# Multi-Objective Bayesian Optimization using Deep Gaussian Processes with Applications to Copper Smelting Optimization

Liwen Kang[1,2], Xuelei Wang[1*], Zhiheng Wu[1,2], and Ruihua Wang[1,2]

[1]Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

***Abstract*—Copper smelting is a complex industrial process that involves a lot of long procedures and inter-process connections. Moreover, there are non-stationary, noisy, and multi-objective challenges in copper smelting optimization. The traditional methods of process optimization rely on experience to adjust repeatedly, which is time-consuming and laborious, as well as difficult to find the optimal point. Bayesian optimization is an effective method to discover the optimal point of an expensive black-box function using few samples. In this paper, Bayesian optimization is introduced to solve the copper smelting optimization problem. The surrogate model is constructed based on noisy deep Gaussian processes to cope with the non-stationary process and observational noise of copper smelting. Then, the expected hypervolume improvement is used as the acquisition function, considering multiple objectives when selecting the new sampling point. We conduct experiments on standard test functions and a simulation model of copper flash smelting. The experimental results demonstrate that the proposed method performs well in terms of convergence and diversity.**

***Index Terms*—Bayesian optimization, copper smelting, deep Gaussian processes, multi-objective optimization**

## I. Introduction

Copper smelting is a typical complex industrial process, characterized by numerous long procedures and inter-process coupling [1]. There are several problems associated with copper smelting, such as low utilization of resources, high energy and material consumption, poor quality, and serious environmental pollution. It is important to note that the composition of raw materials, the equipment condition, and production process parameters will all have an impact on copper smelting. Therefore, optimizing the comprehensive production index is regarded as a challenge.

Copper smelting process involves complex physical and chemical reactions [2]–[4]. The relationship between the production index and the decision variables is nonlinear, uncertain, and complex. Therefore, it's hard to model the copper smelting process in an analytical form. In other words, it is a black box. Due to the lack of objective function and gradient, traditional optimization methods cannot be used. Since the input and output of black box contain valuable experience and knowledge, data-driven optimization is an effective way to optimize copper smelting process. However, copper smelting process is computationally expensive and time-consuming. Evolutionary algorithms [5]–[7] need to evaluate a large number of samples, which is infeasible.

Moreover, copper smelting optimization faces the following three challenges. First, copper smelting process is non-stationary. A sudden change in operating conditions can affect chemical reaction states, such as feeding [8]. Therefore, the objective function exhibits different characteristics in different regions. Second, the observational noise needs to be considered, which often appears in actual production processes. Last, copper smelting optimization involves multiple competing objective functions. For example, maximizing copper production while minimizing off-gas emissions [9]. As these two objectives have different units and are in conflict with each other, no single solution will satisfy both. Therefore, copper smelting optimization problem is non-stationary, noisy, and multi-objective.

Bayesian optimization provides an effective solution to the above expensive black-box optimization problem [10]–[12]. As a method of optimal decision-making, Bayesian optimization uses the Bayesian theory to guide the search for the optimal value of the objective function. In each iteration, the surrogate model and the acquisition function are constructed based on the previously observed data, in preparation for the next optimization. As a result, the global optimal point can be found with fewer iterations, greatly improving the search efficiency. Due to its ability to optimize expensive black-box models, Bayesian optimization is widely used in various fields. Similarly, it is suitable to address the high computational cost of copper smelting optimization.

In this paper, we propose a novel Bayesian optimization algorithm to solve the decision optimization problem of copper smelting process. Inspired by deep learning theory [13], we use deep Gaussian processes [14] as the surrogate model to learn the complex copper smelting process. Then, Gaussian noise is added to the surrogate model to simulate the noise inherent in the actual production process. Finally, we design an acquisition function to approximate the Pareto front and solve the multi-objective optimization problem of copper smelting. The main contributions of this paper are as follows.

1) In order to model the non-stationary process and observational noise, noisy deep Gaussian processes are

This work is supported by National Key Research and Development Program of China [Grant No.2019YFB1704704]

