Optimal-Setting for Ore and Water Feeding in Grinding Process Based on Improved Case-Based Reasoning

Bingyu Liu †,‡, Dezhi Hao *,†,‡, Xianwen Gao †,‡ and Dingsen Zhang †,‡

Abstract: The grinding product particle size is the most crucial operational index of mineral grinding processes. The size and consistency of the product directly affects the subsequent dressing and sintering. In this paper, a novel expert system is proposed for guiding the operating variables to keep the product stable with the wildly varying ore properties. First, case-based reasoning (CBR) is introduced to describe the whole grinding process with the historical data and expert experience. Second, the generative adversarial network (GAN) is employed to extend the raw data to enhance the flexibility of CBR. Moreover, the weights of different features in CBR is optimized by improved non-dominated sorting genetic algorithm II (NSGA-II). Finally, the proposed method is validated by a set of actual data collected from a Chinese dressing plant. The experimental result demonstrates the effectiveness of the proposed method.

Keywords: case-based reasoning; GAN; optimization; mineral grinding processes; particle size

1. Introduction

In mineral processing, grinding is used to reduce the particle size of the ore so that the valuable mineral constituent is exposed and can be recovered in the subsequent beneficiation operation, such as flotation or magnetic separation. The product particle size (PPS) is the most crucial production index which is a required constant to ensure the quality of the subsequent product. However, grinding, which is a complex and time-variant process, is affected by a large number of operational parameters. What is more, the iron ore’s composition is unstable with the decrease in mineral resources [1].

Because of these difficulties, the research on the grinding process has received much attention. Generally, the PPS needs to be controlled within the desired range suitable for the subsequent beneficiation process. With the development of new intelligent modeling and control methods, many intelligent modeling methods have been successfully implemented in the modeling process of grinding processing. Recently, the advanced control methods based on model predictive control [2–4] or supervisory control [5,6] for the grinding process have been discussed. It is worth noting that the results mentioned above are stringent based on the mathematical models of the system. However, the grinding process is difficult to describe by mathematical models since the actual industrial process is essentially a multi-input–multi-output system (MIMO) with large inertia, nonlinearity, strong coupling, uncertainty, and interference factors [7]. Moreover, due to the lack of online analyzers measuring the grinding particle size and the difficulty of obtaining transfer functions of the grinding process, many concentrators cannot realize optimal control by adopting the model-based control method proved to be effective in other plants [8]. The distinct element method, finite element method, smooth particle hydrodynamics method and discrete element coupling algorithm were adapted to the grinding process, focusing on the media motion state, mechanism research, mill parameter optimization research, ball mill
power consumption and energy distribution \[9,10\]. The coupling of various methods can realize the combination of advantages of various methods and simulate the actual working conditions of the ball mill more truly and comprehensively \[11–13\]. However, due to the complex and unstable model, the uncertainty of ore properties, high technical experience requirement and high computational cost, the progress of the coupling method in the mill simulation is difficult to implement. With the development of science and technology, a hybrid intelligent optimal control method based on an expert system for the operation of the complex industrial process is proposed. CBR \[14,15\], rule-based reasoning (RBR) \[16,17\] and a fuzzy expert system \[18–20\] are applied to provide potential solutions for such grinding problems. Among these methods, CBR is an approach to problem solving that utilizes previous cases and experiences that are similar to the current one. The advantage of this method is that it bypasses the unavailable parameters and makes full use of the available experience and data to obtain satisfactory results with little change in the process equipment. Compared with other intelligent algorithm, another important advantage of CBR is that it enables sustained learning by updating the case base after a problem has been solved.

Based on the previous research, the ball mill ore and water feeding intelligent optimization based on CBR are proposed. Historical production data are firstly switched into the form of the case, which include characteristics and characteristic solutions. The characteristics consist of the crushing ore size, ore taste, magnetic tube $mFe$ content, magnetic tube recovery rate, raw ore $FeO$ content, raw ore $mFe$ content, and overflow particle size target value, while the characteristic solution is the ball mill ore feeding and water feeding. Then $k$-nearest neighbor (KNN) algorithm is applied to calculate the similarity between cases in cases retrieved with the Euclidean distance. The accuracy of CBR is mainly affected by the quantity, quality, and weight of cases. Considering that it is very difficult to obtain a large number of samples in industrial practice, this paper adopts the method of data expansion to increase the sample density. GAN, proposed by Goodfellow et al. in 2014, has been proved to be the most effective method of data augmentation in recent years \[21\]. GAN is different from other data expansion methods, such as the synthetic minority oversampling technique (SMOTE) and variational auto encode, in that it pays more attention to learning the internal distribution of the original data rather than differentiating or adding noise to the original data. Through the zero-sum game, GAN can be trained to generate more realistic data. GANs have been successfully applied in generating realistic data in many fields. Shao et al. employed GAN to enrich fault samples to detect the faults in an induction motor \[22\]. Wang et al. proposed a new model of fault diagnosis by combining GAN and stacked denoising autoencoders to recognize the planetary gearbox fault pattern. Mao et al. \[23\] and Liu et al. adopted GAN to generate a large number of fault samples to obtain a balanced data set and then input it into a classifier to realize the fault diagnosis of the induction motor \[24\]. Spyridon proposed an unsupervised fault detection scheme based on GAN in which the judgment on fault occurrence is made by the discriminator \[25\].

