

## Article

# Long Short-Term Memory Parameter Optimization Based on Improved Sparrow Search Algorithm for Molten Iron Quality Prediction

Ziwen Zhang, Ruiyao Zhang and Ping Zhou \* 

State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China; 2102049@stu.neu.edu.cn (Z.Z.); 2190064@stu.neu.edu.cn (R.Z.)

\* Correspondence: zhouping@mail.neu.edu.cn

**Abstract:** Blast furnace (BF) ironmaking is a key process in iron and steel production. Because BF ironmaking is a dynamic time series process, it is more appropriate to use a recurrent neural network for modeling. The long short-term memory (LSTM) network is commonly used to model time series data. However, its model performance and generalization ability heavily depend on the parameter configuration. Therefore, it is necessary to study parameter optimization for the LSTM model. The sparrow search algorithm (SSA) holds advantages over traditional optimization algorithms in several aspects, such as no need for prior knowledge, fewer parameters, fast convergence, and high scalability. However, the algorithm still faces some challenges, such as the tendency to become trapped in the local optimum and the imbalance between global search ability and local search ability. Therefore, on the basis of SSA, this study examined the Levy flight strategy, sine search strategy, and step size factor adjustment strategy to improve it. This algorithm, improved by three strategies, is called the improved sparrow search algorithm (ISSA). Then, the ISSA-LSTM model was established. Furthermore, considering the limitations of SSA in dealing with multi-objective problems, the fast non-dominated sorting genetic algorithm (NSGAI) was introduced, and the ISSA-NSGAI model was established. Finally, experimental validation was performed using real blast furnace operation data, which demonstrated the proposed algorithm's superiority in parameter optimization for the LSTM model and prediction for real industrial data.



**Citation:** Zhang, Z.; Zhang, R.; Zhou, P. Long Short-Term Memory Parameter Optimization Based on Improved Sparrow Search Algorithm for Molten Iron Quality Prediction. *Metals* **2024**, *14*, 529. <https://doi.org/10.3390/met14050529>

Academic Editor: Henrik Saxen

Received: 17 March 2024

Revised: 26 April 2024

Accepted: 29 April 2024

Published: 30 April 2024

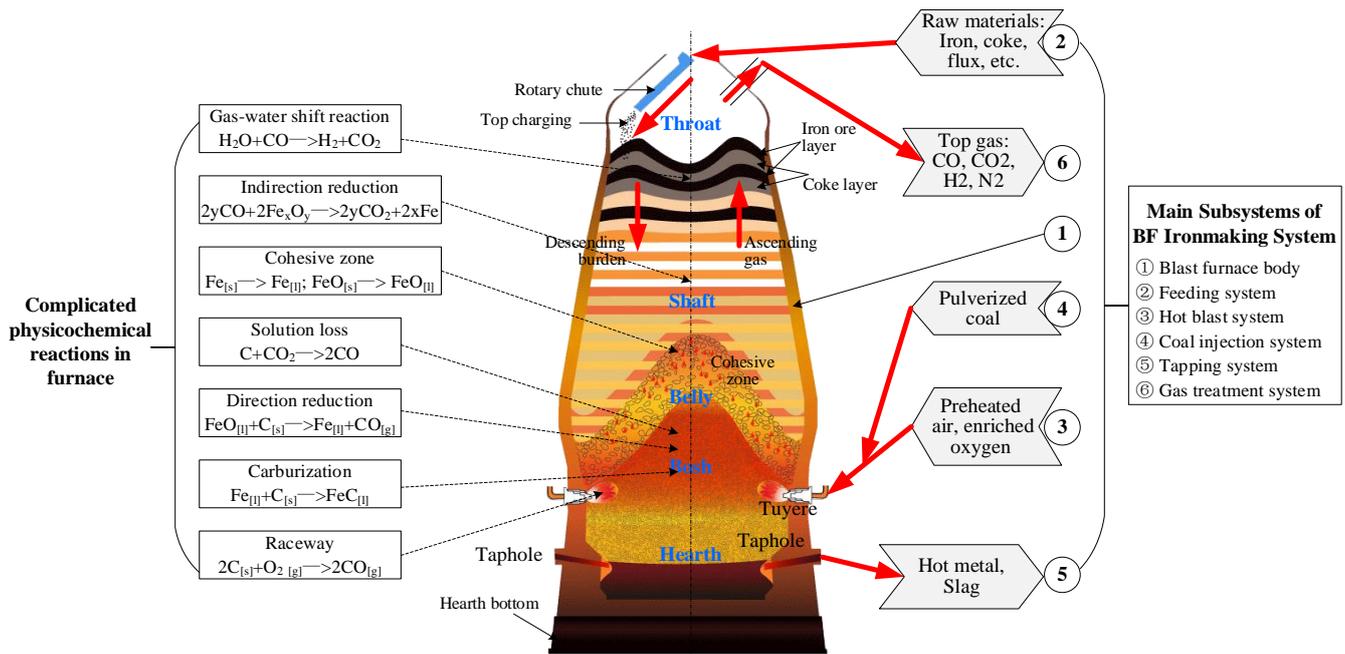


**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** LSTM; sparrow search algorithm; levy flight strategy; NSGAI; blast furnace ironmaking