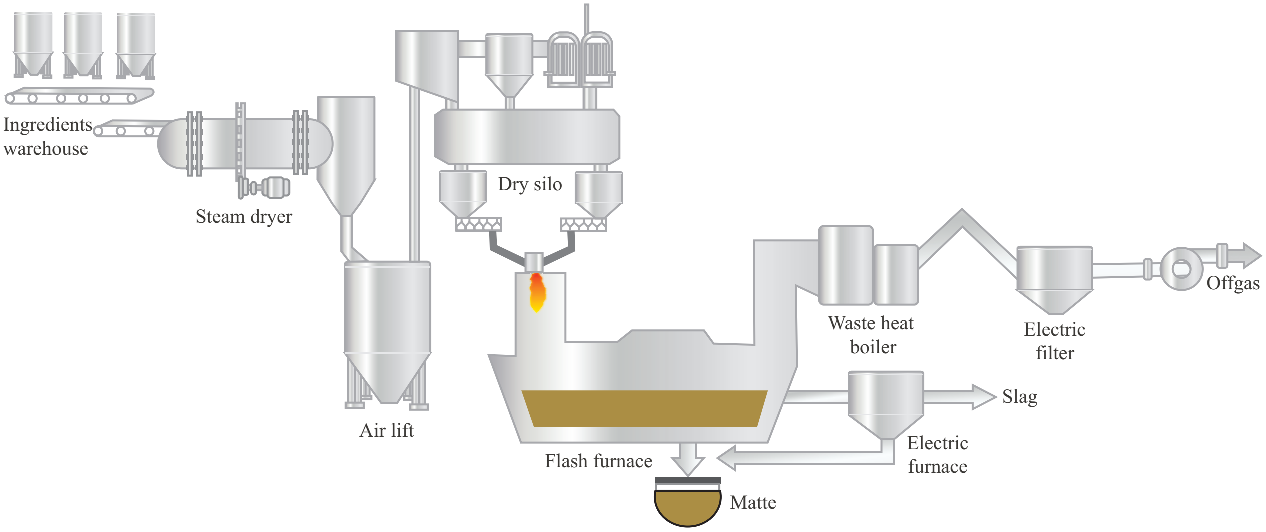


Fig. 1. The flowchart of copper flash smelting.

adopted, and the multi-layer Gaussian processes better fit the complex objective function. In the case of multiple competing objective functions, multiple noisy deep Gaussian processes are used to model multiple objectives separately. Then, we calculate the expected hypervolume improvement and find the Pareto solution.

2) The proposed Bayesian optimization framework is applied to solve the complex copper smelting optimization. Using historical data and few generated samples, the model is iteratively updated to quickly identify the parameter combination closest to the ideal.
3) The experimental results of the standard test functions and the simulation model of copper flash smelting show that the proposed algorithm is efficient and effective.

The rest of the paper is organized as follows. The background is discussed in Section II. Multi-objective Bayesian optimization using noisy deep Gaussian processes is described in Section III. Experiments of standard test functions and the flash smelting simulation model are provided in Section IV. Finally, the paper is concluded in Section V.

## II. Background

### A. Bayesian Optimization

Bayesian optimization is a sequential model-based optimization, which is suitable for solving problems with high computational complexity or long computational time [15]. It consists of two main parts, a surrogate model for modeling the objective function and an acquisition function for determining the next sampling location. A popular Bayesian optimization algorithm is Efficient Global Optimization (EGO) [16], which uses Gaussian process (GP) as the surrogate model. GP assumes that unknown variables are randomly drawn from a multivariate normal distribution. For a finite point set $X = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}]$, the objective function vector $f(X)$ follows the Gaussian distribution:

$$f(X) = [f(\boldsymbol{x}^{(1)}), \ldots, f(\boldsymbol{x}^{(n)})] \sim \mathcal{N}(\mu(X), K(X, X)), \quad (1)$$

where $\mu(\cdot)$ and $K(\cdot, \cdot)$ are the mean function and the covariance function of GP, respectively, and $K(\cdot, \cdot)$ depends on multiple hyperparameters $\Theta$.

GP is trained by maximum likelihood estimation to determine the optimal hyperparameters $\hat{\mu}$ and $\hat{\Theta}$, and then get $\hat{K}$. Given the existing observation $f(X) = \boldsymbol{y}$, the mean $\hat{f}(\boldsymbol{x})$ and the variance $\hat{s}(\boldsymbol{x})^2$ of the posterior distribution at a new point $\boldsymbol{x}$ are as follows:

$$\hat{f}(\boldsymbol{x}) = \hat{\mu} + \hat{K}(\boldsymbol{x}, X) \cdot \hat{K}^{-1}(X, X) \cdot (\boldsymbol{y} - \hat{\mu}), \quad (2)$$

$$\hat{s}(\boldsymbol{x})^2 = \hat{K}(\boldsymbol{x}, \boldsymbol{x}) - \hat{K}(\boldsymbol{x}, X) \cdot \hat{K}^{-1}(X, X) \cdot \hat{K}(X, \boldsymbol{x}). \quad (3)$$

After updating the surrogate model, the acquisition function will be designed. It is important for the acquisition function to balance exploration (regions with large variance) and exploitation (regions with small mean) to select the next most potential point. Commonly used acquisition functions include probability of improvement (PI) [17], expected improvement (EI) [18], and upper confidence bound (UCB) [19]. Based on PI, EI further considers the improvement amount $I(\boldsymbol{x}) = \max\{0, y_{\min} - f(\boldsymbol{x})\}$. EI is calculated as

$$\mathrm{EI}(\boldsymbol{x}) = (y_{\min} - \hat{f}(\boldsymbol{x}))\Phi(\frac{y_{\min} - \hat{f}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}) + \hat{s}(\boldsymbol{x})\phi(\frac{y_{\min} - \hat{f}(\boldsymbol{x})}{\hat{s}(\boldsymbol{x})}), \quad (4)$$

where $\Phi$ and $\phi$ represent the cumulative distribution function and probability density function of the standard normal distribution, respectively, and $y_{\min}$ is the minimum value in the previous iteration.