By its very nature, the distribution of attribute weights also influences the process of similarity assessment. Considering that the operating parameters of the grinding process are not single, it means all case solutions should be considered when weighting the case description. However the commonly used CBR assigns weights through expert experience, which is generally qualitative rather than quantitative. To address the above issues, this paper used NSGA-II which has been used in industrial production \[26–28\] for multi-objective optimization of weights based on historical cases.

The main contributions of this paper can be summarized as follows. (1) The CBR-based grinding parameter optimization algorithm is proposed to maximize the use of available data and expert experience for plants with changing ore properties and incomplete monitoring equipment. (2) The operational parameters and observation data are simplified by combining the process and actual field conditions, and the GAN algorithm is used to realize the expansion of data. (3) The paper introduces an allocation method of the attribute weights, based on NSGA-II, to make it efficacious and precise to multiple case solutions.
(4) The method proposed in this paper was experimented in a dressing plant and achieved satisfactory results, which provides experience for subsequent studies.

The paper is organized as follows: Section 2 presents background information about the ball mill grinding process and the data augmentation method. Section 3 introduces CBR and optimizes the weight of each characteristic by NSGA-II. This is followed by Section 4, which presents the simulation results of the optimization, comparing them with the other algorithms. Section 5 presents the application results in a Chinese ore dressing mill. The last part gives the conclusion.

2. Parameter Selection and Data Reduction

In this section, two-stage closed-circuit mineral grinding processes are first introduced. Combined with the actual situation, some key parameters are selected to optimize the grinding processes. Then, the collected data are expanded by the data augmentation method to make the data-driven model more precise.

2.1. Parameter Selection

Grinding process is the sequel of the ore crushing process, whose purpose is to produce useful components of the ore to reach all or most of the monomer separation, while avoiding an excessive wear phenomenon and achieving the particle size requirements for sorting operations. The typical two-stage closed-circuit wet-type mineral grinding process considered in this paper is shown in Figure 1. The coarse ore from the primary crusher is mixed in the ore bin and then fed into the ball mill together with water for wet grinding. By changing the frequency of the pendulum feeder, we can control the quantity entering the ball mill. The tumbling and crushing action with the grinding medium grinds the coarse ore to finer sizes. The slurry containing the fine product is discharged from the ball mill to the pump sump and then pumped to hydrocyclones for classification. The slurry is separated into two streams in the hydrocyclones: the underflow with the larger particles and the overflow with the finer particles. The underflow is recycled back to the ball mill with the new coarse ore for further regrinding. The overflow slurry flows to the next stage of the grinding system for finer particle grinding. The next stage of the grinding process is the same as the first stage, and the overflow flows into the subsequent beneficiation operation as the desired product.

![Figure 1. Flow chart of two-stage grinding process.](image)

The grinding process is a complex controlled object that is influenced by many factors. PPS is the key operational index that dictate the product quality and is usually used as the controlled variable in the grinding process. The manipulated variables are listed as follows: the milling ore feed velocity (denoted by \( f_1 \)), the water feed velocity of the two-stage hydrocyclones (denoted by \( w_{11}, w_{12} \)), the pump water feed velocity of the two-stage sumps (denoted by \( w_{21}, w_{22} \)), and the frequency of the two-stage pumps (denoted by \( p_1, p_2 \)). Combined with the actual situation of a concentrator in northeast China, the fresh ore is mainly composed of hematite, magnetic ore, and carbonate iron ore. We use
the characteristics of the crushing ore size, ore grade, magnetic tube \( mFe \), magnetic tube recovery rate, raw ore \( FeO \) content, and raw ore \( mFe \) content to describe them in this paper.

The calculation of adjusting all of the above-mentioned operating parameters, according to the nature of the ore, is too complex, and the construction on site has a large lag and uncertainty. So combined with the actual situation of a concentrator in northeast China, this paper simplifies as follows. (1) Long-term industrial tests showed that the grinding efficiency is highest when the underflow concentration of the two-stage cyclone are around 65% and 85%, respectively. When the amount of ore and water is well controlled so that the concentration of one stage of underflow is 65%, the concentration of the second-stage underflow is exactly 85% without adding water to the cyclone. So the water feed rate of the second-stage hydrocyclones is set to zero. (2) The main function of water feeding to the sumps is to avoid slurry precipitation. So the water feed rate of the two-stage sumps is a fixed value that does not changes as the manipulated variable. (3) The frequency of the two-stage pumps is also fixed, based on long-term industrial trials. (4) Operators can adjust the amount of steel ball supplied according to complete real-time regulation. According to the above assumptions, this paper’s operating parameters are simplified to the water feed velocity of the first-stage hydrocyclones and the milling ore feed velocity. These parameters need to be adjusted, according to the ore properties fluctuation and PPS requirements.

2.2. Data Augmentation

Limited by the upgrading of technology and the backwardness of modern equipment on the industrial site, there are not enough data for parameter optimization and calculation. The primary purpose of the method in this section is to generate a large amount of data so as to provide sufficient samples for data-driven methods to calculate.

The GAN was proposed by Goodfellow et al. in 2014, and consists of two essential components, namely, the generator and the discriminator. Through the zero-sum game, GANs can be trained to generate more realistic data. GAN is widely used in image generation and recognition, speech generation and recognition, data generation, and other fields. The main thought behind GAN is using adversarial networks to improve the quality of generated data. The generator learns the probability distribution of the original data and generates artificial samples that mimic the pattern, using random noises. The discriminator discriminates the artificially generated data from the actual data and prompts the generator to produce better quality data in the next iteration. The general semantics flow chart of GAN is depicted in Figure 2.

