second half. This suggests that predictions by random forest regression are inaccurate when the data is small.

Comparing the proposed method using linear regression and lasso regression, lasso regression has smaller minimum values, especially in the first half. This suggests that prediction accuracy by lasso regression is higher because of the sparsity assumption with few data.

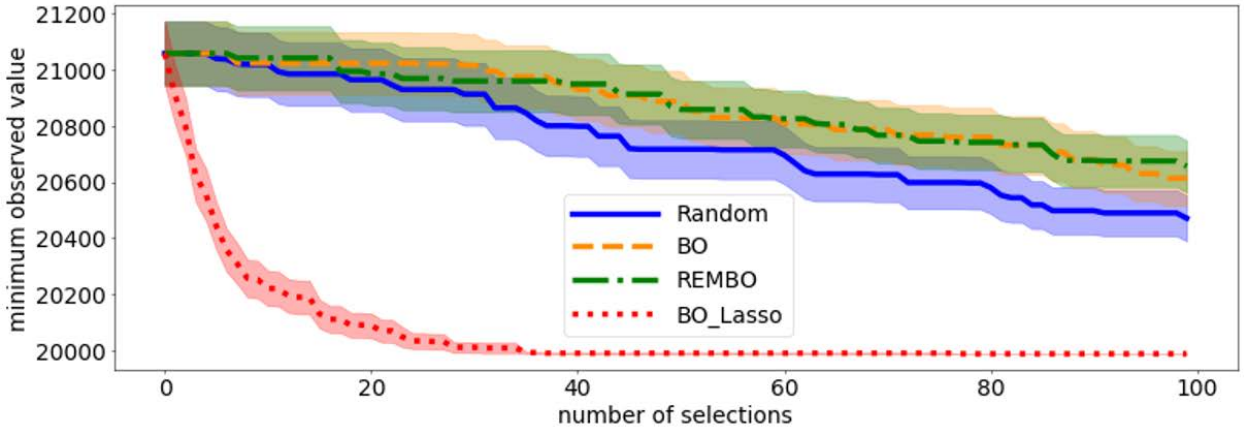

Figure 3: Average minimum observed values for each selection under each method when the search space is small.

Finally, we confirmed how the search space size affects efficiency of the proposed method. The experimental conditions are as described above, except that the search space size is small. We compared average indexes between the proposed and conventional methods when the search space is small. Table 1 shows the average number of selections under the proposed method (BO_Lasso) and three conventional methods (Random, BO, and REMBO). Figure 3 shows the average minimum observed objective function value for each selection under each of the above methods with the smaller search space.

The conventional methods, BO and REMBO, did not reduce the number of selections as compared with Random, while the proposed method reduced the number of selections by about 90%. From the higher prediction accuracy of lasso regression, the smaller the search space, the more the proposed method reduced the average number of simulations.

5.2 Numerical Experiments with Test Problems

To confirm whether the proposed method is effective for problems in other fields, we also compared the methods in linear and nonlinear test problems. As in the actual problem, the dimension D of parameters x is 26 and the dimension d of outputs z is 8. In the linear problem, we constructed a matrix $A \in \mathbb{R}^{d \times D}$ at random and calculated outputs z as $z = Ax$. In the nonlinear problem, we generated d functions g by sampling from a Gaussian process and calculated outputs z as $z = g(x)$. We created 1000 parameters at random and calculated 1000 outputs for each problem. We selected z^* from 1000 outputs and calculated the objective function y as $y = \|z - z^*\|_2^2$. To imitate a realistic problem setting, we first randomly selected 200 of 1000 samples as initial data. Next, each method sequentially selected 100 of the remaining samples. Repeating this trial 100 times in each method, we compared the average number of selections and found the sample with smallest error y.