### B. Copper Flash Smelting

The flowchart of copper flash smelting is shown in Fig. 1, which includes steam dryer, flash furnace, electric furnace,

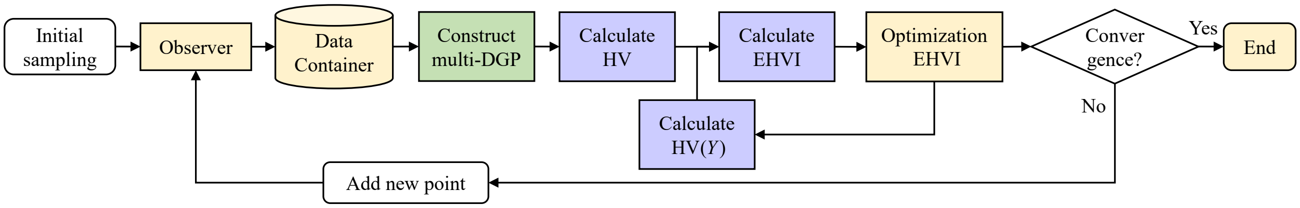


Fig. 2. The specific process of DGMBO. The green part represents the construction of the surrogate model using deep Gaussian processes, while the purple parts represent the design of the multi-objective acquisition function. During the iteration, the most promising sample is selected based on the current surrogate model coupled with the acquisition function, and then added to the data container until the convergence is reached.

waste heat boiler, and so on. Powdered copper concentrate is thoroughly dried in the steam dryer, mixed with air in the nozzle, then injected into the high temperature (1450-1550 °C) reaction shaft of the flash furnace at high speed (60-70 m/s) [20]. At this time, copper concentrate is in a suspended state, so the decomposition, oxidation, and melting processes of copper sulfide are generally completed in 2-3 seconds. The molten sulfides and oxides fall into the sedimentation tank of the flash furnace to collect and continue the reaction to form matte and slag, as well as to conduct precipitation separation. Slag is processed in the electric furnace and then discarded.

The goal of copper flash smelting optimization is to find a set of process parameters (e.g., raw material composition, production conditions, operating parameters, etc.) to optimize production indicators (e.g., matte output, off-gas output, etc.). Due to the complexity of copper flash smelting, it is primarily dependent on long-term experience and process knowledge to make optimal decisions in practice [21]. Since there are many variables that can affect the production process, this type of decision is arbitrary and inaccurate, and the optimization of production indicators cannot be guaranteed. Biswas *et al.* increased copper production by optimizing the chemistry of the smelting process, specifically to minimize slag production and copper entrainment in slag [5]. However, since this strategy is based on process chemistry, it is sometimes infeasible due to quality restrictions and other operational issues. Jiang *et al.* developed an optimal control model for the matte grade, and applied an improved genetic algorithm to find the optimal matte grade [6]. Franks *et al.* established a model with energy consumption as the objective function and proposed a global stochastic optimization genetic algorithm to solve the problem of unreasonable energy distribution [7]. The above two methods use evolutionary algorithms and require a large number of samples to be generated, which cannot get the optimal point instantly. For the first time, we apply Bayesian optimization to copper smelting optimization, and try to quickly get the optimal parameters of copper smelting through few samples.

## III. Multi-Objective Bayesian Optimization using Deep Gaussian Processes

In this paper, a multi-objective Bayesian optimization algorithm based on deep Gaussian processes (DGMBO) is proposed and applied to copper smelting optimization. The surrogate model and the acquisition function are specially designed according to the characteristics of the copper smelting process. First, for the case that the copper smelting process is non-stationary and noisy, noisy deep Gaussian processes are used as the surrogate model. Then, for the multi-objective situation in copper smelting, the expected hypervolume improvement is used as the acquisition function. In order to combine the multi-objective with the deep Gaussian processes model, multiple deep Gaussian processes are constructed, and the multi-objective problem is integrated into updating the surrogate model. The specific process of the algorithm is presented in Fig. 2.

### A. Noisy Deep Gaussian Processes

There is a non-stationary problem in the copper smelting process. Sudden changes in working conditions will cause chemical reaction states to change, so the objective function will exhibit different characteristics in different regions. However, the classical GP requires the covariance function to be stationary, i.e., $K(\boldsymbol{x}+\boldsymbol{c}, \boldsymbol{x}'+\boldsymbol{c}) = K(\boldsymbol{x}, \boldsymbol{x}')$. So we use deep Gaussian processes to build the surrogate model to learn complex functions for copper smelting. Deep Gaussian processes (DGP) are deep networks in which each layer is modeled by a Gaussian process. With the flexible deep structure, DGP is capable of constructing complex models for complex data, while being able to deal with non-stationary problems.