The GANs framework is equivalent to a minimax two-player game, which can be described by the following function [21]:

\[
\min_G \max_D V(D,G) = \min_G \max_D \left( \mathbb{E}_{x \sim p_{data}} \left[ \log D(x) \right] + \mathbb{E}_{z \sim p_z} \left[ \log (1 - D(G(z))) \right] \right)
\]

where \( x \) is a real datum with a distribution, and \( p_x \) and \( z \) is a random vector. \( z \) is made to be an input of \( G \) through which the generated datum \( G(z) \) can be obtained. \( p_g \) is denoted

---

**Figure 2.** Typical architecture of GAN.
as the distribution of $G(z)$. Then, $D$ represents the probability that $x$ comes from $p_x$, rather than from $p_g$. Ideally, $D(x) = 1$ if $x \sim p_x$ and $D(x) = 0$ if $x \sim p_g$. $D$ is trained to maximize the probability of assigning the correct label to both the real datum $x$ and the generated datum $G(z)$.

In this paper, GAN is used to generate grinding industry data. The data include the crushing ore size, ore grade, magnetic tube $mFe$, magnetic tube recovery rate, raw ore $FeO$ content, raw ore $mFe$ content, the water feed rate, and the fresh ore feed rate. After the actual field investigation records, 182 sets of data were used as the original sample.

The data from the actual production process have some incorrect data. Some of the values are 0, while others are either unreasonably big or small. To get rid of the abnormal data, Formula (2) is used firstly for data screening.

$$|X - \bar{X}| \geq a \times \sqrt{\frac{1}{n} \sum_{i=1}^{n} (X_i - \bar{X})^2}$$

where $X_i$ is the $i$th parameter value; $\bar{X}$ is the average of the values of the set of parameters. When one of the parameter values is greater than $a$ times the standard deviation, the sample should be deleted.

After data screening, 164 sets of data are left to generate samples. In the generator, the input is composed of the noise input, which contains 80-dimensional data, randomly generated by Gaussian distribution. There are two hidden layers, followed by the rectified linear unit (ReLU) as an activation function, and the kernel sizes are 120 and 240. The output of the generator is a one-dimensional data sample, using Sigmoid as the activation function. As for the discriminator, the input layer is with eight kernels, using LeakyReLU as an activation function, and the kernel size is 120. Another layer with 240 kernels using LeakyReLU is added and followed by dropout with the probability of 0.5. Finally, the model layer is flattened and linked to one fully connected layer with 0.5 dropouts.

GAN generates 886 samples. To verify the accuracy of the generated samples, the back-propagation neural network (BP), support vector machines (SVM), and extreme learning machine (ELM) are used to compare the original data and the generated data. We used 30% of the original samples as test samples.

The mean absolute error (MAE), mean absolute percentage error (MAPE) and root mean square error (RMSE) of the generated data and the original data under different algorithms are shown in Tables 1 and 2.

| Table 1. Comparison of setting errors of grinding ore feeding. |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                | BP       | SVR       | ELM       |                 |
| MAE             | Original  | Generated | Original  | Generated       |
|                 | 6.761    | 5.8759    | 8.0414    | 5.629           |
| MAPE            | 0.02219  | 0.0194    | 0.0266    | 0.0184          |
| RMSE            | 8.7792   | 7.2729    | 10.25     | 6.9361          |

| Table 2. Comparison of setting errors of water feeding. |
|-----------------|-----------------|-----------------|-----------------|-----------------|
|                | BP       | SVR       | ELM       |                 |
| MAE             | Original  | Generated | Original  | Generated       |
|                 | 0.5184    | 0.4183    | 0.5993    | 0.3826          |
| MAPE            | 0.017    | 0.0137    | 0.0197    | 0.0124          |
| RMSE            | 0.9114   | 0.5039    | 0.7644    | 0.5097          |

As can be seen from the tables, the generated samples can better improve each algorithm’s accuracy. The reason is that the original data volume is relatively small and the
data interval is large, which makes it difficult to fit the data well. GAN can supplement the data to simulate some working conditions that are not in the original data but may occur, so that the database contains more working conditions. Although the generated data may not definitely have a positive effect on the optimization, the validity of the generated data can be proven after several generation experiments when all different classification methods show that the expanded data fit better than the original data.

3. Optimizing Setting Based on CBR

Generating samples can improve the accuracy of the algorithm when the original samples are insufficient. However, the generated data do not have the same accurate guiding significance as the measured data. Some generated data may also be misleading for subsequent data-driven training. Therefore, an algorithm is needed to make the original sample and the generated sample with different weight coefficients so that the generated data only play an auxiliary role rather than a deciding role. As an algorithm that relies on expert experience, CBR has the advantages of fast speed, strong adaptability, and low hardware equipment requirements. These advantages meet the needs of the grinding site very well. This chapter improves CBR by distinguishing between the generated samples and original samples. What is more, the feature weights are optimized by an evolutionary algorithm.

3.1. Basic Flowchart of Case-Based Reasoning

The case-based reasoning process is mainly divided into four basic steps: (1) case retrieval in which, by a series of searching and similarity calculations, the most similar case with the current problem is found in the case base; (2) case reuse, which compares the differences between the source case and the target case. Then, the solution case recognized by the user is submitted to the user, and the election of its application is observed; (3) case revision in which the solution strategy is adjusted by combining the elect of the case reuse and the current issue to fit the current problem; and (4) case storage in which the current issue is resolved and stored in the case database for future use [28]. In the CBR process, case retrieval is the core of CBR technology, which directly determines the decision-making speed and accuracy.