We first compared average indexes between the proposed and conventional methods. Table 1 shows the average number of selections under the proposed method using lasso regression (BO_Lasso) and four conventional methods using Random, BO, REMBO, and 2norm BO (Uhrenholt et al. 2019) as benchmarks. Note that we did not compare our method with that of (Astudillo et al. 2019), because that method is essentially the same as that by (Uhrenholt et al. 2019) when the composite function is the norm of the parameters.

For the linear test problem, Table 1 shows that the conventional methods (BO, REMBO, and 2norm BO) reduced the number of selections by about 20–80% as compared with Random, while reduction by the proposed method was about 99%. The proposed method was also more effective than the conventional methods, reducing the number of selections by about 95%. In the nonlinear test problem, the conventional

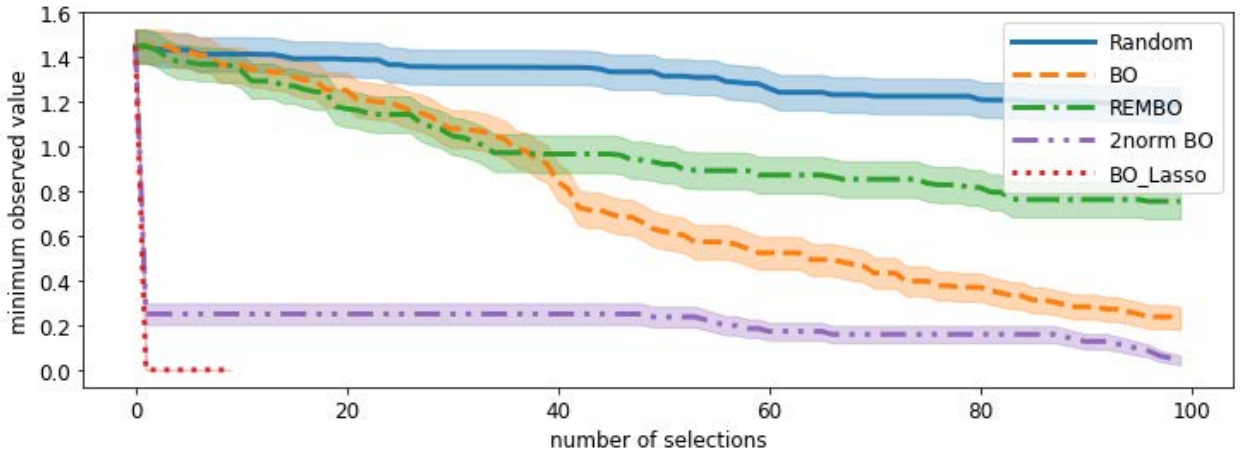

Figure 4: Average minimum observed values for each selection under each method in the linear test problem.

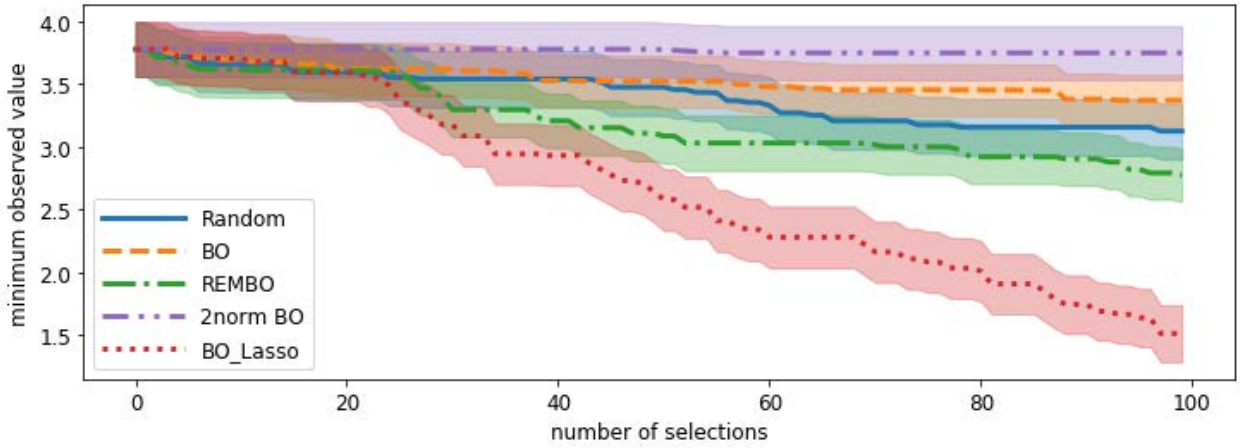

Figure 5: Average minimum observed values for each selection under each method in the nonlinear test problem.