In addition, Gaussian noise is added to the last layer of DGP in order to simulate the noise associated with copper smelting. The structure of noisy DGP is shown in Fig. 3. For the $l$-th layer, the input and output are $H_{l-1}$ and $H_l$, respectively, and the multi-output GP is $\boldsymbol{f}_l(\cdot)$, satisfying $H_l = \boldsymbol{f}_l(H_{l-1})$. The $L$-th layer is the last layer, which is single-output with Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$, i.e., $y = f_L(H_{L-1}) + \epsilon$. Therefore, the input and output of noisy DGP are related as follows:

$$y = f_L(\boldsymbol{f}_{L-1}(\ldots(\boldsymbol{f}_2(\boldsymbol{f}_1(\boldsymbol{x}))))) + \epsilon. \tag{5}$$

In order to optimize the parameters of noisy DGP, the marginal likelihood of all samples needs to be calculated. Taking the non-observable variables of each layer into account,

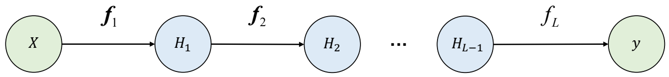


Fig. 3. The structure of noisy deep Gaussian processes. There are multi-output GPs in the hidden layers and a single output GP with Gaussian noise in the final layer.

the marginal likelihood is calculated as follows:

$$p(\boldsymbol{y}|X) = \int\limits_{H_1} \ldots \int\limits_{H_L} p\left(\boldsymbol{y}, H_1, \ldots, H_L|X\right) \mathrm{d}H_1 \ldots \mathrm{d}H_L$$
$$= \int\limits_{\{H_1,\ldots,H_L\}} p(\boldsymbol{y}|H_{L-1}) \left(\prod_{l=2}^{L-1} p(H_l|H_{l-1})\right) p(H_1|X)\mathrm{d}\{H_1,\ldots,H_L\}, \tag{6}$$

with the conditional probability being calculated using Gaussian distribution:

$$p(H_l|H_{l-1}) = \mathcal{N}(\mu(H_{l-1}), K(H_{l-1}, H_{l-1})). \tag{7}$$

However, due to the inversion operation of multiple covariance matrices in (6), the integration is difficult to calculate directly. We minimize the Kullback-Leibler (KL) divergence between the variational posterior and the true posterior to overcome this difficulty [22]. DGP is trained by maximizing the lower bound on the marginal likelihood:

$$\mathcal{L} = \sum_{i=1}^{n} \mathbb{E}_{q(y^{(i)})}[\log p(g^{(i)}|y^{(i)})] - \sum_{l=1}^{L} \mathrm{KL}[q(U_l)||p(U_l)], \tag{8}$$

where $g^{(i)}$ and $y^{(i)}$ are the actual observed value and the predicted value of the $i$-th sample. $Z_l = [\mathbf{z}_l^{(1)}, \cdots \mathbf{z}_l^{(d)}]^T$ is the set of inducing variables introduced in the $l$-th layer, and $U_l = \boldsymbol{f}_l(Z_l)$. Hence, the true posteriors of latent variables $H_l$ and $U_l$ are approximated by variational posteriors $q(H_l)$ and $q(U_l)$. The first and second terms of (8) can optimize the prediction and the posterior distribution, respectively. The hyperparameters to be optimized include the GP parameters $\Theta_l$, the inducing variables $Z_l$, and the variational distribution parameters $q(U_l)$, $l = 1, \ldots, L$. They are optimized by using adaptive moment estimation (Adam) [23] on the loss function in (8).

### B. Multi-Objective Acquisition Function

Copper smelting optimization involves multiple objective functions, aiming to seek a compromise solution between matte production and off-gas emission. It is necessary to design a multi-objective acquisition function for DGP, so that the joint optimization of multiple objectives is considered when selecting the next evaluation point. We construct two DGP models based on the two objectives of copper smelting. Through the DGP training method described previously, the parameters of two DGPs can be determined, and the posterior distribution of each point can be derived.

In order to extend Beyesian optimization to a multi-objective case, we calculate the expected improvement of the hypervolume indicator. The hypervolume indicator (HV) is defined as the $m$-dimensional Lebesgue measure $\lambda_m$ of the dominated subspace limited from above by the reference point $\boldsymbol{r}$, where $m$ is the number of objective functions [24]. Here, HV measures the size of the dominated subspace of two DGP objective functions, and it is defined as follows:

$$\mathrm{HV}(Y) = \lambda_m(\{\boldsymbol{z} \in \mathbb{R}^m \mid \exists \boldsymbol{y} \in Y : \boldsymbol{y} \prec \boldsymbol{z} \wedge \boldsymbol{z} \prec \boldsymbol{r}\}), \tag{9}$$

where $\prec$ represents the Pareto dominance order, $\boldsymbol{y} \prec \boldsymbol{z} \Leftrightarrow (\forall i \in \{1, \ldots, m\}, y_i \leq z_i)$ and $\boldsymbol{y} \neq \boldsymbol{z}$, and $Y$ represents the current Pareto-front approximation set.