## 1. Introduction

The blast furnace (BF) ironmaking system consists of several parts, such as the BF body, charging system, top gas treatment system, pulverized coal blowing system, hot air system, and iron discharge system. A typical BF ironmaking system is shown in Figure 1. During the process of BF ironmaking, the iron ore, basic fuel, and slag-forming additives, which are mixed in a specific proportion, are loaded from the top of the furnace and then move downward. At the same time, the hot air as well as pulverized coal are blown into the lower air outlet of the BF. A series of complex physicochemical reactions occur, and a significant amount of reducing gas is produced at high temperatures. After a series of processes, including heating, reduction, melting, slagging, carburizing, and desulfurization, the furnace burden finally generates liquid slag and pig iron. The slag and pig iron are separated by the iron discharge system. Then the qualified pig iron can be used for iron and steel production [1,2]. The physical and chemical reactions in BF ironmaking are complex, and the process is numerous. There are numerous parameters in the smelting process, including state parameters, control parameters, and molten iron quality (MIQ) parameters.



**Figure 1.** Schematic diagram of the blast furnace ironmaking system.

The MIQ indicators, which reflect the overall operational performance of the ironmaking process, are the focus of BF ironmaking modeling [3,4]. However, due to the inability of existing technology and measurement devices to directly obtain information about the MIQ in the BF, offline analysis of the molten iron is conducted after tapping, resulting in a lag between the detection and adjustment of the MIQ indicators. Additionally, with its time-varying and nonlinear characteristics, BF ironmaking is an extremely complex industrial process involving numerous physical and chemical reactions [5,6]. Thanks to the rapid advances in computer and sensor technology, as well as the widespread use of modern distributed control systems, a significant amount of data is stored in time series format. By analyzing this data, the variations and trends of the data can be revealed, providing powerful support for predicting future trends and making decisions. Therefore, how to mine useful information from massive time series data has become an important issue. As artificial intelligence technology continues to develop, a variety of machine learning algorithms have been applied to time series analysis. Common methods include support vector machines, artificial neural networks, decision trees, and random forests [7–9]. Jian [10] used a support vector machine to classify the silicon content of molten iron into multiple categories, which improved the accuracy of predicting the silicon content. Chen et al. [11] used a backpropagation (BP) neural network to establish a prediction model of the silicon content and achieved good prediction results. Zhang [12] introduced the autoencoder and principal component analysis into the conventional random vector functional link network (RVFLNs), which enhanced the computational speed and accuracy of the MIQ model. Lv [13] improved the robustness and adaptive capability of conventional RVFLNs based on the theory of robust estimation and the online sequential learning technique and used the improved model to predict the silicon content and molten iron temperature. Dai [14] proposed a subspace identification algorithm based on a bilinear system to establish a model of MIQ indicators, which was applied in the nonlinear adaptive predictive control of MIQ indicators. Zhou [15] proposed a robust least-squares support vector machine model based on a nonlinear autoregressive model to simultaneously predict multiple MIQ indicators. These methods mainly model data through feature extraction and model training, ignoring the dynamics of time series data as well as dependencies among the series.

Recurrent neural networks (RNNs) were introduced as a solution to address these problems, which originated from the Hopfield network proposed by Hopfield in 1982 [16].

At each time step, RNN receives the current inputs along with the hidden state from the previous time step. It uses activation functions and weight matrices to transmit and update information over time. However, RNN suffers from problems such as vanishing gradients and exploding gradients, making it difficult to converge during training [17,18]. Hochreiter and Schmidhuber [19] introduced long-term memory and gating mechanisms into RNN to create the long short-term memory (LSTM) network. LSTM has a stronger memory capacity and is good at capturing long-term dependencies, which solves the problems of traditional RNN in dealing with long sequence data. Consequently, LSTM has been generally applied to the modeling of time series data. However, the performance and generalization ability of the LSTM model heavily depend on its parameter configuration [20], so research on optimizing the parameters of the LSTM model has become an important direction for improving its performance.