methods hardly reduced the number of selections as compared with Random, while reduction by the proposed method was about 20%.

Figures 4 and 5 show the average minimum observed objective function value for each selection under each of the above methods in the linear and nonlinear test problems, respectively. For both the linear and nonlinear test problems, the proposed method always provided the smallest minimum values and demonstrated extremely high performance.

We also compared results of the proposed method using multiple regression methods and BO_Lasso(fix), which first reduces dimension by lasso regression, then applies conventional Bayesian optimization to the new variable with fewer dimensions.

Table 2 shows that again, the proposed method using lasso regression required the smallest average number of selections for both the linear and nonlinear test problems. In the nonlinear test problem, the proposed method using lasso regression is also more effective than the method that first reduces dimensions, reducing the number of selections by 34%. In both the actual and test problems, the proposed method using three forecasting methods is more effective than conventional Bayesian optimization, so the proposed method has robustness under different forecasting methods.

Numerical experiments showed that in our setting, it is always better to use the proposed method with dimension-reduced information, rather than the original information. However, it may be better to use the original information if the dimension of z is very low or if simulating z incurs additional costs.

6 CONCLUSION

We considered a new problem setting like the Bayesian optimization setting in which we can obtain both the objective function and other simulation outputs. Under this problem setting, we proposed a new method that predicts other outputs by the regression method, and predicts the objective function value using the prediction results of other outputs as explanatory variables. Numerical experiments confirmed that the proposed method is more effective than are the conventional methods. In particular, we confirmed that the proposed method using lasso regression can reduce the number of simulations by about 50% compared with the conventional methods.

REFERENCES

Astudillo, R., and P. Frazier. 2019. "Bayesian Optimization of Composite Functions". In *Proceedings of the 36th International Conference on Machine Learning*, June 9th-15th, California, USA, 354-363.

Bruzzone, A., F. Longo, L. Nicoletti, E. Bottani, and R Montanari. 2012. "Simulation, Analysis and Optimization of Container Terminals Processes". *International Journal of Modeling, Simulation, and Scientific Computing*, 3(04), 1240006.

Chen, B., R. M. Castro, and A. Krause. 2012. "Joint Optimization and Variable Selection of High-dimensional Gaussian Processes". In *Proceedings of the 29th International Conference on Machine Learning*, June 26th-July 1st, Edinburgh, Scotland, 1379-1386.

Hsieh, B. W., C. H. Chen, and S. C. Chang. 2007. "Efficient Simulation-Based Composition of Scheduling Policies by Integrating Ordinal Optimization With Design of Experiment". *IEEE Transactions on Automation Science and Engineering*, 4(4): 553-568.

Ichikawa, T., D. Ichinose, K. Kawabata, and N. Tamaoki. 2010. "Topography Simulation of BiCS Memory Hole Etching Modeled by Elementary Experiments of SiO 2 and Si Etching". In *Proceedings of 2010 International Conference on Simulation of Semiconductor Processes and Devices*, September 6th-8th, Bologna, Italy, 45-48.

Kandasamy, K., J. Schneider, and B. Póczos. 2015. "High Dimensional Bayesian Optimisation and Bandits via Additive Models". In *Proceedings of the 32nd International Conference on Machine Learning*, July 6th-11th, Lille, France, 295-304.