For a new solution $\boldsymbol{y} \in \mathbb{R}^m$, the hypervolume improvement (HVI) of adding it to the above Pareto-front approximation set is as follows:

$$\mathrm{HVI}(Y, \boldsymbol{y}) = \mathrm{HV}(Y \cup \{\boldsymbol{y}\}) - \mathrm{HV}(Y). \tag{10}$$

Considering the prediction uncertainty (variance) of DGP, it is necessary to measure how much the expected hypervolume improvement (EHVI) can be achieved by a new solution. Assuming that there is no correlation between random variables from two DGPs. Given the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ of the predicted multivariate distribution, and the Pareto-front approximation set $Y$, the EHVI is defined as

$$\mathrm{EHVI}(\boldsymbol{\mu}, \boldsymbol{\sigma}, Y, \boldsymbol{y}) = \int\limits_{\mathbb{R}^m} \mathrm{HVI}(Y, \boldsymbol{y}) \cdot \xi_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{y})\mathrm{d}\boldsymbol{y}, \tag{11}$$

where $\xi$ is the probability density function of the multivariate independent normal distribution.

However, due to the need to calculate the volume of a non-rectangular polytope, the time complexity of EHVI is high, which seriously affects the Bayesian optimization efficiency. We implement the box decomposition algorithm [25] to accelerate the computation. Decompose the region into disjoint axis-aligned hyperrectangles, then calculate each volume and sum over all. Finally, using piecewise integration, EHVI is calculated as follows:

$$\begin{aligned}\mathrm{EHVI} = &\sum_{i=1}^{n+1} (y_1^{(i-1)} - y_1^{(i)}) \cdot \Phi(\frac{y_1^{(i)} - \mu_1}{\sigma_1}) \cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2) \\ &+ \sum_{i=1}^{n+1} (\Psi(y_1^{(i-1)}, y_1^{(i-1)}, \mu_1, \sigma_1) - \Psi(y_1^{(i-1)}, y_1^{(i)}, \mu_1, \sigma_1)) \\ &\cdot \Psi(y_2^{(i)}, y_2^{(i)}, \mu_2, \sigma_2),\end{aligned} \tag{12}$$

where $\Psi$ is a defined function with $\Psi(a, b, \mu, \sigma) := \int_{-\infty}^{b}(a - z)\frac{1}{\sigma}\phi(\frac{z-\mu}{\sigma})\mathrm{d}z$, $\Phi$ and $\phi$ represent the cumulative distribution function and probability density function of the standard normal distribution, respectively. The subscripts 1 and 2 of $y, \mu, \sigma$ correspond to the prediction of different DGPs.

### C. DGMBO Algorithm

Therefore, DGMBO calculates the EHVI based on two DGP surrogate models and then selects the point with the largest EHVI. After adding the true values of this point to the data set, the surrogate models will be updated, which is an optimization loop. Repeat the above processes until the stopping criterion is reached. The main method is summarized as Algorithm 1.

**Algorithm 1:** DGMBO

```
Input: Number of initial samplings $n_0$, number of
       added points $n_1$, black-box functions $\boldsymbol{g}$
Output: Pareto-front approximation set $Y$
Initialize by Sobol sampling $(\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n_0)})$;
Evaluate initial points and store them in
  $D = ((\boldsymbol{x}^{(1)}, \boldsymbol{g}(\boldsymbol{x}^{(1)})), \dots, (\boldsymbol{x}^{(n_0)}, \boldsymbol{g}(\boldsymbol{x}^{(n_0)})))$;
Calculate the non-dominated subset $Y$;
$i = 0$;
while $i \le n_1$ do
    Train noisy DGP models $F_1, \dots, F_m$ based on $D$;
    Calculate EHVI based on $F_1, \dots, F_m$;
    Maximize EHVI to find the promising point $\boldsymbol{x}^*$;
    Update $D$ by $D = D \cup (\boldsymbol{x}^*, \boldsymbol{g}(\boldsymbol{x}^*))$;
    Update $Y$ by the non-dominated subset of $D$;
    $i = i + 1$;
end
return $Y$
```

## IV. Experimental Results and Analysis

In this section, the performance of the proposed algorithm is verified from two aspects, standard test functions and the simulation model of copper flash smelting. The comparison algorithm is multi-objective Bayesian optimization based on GP [26], referred to as GMBO.