Together with the real problems that exist in grinding process control with the theory of case-based reasoning, the basic procedure of the ball mill feed intelligent optimization is shown in Figure 3 [29]. By carrying out a comprehensive mathematical problem in the engineering analysis and case-based reasoning for the complex process, the intelligent ball mill feed is obtained in an optimized manner. The details are presented as followed.

![Figure 3. CBR-based optimization strategy for ball mill feeding.](image)

3.2. Improved Case-Based Reasoning

3.2.1. Case Description

In a CBR model, the samples in the case base should be representative and cover the typical operational conditions as much as possible. The model’s generalization ability
could be limitive if a series of continuous samples are directly selected to be the case base. On the other side, if a large amount of data that might cover all the conditions becomes the case base samples, high-level data redundancy might appear. The model precision might be negatively impacted if the anomaly data are added into the case base.

Usually, each case consists of characteristics and characteristic solutions. To distinguish the impact of confirmed cases and generated cases on retrieval, this article adds case weights to describe each case further. The characteristics contain the crushing ore size \( f_1 \), ore grade \( f_2 \), magnetic tube \( mFe \) \( f_3 \), magnetic tube recovery rate \( f_4 \), raw ore \( FeO \) content \( f_5 \), raw ore \( mFe \) content \( f_6 \) and overflow particle size target value \( f_7 \). The characteristic solutions are the ball mill ore feed \( j_1 \) and water feed \( j_2 \). The weight case is defined as \( \beta \).

The case model in the CBR process is described as \( C_k = \{ F_k, l_k, \beta_k \} \). \( C_k(k = 1, \ldots, m) \) means there are \( m \) cases in total, \( C_k \) stands for the \( k \)th case in the case base. \( F \{ f_{k,1}, f_{k,2}, \cdots, f_{k,7} \} \) expresses the characteristics of what \( C_k \) describes. \( l_k = \{ j_{k,1}, j_{k,2} \} \) expresses the characteristic solutions of the case \( C_k \). \( \beta_k \) expresses the case weight of the case \( C_k \).

### 3.2.2. Case Retrieve and Matching

Case matching and case retrievals are essential steps in the case-based reasoning process and the key of the information extraction from the case databases. In general, the case matching strategy includes the serial and parallel search methods. In the serial search process, the cases are organized in a hierarchical manner. The top-down refinement layer by the layer retrieval approach is adopted, which means that the further down the layer, the higher the similarity. The parallel searching strategy weakens the level features among the cases. The retrieve method returns to the most similar case by retrieving many cases.

The parallel case matching and case retrievals are essential steps in the case-based reasoning process and the key of the information extraction from the case databases. In general, the case matching strategy includes the serial and parallel search methods. In the serial search process, the cases are organized in a hierarchical manner. The top-down refinement layer by the layer retrieval approach is adopted, which means that the further down the layer, the higher the similarity. The parallel searching strategy weakens the level features among the cases. The retrieve method returns to the most similar case by retrieving many cases.

When the new case \( C \) comes, we define the \( \text{SIM} = \{ F, F_k \} \) as the similarity of \( C \) and \( C_k \). We can use the Euclidean distance to calculate the similarity as Equation (3):

\[
\text{SIM}(F, F_k) = \frac{\sum_{i=1}^{7} \omega_i (1 - \frac{|f_i - f_{k,i}|}{\max_f (f_i, f_{k,i})})}{\sum_{i=1}^{7} \omega_i}
\]

(3) where \( \omega_i \) is the weight of \( f_i \).

### 3.2.3. Case Reuse

In the actual generation process, the cases in the case library cannot cover all work situations. Therefore, the similar case solution retrieved must be reused, that is to say, the CBR system adjusts the retrieved case solution, according to the specific circumstances of the new case, to get the solution of the new case. After the case retrieving and matching, cases in the case base are arranged in the order of highest to lowest by the value of \( \text{SIM} \). Then, the k-nearest neighbor algorithm (KNN) is used in this step: the first \( n \) historical cases are chosen as reference cases, and their solutions get reused by using formula (4) [30].

\[
J = \frac{\sum_{i=1}^{n} \text{SIM}(F, F_{k(i)}) \times l_k \times \beta_k}{\sum_{i=1}^{m} \text{SIM}(F, F_{k(i)}) \times \beta_k}
\]

(4) The value of \( n \) definitely affects the final solution. If the value is much larger than the reasonable value, some insufficient cases get involved in the case reuse process. The accuracy of the case solution is affected undesirably. While if the value is too small, there are not enough cases to reflect all the features of the new case. So, either a too-big value or a too-small value makes the solution unfaithful. The value \( \beta_k \) increases the proportion
of the original cases. This parameter reduces the weight of the generated case, which is optimized with $\omega_i$.