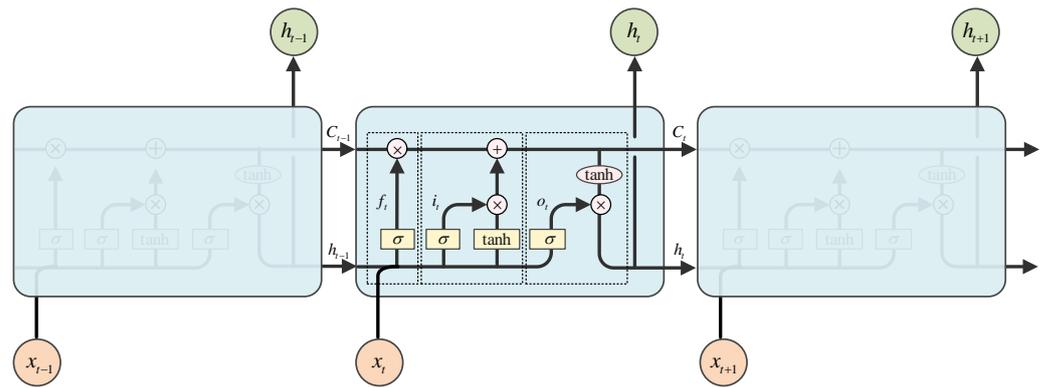
Previous studies have proposed various methods for LSTM hyperparameter optimization. Bengio et al. [21] and Abbasimehr et al. [22] used random search and grid search to find the optimal hyperparameter configuration, respectively. However, these methods have the problems of high computational costs, the tendency to fall into the local optimum, and the inability to deal with dependencies among the hyperparameters. Snoek et al. [23] proposed a method based on Bayesian optimization that models the relationship between the hyperparameters and performance through the Gaussian process model. This method selects the next set of hyperparameters according to the continuously updated Gaussian process model. Although this method is superior to traditional random search and grid search in terms of performance, it relies on prior knowledge and typically assumes that the hyperparameter space is continuous. If the prior knowledge is chosen inappropriately or the hyperparameter space is discrete, it may lead to biased search results. Meanwhile, Gorgolis et al. [24] used the genetic algorithm to simulate the process of selecting and mutating biological evolution to quickly discover the optimal hyperparameter configuration in the predefined search space. Although this method is superior to the traditional methods of randomly generating models in terms of performance, it still suffers from high computational cost, complex parameter selection, and the risk of getting trapped in the local optimum.

Recently, a novel swarm intelligence algorithm called the sparrow search algorithm (SSA) was proposed by Xue et al. [25]. SSA performs local and global searches by imitating the foraging and anti-predation behaviors of sparrows. Compared to traditional optimization algorithms, SSA does not require prior knowledge. It has faster convergence speed, fewer parameters, and strong scalability, which can effectively avoid certain problems, such as vanishing gradients. However, despite the several advantages of SSA, further research has revealed some shortcomings of this algorithm, including the tendency to become trapped in the local optimum and the imbalance between global search and local search ability. To overcome these challenges, this paper proposes an improved sparrow search algorithm (ISSA). In this paper, three strategies, namely, the Levy flight strategy, sine search strategy, and step size factor adjustment strategy, are introduced to improve SSA. Furthermore, ISSA is combined with the fast non-dominated sorting genetic algorithm (NSGAI) to establish the ISSA-NSGAI model. Finally, real operation data of BF is used to verify the superior optimization performance of the proposed algorithm.

## 2. Basic Algorithm

### 2.1. Long Short-Term Memory Networks

Unlike other conventional neural networks, RNN is able to handle data with sequence variations. In order to overcome the problems of vanishing gradients and exploding gradients that occur when training long sequence data, scholars have proposed LSTM, which is a special kind of RNN. It can selectively retain and forget information through the introduction of the forget gate, the input gate, the output gate, and the memory cell. This enables the LSTM network to capture long-term dependencies in sequential data, thus making the network more efficient. The structure diagram of LSTM is shown in Figure 2.