### A. Experimental Setup

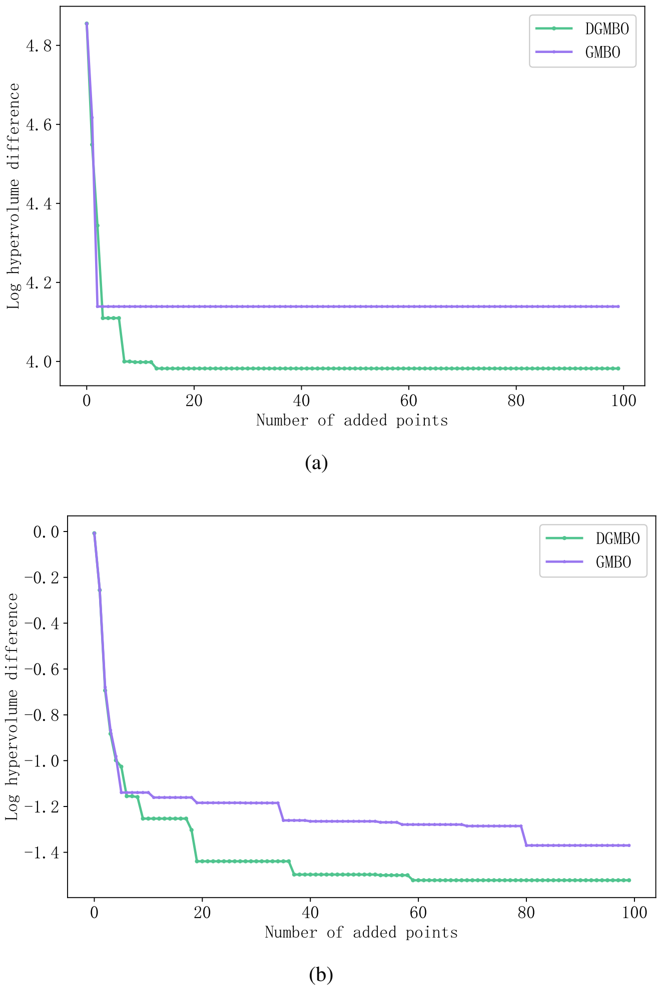
We use the process simulation software *SysCAD* [27] to establish the simulation model of copper flash smelting, and write the interface between Python and *SysCAD*. Data collection and operation of the simulation model can be realized through programming automation, so as to enable real-time Bayesian optimization of the simulation model of copper flash smelting. Each experiment begins with $5 \times d$ initial samplings obtained through Sobol sampling [28], and the number of iteratively added points is $10 \times d$, where $d$ is the number of decision variables. The training of DGP is based on the Adam optimizer, with the learning rate of 0.01, the batch size of 100, and 200 epochs.

### B. Evaluation Indicator

The hypervolume (HV) metric [24] is used to evaluate the performance of algorithms. HV represents the hypercube volume formed by the reference point and the Pareto front obtained by the algorithm. The greater the HV value, the closer the obtained solution set is to the true Pareto front in terms of convergence and diversity. Convergence indicates a good approximation between the obtained solution set and the true Pareto front, while diversity indicates an even and wide distribution of individuals in the obtained solution set. HV is defined as follows:

$$\mathrm{HV}(Y) = \lambda_m\Big(\bigcup_{\boldsymbol{y} \in Y} [\boldsymbol{y}, \boldsymbol{r}]\Big), \tag{13}$$

where $Y$ is the solution set obtained by the algorithm, $\boldsymbol{y}$ is one solution in the solution set, $\boldsymbol{r}$ is the reference point, and $\lambda_m$ represents the $m$-dimensional Lebesgue measure.



(a)

(b)

Fig. 4. Convergence curves of log hypervolume difference on (a) *DTLZ1*, (b) *DTLZ2*.

### C. Evaluation on Standard Test Functions

To validate the effectiveness of the proposed algorithm, the standard multi-objective optimization test functions *DTLZ1* and *DTLZ2* [29] are used for experiments, which exist multiple local optima. For each test function, the number of objective functions is 2, the number of decision variables is 10, and the domain of definition is $0 \le x_i \le 1, i = 1, \cdots 10$. Standard test functions contain the true Pareto front, allowing an accurate and efficient evaluation of different algorithms. We calculate the log hypervolume difference between the true Pareto front and the obtained Pareto front. The smaller the log hypervolume difference, the better the corresponding algorithm.