3.2.4. Case Maintenance

The accuracy of the cases in the case base is directly reflected in the case solution of CBR. Case maintenance stores the new cases and their solutions into a case database, according to a certain strategy. The newly solved case needs to be stored to increase the searching scope of the case base and adapt the case base to the changing working conditions. However, an incorrect case contaminates the case base and greatly affects the accuracy of the CBR algorithm permanently. What is more, the unrestrained storing of new cases increases the calculation time constantly. Therefore, there should be a strategy in judging whether the new case should be retained or not. This paper puts forward the strategy by the following steps:

1. Experts judge whether the new case is correct or not. The influence of errors, such as input error, can be reduced by expert judgment.
2. Set $m_{\text{max}}$ as the maximum storage amount. Set $S_{\text{max}}$ as the maximum storage similarity. Identify the maximum similarity $SIM_{\text{max}}$ of all the $SIM(F, F_{(k)})$. Identify the $m$ as the amount of the latest case base.
3. If the retrieved case with $SIM_{\text{max}}$ is the generated case, delete this case and store the new case in the case base to replace it.
4. If the retrieved case with $SIM_{\text{max}}$ is the original case, make decisions as follows:
   - If $SIM_{\text{max}} \geq S_{\text{max}}$, which means that there are enough similar cases in the case base, save the case to the case base and delete the most similar generated case before. If there are no generated cases in the case library, delete the closest original case.
   - If $SIM_{\text{max}} < S_{\text{max}} \cap m \leq m_{\text{max}}$, save the new case in the case base.
   - If $SIM_{\text{max}} < S_{\text{max}} \cap m > m_{\text{max}}$, save the new case in the case base and delete the oldest case (all generated cases are oldest cases; relatively similar ones are deleted first). The purpose is to update the case base to accommodate the effects of changes in processes, the environment, etc.

3.2.5. Weight Optimization for CBR

$\omega_i$ and $\beta$ are key parameters in the case matching and case retrieval [15]. The value of them directly affects the value of $SIM(F, F_{(k)})$, which in turn affects the final optimization result of CBR. Usually, the value of $\omega_i$ and $\beta$ are distributed by expertise. However, experts can only give $\omega_i$ and $\beta$ qualitative indicators based on experience, and it is difficult for them to determine the exact value. In the paper, we use the NSGA-II intelligent optimization algorithm to calculate $\omega_i$ and $\beta$ based on the historical cases of the case base. The algorithm is summarized in Figure 4. The algorithm steps are as follows:

1. Organize the case base, select 70% original cases and all generated cases as the training samples, and the remaining 30% original cases as the testing samples. Regard the training samples as the new base database. Regard the testing samples as the new bases.
2. Set $\omega_i (i = 1, \ldots, 7)$ and $\beta$ as the decision variables. Generate initial random father populations of $\omega_i$ and $\beta$. Use the method CBR mentioned above to calculate the $J$ of every new case.
3. Build the mathematical model.
   1. Objective functions:
      - The MAE (mean absolute error) between the optimized amount of ore feeding and the actual value of the test set is the smallest.
      - The MAE (mean absolute error) between the optimized amount of water feeding and the actual value of the test set is the smallest.
2. Constraint conditions:

\[
\begin{align*}
\sum_{i=1}^{7} \omega_i &= 1 \\
\text{SIM}(F,F(k)) &\leq 1 \\
0 &\leq \omega_1 \leq 0.1 \\
0.2 &\leq \omega_2 \leq 0.4 \\
0 &\leq \omega_3 \leq 0.2 \\
0 &\leq \omega_4 \leq 0.2 \\
0 &\leq \omega_5 \leq 0.2 \\
0 &\leq \omega_6 \leq 0.1 \\
0.1 &\leq \omega_7 \leq 0.3 \\
0.7 &\leq \beta \leq 0.9
\end{align*}
\] (5)

(4) Adopt the NSGA-II multi-objective optimization algorithm, obtaining the non-inferior solution set through populations, across and mutual.

(5) Adopt the TOPSIS algorithm to obtain the optimal solution of the non-inferior solution set as the optimized feature weight, based on the Euclidean distance.

(6) When the new case comes, adopt the optimized feature weight to case retrieve from the whole case base. Then, solve the ore and water feeding as the typical steps of CBR as introduced above.

Figure 4. Strategy diagram of the proposed optimization combining CBR with NSGA-II.

4. Simulations Results Using Actual Industrial Data

In this section, 164 sets of actual industrial data collected from a Chinese dressing plant are used for simulation to verify the effectiveness and performance of the proposed algorithm preliminarily. The actual production data are expanded with the GAN and optimize the feeding value with the proposed method. In order to verify the accuracy of this algorithm, this algorithm is compared with the SVM, ELM, and traditional CBR algorithms.

4.1. Parameters Training

The 50 sets of the entire 164 sets of original data after data screening are used as the testing data, and the remaining 114 sets and 886 sets of generated data are used as
the training data. Then, by performing multiple simulations in the laboratory, the initial population number is chosen as 200, and evolutionary algebra is set to 80. Figures 5 and 6 show the Pareto optimal fronts for ore and water feeding errors of the first generation and the 80th generation. Inspecting the figures, we can find that training is stabilized in the 80th generation and NSGA-II shows excellent convergence in two optimal objects. The solutions along the Pareto front are equally optimal and the TOPSIS method is applied to select the most appropriate solution.

![Figure 5. Simulation diagram of the first generation.](image)

![Figure 6. Simulation diagram of the 80th generation.](image)

The optimization results of different case features are shown in Table 3. $\omega_1$ to $\omega_7$ describe the importance of each case feature in case retrieval, respectively, and $\omega_8$ represents the importance of the generated cases.

| $\omega_1$ | $\omega_2$ | $\omega_3$ | $\omega_4$ | $\omega_5$ | $\omega_6$ | $\omega_7$ | $\omega_8$ |
|------------|------------|------------|------------|------------|------------|------------|------------|
| 0.067      | 0.358      | 0.179      | 0.126      | 0.047      | 0.043      | 0.18       | 0.83       |
4.2. Further Comparative Analysis

To prove the accuracy of the optimal setting, the other two traditional algorithms (SVR and ELM) are used to compare, based on the expanded data. Further, to give importance to the different weights of the generated case and the original case, the traditional CBR is also added for comparison. Different from the scheme proposed in this paper, the traditional method does not distinguish between generated samples and original samples, and each case description is selected by manual experience. The optimal setting value and actual value of the water and ore feeding with different algorithms are shown in Figures 7 and 8.