Koike, S., T. Ichikawa, S. Yokota, D. Kiribuchi, and T. Nishikawa. 2019. "Bayesian Optimization and Topography Simulation for High Aspect Ratio Etching". In *Proceedings of 2019 International Symposium on Dry Process*, November 21st-22nd, Hiroshima, Japan, 55-56.

Kukita, K., T. Uechi, J. Shimokawa, M. Goto, Y. Yokota, S. Kawanaka, T. Tanamoto, H. Tanimoto, and S. Takagi. 2018. "Simulation of Planar Single-gate Si Tunnel FET with Average Subthreshold Swing of Less Than 60 mV/decade for 0.3 V Operation". *Japanese Journal of Applied Physics*, 57(4S), 04FD09.

Li, C., S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, and A. Shilton. 2017. "High Dimensional Bayesian Optimization Using Dropout". In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, August 19th-25th, Melbourne, Australia, 2096-2102.

Shahriari, B., K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. 2015. "Taking the Human Out of the Loop: A Review of Bayesian Optimization". *Proceedings of the IEEE*, 104(1): 148-175.

Uhrenholt, A. K., and B. S. Jensen. 2019. "Efficient Bayesian Optimization for Target Vector Estimation". In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, April 16th-18th, Okinawa, Japan, 2661-2670.

Wang, Z., F. Hutter, M. Zoghi, D. Matheson, and N. de Feitas. 2016. "Bayesian Optimization in a Billion Dimensions via Random Embeddings". *Journal of Artificial Intelligence Research*, 55: 361-387.

Xu, J., E. Huang, C. H. Chen, and L. H. Lee. 2015. "Simulation Optimization: A Review and Exploration in the New Era of Cloud Computing and Big Data". *Asia-Pacific Journal of Operational Research*, 32(03), 1550019.

Zeng, Z., X. Ma, Y. Hu, J. Li, and D. Bryant. 2012. "A Simulation Study to Improve Quality of Care in the Emergency Department of a Community Hospital". *Journal of emergency Nursing*, 38(4): 322-328.

AUTHOR BIOGRAPHIES

DAIKI KIRIBUCHI is a researcher at the Research & Development Center, Toshiba Corporation, Japan. He received a B.E. in Mathematical Engineering and Information Physics (2013) and an M.S. in Information Physics & Computing (2015) from the University of Tokyo. His research interests include optimization via simulation, reinforcement learning, multi-armed bandit problems, and machine learning. His email address is daiki1.kiribuchi@toshiba.co.jp.

MASASHI TOMITA is a researcher at the Research & Development Center, Toshiba Corporation, Japan. He received a B.E. degree and an M.E. degree from Tohoku University in 2017 and 2019 respectively. His research interests include simulation optimization, multi-objective optimization and machine learning. His e-mail address is masashi2.tomita@toshiba.co.jp.

TAKEICHIRO NISHIKAWA is a senior fellow at the Research & Development Center, Toshiba Corporation, Japan. He receieved B.S. degree, M.S. degree and Ph. D in physics at Osaka University. His research interest include machine learning and materials informatics. His email address is takeichiro.nishikawa@toshiba.co.jp.

SATORU YOKOTA is a researcher at the Institute of Memory Technology Research & Development, Kioxia Corporation, Japan. He received a B.S. and an M.S. in Electrical and Electronic Engineering from Tokyo University of Agriculture and Technology. in 2013 and 2015, respectively. His research interests include Bayesian optimization and machine learning. His email address is satoru4.yokota@kioxia.com.

RYOTA NARASAKI is at the Institute of Memory Technology Research & Development, Kioxia Corporation, Japan. He received a B.S. and an M.S. in Materials Engineering from the University of Tokyo in 2013 and 2015, respectively. His email address is ryota1.narasaki@kioxia.com.

SOH KOIKE is at the Institute of Memory Technology Research & Development, Kioxia Corporation, Japan. He received the B.S. in physics at Tokyo Institute of Technology, the M.S. in physics at Kanazawa University and the Ph.D in material engineering at Tsukuba University. His email address is soh.koike@kioxia.com.