Fig. 4 shows the log hypervolume difference varying with iteration. It can be seen that the log hypervolume difference of DGMBO decreases faster and is ultimately smaller. Consequently, the optimal solution obtained by DGMBO is closer to the true Pareto front. This is due to the fact that DGMBO can

TABLE I
THE REPRESENTATION AND LIMIT OF DECISION VARIABLES.

| Variable | Representation | Unit | Lower limit | Upper limit |
|---|---|---|---|---|
| $x_1$ | Air input to the reaction shaft of the flash furnace | t/h | 40 | 50 |
| $x_2$ | Copper concentrate input to the reaction shaft of the flash furnace | t/h | 80 | 90 |
| $x_3$ | Heating oil input for energy | t/h | 0 | 2 |
| $x_4$ | Electrode input to the electric furnace | t/h | 0 | 2 |
| $x_5$ | Air input to the electric furnace | t/h | 0 | 2 |
| $x_6$ | Boiled feed water input to the waste heat bolier | t/h | 20 | 30 |
| $x_7$ | Mass percentage of C10H8 in heating oil | % | 30 | 50 |
| $x_8$ | The temperature of the steam dryer | °C | 110 | 130 |
| $x_9$ | Required moisture percentage after drying | % | 0.1 | 1 |
| $x_{10}$ | The pressure of waste heat boiler | kPa | 101.33 | 130 |

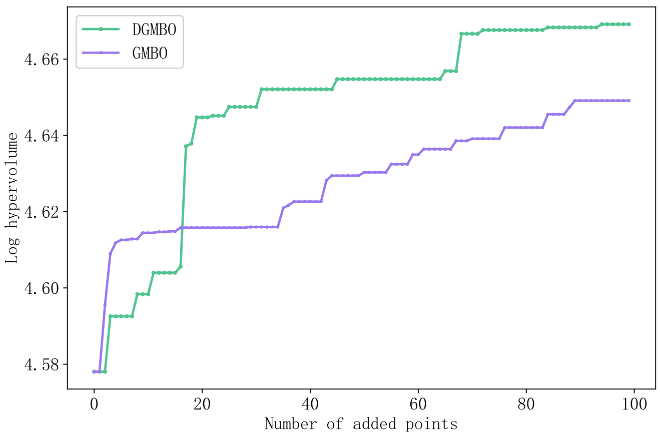


Fig. 5. Optimization performance on copper flash smelting.

TABLE II
THE FINAL RESULTS OF GMBO AND DGMBO.

| | Initialization | GMBO | DGMBO |
|---|---|---|---|
| Matte production (t/h) | 22.72 | 24.50 | **24.88** |
| Off-gas output (m$^3$/h) | 103085.06 | 105246.62 | **102126.99** |

learn complex functions using DGP as the surrogate model. Then, a more effective sampling point can be found in each iteration, and the surrogate model can be updated to approach the true optimum gradually.

### *D. Application to Copper Flash Smelting Optimization*

To further verify DGMBO in the actual process, the experiment is conducted on the simulation model of copper flash smelting. The optimization problem for copper flash smelting includes 10 decision variables, denoted as $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, $n = 10$, and the details are shown in Table I. Copper output $Q(\boldsymbol{x})$ and off-gas output $C(\boldsymbol{x})$ are two objective functions, resulting in a balance between copper production and environmental pollution. To simulate the noise of the actual process, we add Gaussian noise $\epsilon \sim \mathcal{N}(0, 0.3^2)$ to each objective function. The problem can be expressed as follows:

$$\begin{aligned} &J \sim \{\max Q(\boldsymbol{x}), \min C(\boldsymbol{x})\}, \\ &\text{s.t.} \;\; x_i \in [x_{i,\min}, x_{i,\max}], \; i = 1, \cdots, n. \end{aligned} \tag{14}$$

Fig. 5 shows the log hypervolume varying with iteration for different algorithms. An algorithm with a larger log hypervolume value has better convergence and diversity. DGMBO is slightly behind DGMBO at the beginning of the iteration. After several iterations, DGMBO is consistently higher and rises faster. It indicates that with the addition of new points, DGP gradually gains advantages in coping with non-stationary problems, and the model is approaching the truth. As a result, the final HV value of DGMBO is higher, indicating good convergence and diversity of the corresponding solution set.

Table II shows the final optimization results of the two algorithms on the copper flash smelting model. While GMBO increases matte production, it also leads to an increase in off-gas emissions, resulting in environmental pollution. Compared to GMBO, DGMBO significantly improves matte production as well as reduces off-gas emissions. It indicates that DGMBO is more suitable for dealing with the non-stationary copper smelting, and it can balance multiple objectives with better results in terms of economy and environment.

## V. CONCLUSION

Copper smelting optimization is a non-stationary, noisy, and multi-objective problem. As a solution, we apply Bayesian optimization to the complex copper smelting optimization, and make adaptive adjustments in conjunction with specific application scenarios. Firstly, in view of the non-stationary nature and inherent observation noise of the copper smelting process, noisy deep Gaussian process is adopted for modeling to fit the objective function better. Secondly, for the case of multiple competing objective functions, multiple noisy deep Gaussian processes are used to model multiple objective functions, respectively. Then the acquisition function is designed based on expected hypervolume improvement to find the Pareto solution. Finally, experiments are conducted on standard test functions and the simulation model of copper flash smelting, which shows that the proposed algorithm can greatly improve the optimization efficiency and lead to better solutions. Overall, DGMBO can effectively solve the non-stationary, noisy and multi-objective problem of copper smelting optimization, and quickly identify the optimal solution under the limited number of evaluations.