![Ore feeding error](image1)

**Figure 7.** Comparison of different methods for ore feeding.

![Water feeding error](image2)

**Figure 8.** Comparison of different methods for water feeding.

The MAE, MAPE, and RMSE of each algorithm are shown in Table 4.

**Table 4.** Comparison of setting errors.

|       | Ore    | Water  | Ore    | Water  | Ore    | Water  | Ore    | Water  |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|
| SVR   | 5.6290 | 0.3826 | 3.0780 | 0.3108 | 2.9606 | 0.44   | 1.9077 | 0.2699 |
| ELM   | 0.0184 | 0.0124 | 0.0125 | 0.0103 | 0.0096 | 0.0143 | 0.0062 | 0.0088 |
| CBR   | 6.9361 | 0.5097 | 5.0378 | 0.5490 | 3.4054 | 0.5310 | 2.2030 | 0.3003 |
As can be seen from the figures and table, the results obtained by CBR accurately vary with the given values. Although there are some deviations, the deviation values are much smaller than other algorithms. At the same time, compared with other algorithms, the CBR have almost no abnormal points. The range of change is smaller and the MAPE is much smaller for both ore and water feeding. The reason is that, compared with other machine learning algorithms, CBR assigns human experience to different case descriptions and places more emphasis on important variables. What is more, the case solutions are jointly derived from similar cases and less affected by outliers. The training process is also simpler than other machine learning algorithms and does not suffer from overfitting or underfitting. Therefore, this method is very suitable for application in practical engineering.

5. Industrial Application Experiments

After testing and comparison with the actual industrial data, the proposed approach was applied to the dressing mill to verify the effectiveness. The average grade of the processed iron ore is 30%. The algorithm is programmed by MATLAB, and the human–computer surface (as shown in Figure 9) is designed by Visual Basic. The surface is a simple version, and the actual application version’s language is Chinese with some additional features, such as querying historical data curves, setting parameters, and so on. Figure 10 is a typical flowsheet of the optimal-setting system running manually. The auto-run indicates that the soft-sensor system carries out estimation operations in accordance with a specified time interval.

![Grinding Process Operating System](image)

Figure 9. Human–computer interaction interface.

When a new set of ore characteristics or new overflow particle size target are obtained, the operator firstly judges whether to update the feature weights or not, then presses “calculate” to run the system. When they get the feedback of the actual overflow particle size, they need to input the data to the system, and the entire group of data saves in the host in the format of EXCEL. After two months of successful operation, a part of the data is shown in Table 5. $f_1$ to $f_6$ in the table represent the ore properties retrieval conditions and $f_7$ represents the target particle size requirement. $J_1$ and $J_2$ are the optimized feeding values and $O$ represents the actual PPS achieved with the optimized feeding values.
Figure 10. Flowsheet of the optimal-setting system.

Table 5. Practical data display.

| Time        | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $J_1$ | $J_2$ | $O$  |
|-------------|-------|-------|-------|-------|-------|-------|------|
| 2019/9/4    | 93.5  | 31.13 | 57.64 | 10    | 77    | 322   | 32.32| 76.7 |
| 2019/9/5    | 93.75 | 32.32 | 57.15 | 8     | 10.55 | 7.72  | 75   | 328  | 31.48| 75.4 |
| 2019/9/6    | 92.75 | 32.28 | 58.34 | 11    | 11.3  | 9.79  | 75   | 325  | 32.46| 75.3 |
| 2019/9/7    | 92.75 | 31.06 | 61.21 | 12    | 15.2  | 8.8   | 72   | 314  | 30.75| 71.3 |
| 2019/9/8    | 93.8  | 29.6  | 61.15 | 21    | 18.3  | 10.6  | 73   | 327  | 31.41| 73.5 |
| 2019/9/9    | 93.8  | 32.03 | 59.01 | 13    | 16.75 | 10.86 | 77   | 300  | 30.96| 75.8 |
| 2019/9/10   | 93.25 | 32.03 | 55.92 | 12    | 12.6  | 11.67 | 75   | 316  | 32.2 | 76.1 |
| 2019/9/11   | 93.25 | 29.65 | 60.84 | 17    | 10.1  | 11.14 | 77   | 298  | 30.84| 77.3 |
| 2019/9/12   | 93.75 | 28.24 | 56.34 | 9     | 6.4   | 7.36  | 75   | 325  | 32.89| 75.4 |
| 2019/9/13   | 93.75 | 33.44 | 57.94 | 5     | 8.2   | 7     | 80   | 313  | 32.1 | 78.3 |
| 2019/9/14   | 93.5  | 32.11 | 58.77 | 7     | 4.2   | 8.26  | 78   | 284  | 29.3 | 77.3 |
| 2019/9/15   | 93.75 | 25.19 | 56.12 | 7     | 9.9   | 9.88  | 69   | 307  | 30.36| 71.1 |
| 2019/9/16   | 93.75 | 25.71 | 60.06 | 8     | 8.1   | 11.8  | 78   | 309  | 30.09| 76.7 |
| 2019/9/17   | 93.25 | 28.52 | 57.46 | 14    | 13.4  | 10.78 | 78   | 311  | 30.89| 77.8 |
| 2019/9/18   | 93.75 | 25.19 | 56.12 | 7     | 9.9   | 9.88  | 69   | 307  | 30.36| 71.1 |
| 2019/9/19   | 93.75 | 25.71 | 60.06 | 8     | 8.1   | 11.8  | 78   | 309  | 30.09| 76.7 |
| 2019/9/20   | 93    | 28.23 | 55.9  | 14    | 16.9  | 11.15 | 75   | 318  | 31.8 | 75.2 |
| 2019/9/21   | 93.5  | 27.88 | 58.53 | 17    | 10.5  | 7.36  | 74   | 321  | 31.88| 74.3 |
| 2019/9/22   | 93.5  | 26.35 | 57.04 | 12.4  | 9.2   | 9.7   | 74   | 306  | 30.56| 73.8 |
| 2019/9/23   | 93.75 | 26.03 | 61.49 | 7     | 10.8  | 8.62  | 76   | 299  | 29.17| 76.4 |
| 2019/9/24   | 93.75 | 26.87 | 62.19 | 8     | 10    | 8.62  | 77   | 282  | 29.47| 76.8 |
| 2019/9/25   | 93.75 | 32.69 | 60.49 | 14    | 13.5  | 8.08  | 79   | 326  | 31.35| 78.3 |
| 2019/9/26   | 93.75 | 30.38 | 58.95 | 21    | 18.5  | 9.7   | 76   | 325  | 32.75| 75.6 |
| 2019/10/1   | 93    | 29.93 | 60.13 | 6     | 12.6  | 9.7   | 83   | 329  | 33.28| 80.2 |

Figure 11 shows the target and the actual value of the overflow particle size. The mean absolute error (MAE) of the target and the actual value is 1.1289. This error is within the allowable range. What is more, the curve of the actual value and the target value are basically consistent. The algorithm can quickly and accurately calculate the set value of the ore and water feeding, according to the ore characteristics and the target of the overflow particle size compared with the traditional manual method. However, when the target value is less than 72 or greater than 80, the error is large. This is due to the following factors:
(1) The cases in the case base are not enough, especially beyond 72–80. The cases cannot contain all the characteristics of the ore. (2) The target value beyond 72–80 means that the ore is much better or worse than the normal state. The other operating parameters of the grinding process are not adjusted according to the ore properties. In industrial applications, some error, such as human error and equipment error, makes the algorithm have some deviation, and some special cases also affect the accuracy of the system. As the optimization system is used for a long time, a large number of cases covering various situations are stored in the case base and the algorithm will be more accurate.

6. Conclusions

To solve the unstable production process caused by the ore and water feeding’s fluctuation properties in the ball mill, the improved CBR is proposed in this paper. Firstly, the GAN algorithm is used to expand the data in view of the shortage of field data collection. Then CBR algorithm solves a new problem with the experience of dealing with similar ones. Based on the similarity between the new case and the old cases, CBR can give the ball mill feed quantity based on historical data. CBR can also update its case base according to new cases to constantly adapt to new conditions. In order to improve the accuracy of the algorithm, NSGA-II is applied to identify the key parameter, which is the weight of each characteristic and the coefficients that distinguish between accurate and generated data. The method is then verified by experimental simulation that it is feasible and has higher accuracy than other algorithms. The application results show that the system achieved satisfactory estimation accuracy and adaptation ability under an actual production environment.

Author Contributions: Conceptualization, X.G., B.L., and D.H.; formal analysis, D.H.; funding acquisition, B.L. and X.G.; methodology, D.H.; software, D.H. and D.Z.; writing—original draft, B.L. and D.H.; writing—review and editing, B.L., X.G., D.H. and D.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 61573088.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.
Data Availability Statement: Study data are available from the corresponding author on request.

Acknowledgments: The authors are grateful to ANSTEEL Group Corporation for offering us the experimental environment.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations
The following abbreviations are used in this manuscript:

- GAN: generative adversarial network
- CBR: case-based reasoning
- PPS: product particle size
- MIMO: multi-input–multi-output system
- RBR: rule-based reasoning
- SMOTE: synthetic minority oversampling technique
- KNN: k-nearest neighbor
- NSGA-II: nondominated sorting genetic algorithm II
- ReLU: rectified linear unit
- BP: back-propagation neural network
- SVM: support vector machines
- ELM: extreme learning machine
- MAE: mean absolute error
- MAPE: mean absolute percentage error
- RMSE: root mean square error