## REFERENCES

[1] J. H. Liu, W. H. Gui, Y. F. Xie, and Z. H. Jiang, "Solving the transient cost-related optimization problem for copper flash smelting process with legendre pseudospectral method," *Materials Transactions*, vol. 54, no. 3, pp. 350–356, 2013.

[2] J. Wang, W. Zhang, and C. Tong, "Thermodynamic model of freeze slag inside reaction shaft of copper flash smelting furnace," *Nonferrous Metals Science and Engineering*, 2014.

[3] S. H. Ahn, M. Lee, J. Chang, S. Y. Lee, and H. S. Sohn, "Dust reaction model in waste heat boiler of copper smelting," in *Proc. International Copper Conference*, 2019.

[4] Q. M. Wang, X. Y. Guo, and Q. H. Tian, "Copper smelting mechanism in oxygen bottom-blown furnace," *Transactions of Nonferrous Metals Society of China*, vol. 27, 2017.

[5] A. K. Biswas and W. G. Davenport, *Extractive Metallurgy of Copper*. Pergamon, 2011.

[6] L. Jiang and D. Zhu, "Mathematical model and application of optimal control matte grade in copper smelting process," *Nonferrous Metals Engineering*, vol. 8, no. 4, pp. 68–72, 2018.

[7] J. Oshitani, K. Teramoto, M. Yoshida, Y. Kubo, S. Nakatsukasa *et al.*, "Dry beneficiation of fine coal using density-segregation in a gas-solid fluidized bed," *Advanced Powder Technology*, vol. 27, pp. 1689–1693, 2016.

[8] X. Ying, C. Wei, D. Apley, and X. Ding, "A non-stationary covariance-based kriging method for metamodelling in engineering design," *International Journal for Numerical Methods in Engineering*, vol. 71, no. 6, pp. 733–756, 2010.

[9] Q. Zhang, J. Chen, Y. Wu, H. Liu, and H. Yang, "Suppressing so3 formation in copper smelting flue gas by ejecting pyrite into flue," *Environmental Science and Pollution Research*, no. 2, 2020.

[10] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. D. Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[11] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Advances in Neural Information Processing Systems*, vol. 4, 2012.

[12] E. Brochu, V. M. Cora, and N. D. Freitas, "A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *Computer Science*, 2010.

[13] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[14] A. Hebbal, L. Brevault, M. Balesdent, E.-G. Talbi, and N. Melab, "Bayesian optimization using deep gaussian processes with applications to aerospace system design," *Optimization and Engineering*, vol. 22, pp. 321–361, 2021.

[15] J. R. Gardner, M. J. Kusner, Z. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints," in *Proc. International Conference on Machine Learning (ICML))*, Beijing, China, 2014.

[16] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[17] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[18] A. D. Bull, "Convergence rates of efficient global optimization algorithms," *Journal of Machine Learning Research*, vol. 12, no. 10, pp. 3–4, 2011.

[19] N. D. Freitas, A. Smola, and M. Zoghi, "Exponential regret bounds for gaussian process bandits with deterministic observations," in *Proc. International Conference on Machine Learning (ICML)*, 2012.

[20] X. Peng, W. Gui, Y. Li, Z. Hu, and L. Wang, "Operational pattern optimization for copper flash smelting process based on pattern decomposition of fuzzy neural networks," in *Proc. IEEE International Conference on Control and Automation*, 2007.

[21] J. L. Ding, C. E. Yang, Y. D. Chen, and T. Y. Chai, "Research progress and prospects of intelligent optimization decision making in complex industrial process," *ACTA AUTOMATICA SINICA*, vol. 44, no. 11, pp. 1931–1943, 2018.

[22] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep gaussian processes," in *Proc. Conference on Neural Information Processing Systems (NIPS)*, 2017.

[23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.

[24] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE transactions on evolutionary computation*, vol. 7, no. 2, pp. 117–132, 2003.

[25] K. Yang, M. Emmerich, A. Deutz, and T. Bäck, "Efficient computation of expected hypervolume improvement using box decomposition algorithms," *Journal of Global Optimization*, vol. 75, no. 1, pp. 3–34, 2019.

[26] L. Chang, K. Shimoyama, and S. Obayashi, "Kriging model based many-objective optimization with efficient calculation of expected hypervolume improvement," in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 290–294.

[27] L. Tan, "Syscad and its application in flowsheet simulation for process industries," *Computers and Applied Chemistry*, vol. 26, no. 10, pp. 1217–1227, 2009.

[28] N. Dige, "Efficient sampling algorithm for optimization under uncertainty," 2016.

[29] H. Masuda, Y. Nojima, and H. Ishibuchi, "Common properties of scalable multiobjective problems and a new framework of test problems," in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, 2016.