References
1. Lv, Z.; Liu, Y.; Zhao, J.; Wang, W. Soft computing for overflow particle size in grinding process based on hybrid case based reasoning. *Appl. Soft Comput.* 2015, 27, 533–542. [CrossRef]
2. Ramasamy, M.; Narayanan, S.; Rao, C. Control of ball mill grinding circuit using model predictive control scheme. *J. Process Control* 2005, 15, 273–283. [CrossRef]
3. Lu, X.; Krstić, M.; Chai, T.; Fu, J. Hardware-in-the-Loop Multiobjective Extremum-Seeking Control of Mineral Grinding. *IEEE Trans. Control Syst. Technol.* 2020, 29, 1–11. [CrossRef]
4. Botha, S.; Roux, J.; Craig, I. Hardware-in-the-Loop Multiobjective Extremum-Seeking Control of Mineral Grinding. *Miner. Eng.* 2018, 123, 49–62. [CrossRef]
5. Chen, X.; Li, Q.; Fei, S. Supervisory expert control for ball mill grinding circuits. *Expert Syst. Appl.* 2008, 34, 1877–1885. [CrossRef]
6. Zhou, P.; Chai, T.; Sun, J. Intelligence-Based Supervisory Control for Optimal Operation of a DCS-Controlled Grinding System. *IEEE Trans. Control Syst. Technol.* 2013, 21, 162–175. [CrossRef]
7. Zhao, D.; Chai, T. Intelligent optimal control system for ball mill grinding process. *J. Control Theory Appl.* 2013, 11, 454–462. [CrossRef]
8. Chai, T.; Ding, J.; Yu, G.; Wang, H. Integrated optimization for the automation systems of mineral processing. *IEEE Trans. Autom. Sci. Eng.* 2014, 11, 965–982. [CrossRef]
9. Cleary, P.; Sinnott, M.; Morrison, R. Prediction of slurry transport in SAG mills using SPH fluid flow in a dynamic DEM based porous media. *Miner. Eng.* 2006, 19, 1517–1527. [CrossRef]
10. Biau, X.; Wang, G.; Wang, H.; Wang, S.; Lv, W. Effect of lifters and mill speed on particle behaviour, torque, and power consumption of a tumbling ball mill: Experimental study and DEM simulation. *Miner. Eng.* 2017, 105, 22–35. [CrossRef]
11. Cleary, P. Prediction of coupled particle and fluid flows using DEM and SPH. *Miner. Eng.* 2015, 73, 85–99. [CrossRef]
12. Sinnott, M.; Cleary, P.; Morrison, R. Combined DEM and SPH simulation of overflow ball mill discharge and trommel flow. *Miner. Eng.* 2017, 108, 93–108. [CrossRef]
13. Mayank, K.; Malaehe, M.; Govender, L.; Mangadoddy, N. Coupled DEM-CFD model to predict the tumbling mill dynamics. *Procedia IUTAM* 2015, 15, 139–149. [CrossRef]
14. Mustapha, S. Case-based reasoning for identifying knowledge leader within online community. *Expert Syst. Appl.* 2018, 97, 244–252. [CrossRef]
15. Yan, A.; Shao, H.; Guo, Z. Weight optimization for case-based reasoning using membrane computing. *Inf. Sci.* 2014, 287, 109–120. [CrossRef]
16. Zhang, X.; Deng, Z.; An, W.; Cao, H. A methodology for contour error intelligent precompensation in cam grinding. *Int. J. Adv. Manuf. Technol.* 2013, 64, 165–170. [CrossRef]
17. Li, F.; Shang, C.; Shen, Q. Fuzzy knowledge-based prediction through weighted rule interpolation. *IEEE Trans. Cybern.* 2019, 50, 4508–4517. [CrossRef]
18. Shokouhyar, S.; Seifhashemi, S.; Siadat, H.; Ahmadi, M. Implementing a fuzzy expert system for ensuring information technology supply chain. Expert Syst. 2019, 36, 4508–4517. [CrossRef]
19. Hamedan, F.; Orooji, A.; Sanadgol, H.; Sheikhtaheri, A. Clinical decision support system to predict chronic kidney disease: A fuzzy expert system approach. Int. J. Med. Inf. 2020, 138, 104134. [CrossRef]
20. Hadizadeh, M.; Farzanegan, A.; Noaparast, M. Supervisory Fuzzy Expert Controller for Sag Mill Grinding Circuits: Sungun Copper Concentrator. Miner. Process. Extr. Metall. Rev. 2017, 38, 168–179. [CrossRef]
21. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. Adv. Neural Inf. Process. Syst. 2014, 3, 2672–2680. [CrossRef]
22. Shao, S.; Wang, P.; Yan, R. Generative adversarial networks for data augmentation in machine fault diagnosis. Comput. Ind. 2019, 106, 85–93. [CrossRef]
23. Wang, Z.; Wang, J.; Wang, Y. An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition. Neurocomputing 2018, 310, 213–222. [CrossRef]
24. Liu, H.; Zhou, J.; Xu, Y.; Zheng, Y.; Peng, X.; Jiang, W. Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks. Neurocomputing 2018, 315, 412–424. [CrossRef]
25. Liu, J.; Qu, F.; Hong, X.; Zhang, H. A small-sample wind turbine fault detection method with synthetic fault data using generative adversarial nets. IEEE Trans. Ind. Inf. 2018, 15, 3877–3888. [CrossRef]
26. Hua, Y.; Zhu, H.; Gao, M.; Zhang, H.; Ji, Z. Multi-Objective Optimization Design of Permanent Magnet Assisted Bearingless Synchronous Reluctance Motor Using NSGA-II. IEEE Trans. Ind. Electr. 2020. [CrossRef]
27. Orouskhani, M.; Shi, D.; Cheng, X. A Fuzzy Adaptive Dynamic NSGA-II With Fuzzy-Based Borda Ranking Method and its Application to Multimedia Data Analysis. IEEE Trans. Fuzzy Syst. 2020, 29, 118–128. [CrossRef]
28. Mohammadi, A.; Trovao, J.; Antunes, C. Component-level optimization of hybrid excitation synchronous machines for a specified hybridization ratio using NSGA-II. IEEE Trans. Energy Convers. 2020, 35, 1596–1605. [CrossRef]
29. Wang, J.; Sun, S. Optimized set-point model of grinding process based on case-based reasoning method. In Proceedings of the International Conference on System Science and Engineering, Maui, HA, USA, 4–7 January 2012; pp. 107–111.
30. Liu, B.; Liu, B.; Gao, X.; Zhang, D.; Hao, D.; Li, X. A soft sensor based on case-based reasoning for iron ores flotation. Ironmaking Steelmak. 2020, 47, 150–158. [CrossRef]