**Figure 2.** Schematic diagram of the LSTM structure.

Based on the current input  $x_t$  and the previous hidden state  $h_{t-1}$ , the update formula of the LSTM network is represented as

$$\begin{cases} f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ a_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ C_t = f_t \cdot C_{t-1} + i_t \cdot a_t \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t \cdot \tanh(C_t) \end{cases} \quad (1)$$

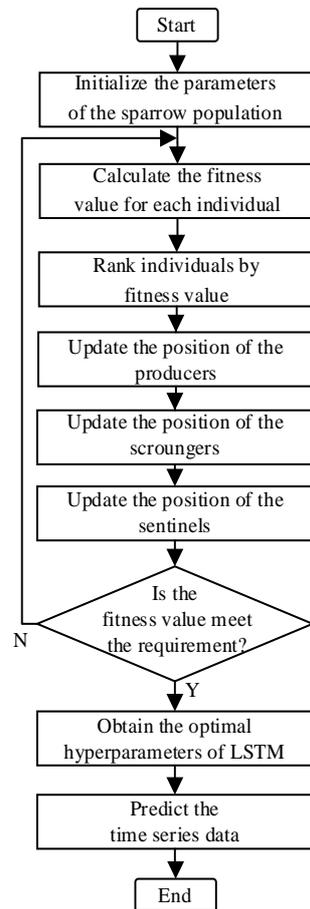
where  $f$ ,  $i$ , and  $o$  denote the forget gate, the input gate, and the output gate, respectively;  $C$  denotes the unit state of the network;  $h_t$  and  $h_{t-1}$  denote the hidden state at moment  $t$  and  $t - 1$ , respectively;  $W$  and  $b$  are the weight and bias matrices, respectively;  $\sigma$  denotes the sigmoid activation function; and  $\tanh$  is a hyperbolic tangent activation function.

The prediction quality of the LSTM model is directly influenced by the selection and adjustment of hyperparameters [26]. Some of the key hyperparameters include the number of layers of the model, the number of nodes in the hidden layer, the learning rate, and the number of iterations. If a model has too many layers, it may lead to gradient problems, which will affect the model's training. Increasing the number of nodes in the hidden layers can improve the expressiveness of the network, but it also increases the complexity of the model, which may lead to overfitting and longer training times. If the learning rate is set too high or too low, convergence will be affected. Increasing the number of iterations can better fit the data and improve the prediction performance of the network. However, too many iterations can lead to overfitting and reduce the ability of the model to generalize new data. The choice of these hyperparameters significantly affects the performance of the LSTM model. Generally, there are multiple hyperparameters that need to be adjusted. Therefore, the optimization of hyperparameters is a composite problem. Traditional methods rely on continuous observation of changes in the loss function to comprehensively consider hyperparameters, making it difficult to find the optimal hyperparameter configuration. Therefore, research on the optimization of LSTM hyperparameters has become important to improve the performance of the LSTM model.

## 2.2. Sparrow Search Algorithm

SSA uses the overall characteristics of the sparrow population and the characteristics of each individual to establish a mathematical model. The sparrows are divided into producers and scroungers. The sparrows that are able to find better sources of food are producers, while the rest are scroungers. The proportion of producers and scroungers in the total population is constant, and their roles change dynamically [27,28]. The fitness value reflects the level of energy reserves of the individual sparrow. In addition, the sparrows, which make up 10% to 20% of the population, are selected so that they can be aware of the danger. Once they detect the danger, they will immediately give up their food. They

are generally located at the periphery of the group [29]. The process of SSA involves initializing the parameters of the sparrow population, calculating the fitness value of each individual and determining the maximum and minimum values, updating the position of producers, updating the position of scroungers, updating the position according to the alert, determining whether to continue or terminate the loop based on whether the fitness values meet the requirements, obtaining the optimal hyperparameters of LSTM, and predicting the time series data. The flow chart of the SSA-LSTM prediction model is shown in Figure 3.



**Figure 3.** Flow chart of the SSA-LSTM prediction model.

1. The formula for updating the position of the producer is described as

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot T_{\max}}\right), & R_2 < ST \\ x_{i,j}^t + Q \cdot L, & R_2 \geq ST \end{cases} \quad (2)$$

where  $t$  denotes the current iteration;  $T_{\max}$  is a constant, which denotes the maximum number of iterations;  $x_{i,j}^t$  denotes the value of the  $j$ -th dimension of the  $i$ -th sparrow at iteration  $t$ ;  $\alpha \in (0, 1]$  is a random number;  $R_2 \in [0, 1]$  denotes the alarm value;  $ST \in [0.5, 1]$  denotes the safety threshold;  $Q$  is a random number that obeys a normal distribution; and  $L$  denotes a matrix of  $1 \times d$  whose elements inside are all 1. When  $R_2 \geq ST$ , it indicates that there are no predators around the foraging area, so the producers can conduct a wide search. When  $R_2 < ST$ , it indicates that some sparrows have detected the predators in the foraging area and alerted the other sparrows in the population, at which point all sparrows need to quickly fly to other safe areas to forage.

- The formula for updating the position of the scrounger is described as

$$x_{ij}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{x_{\text{worst}}^t - x_{ij}^t}{i^2}\right), & i > n/2 \\ x_p^{t+1} + |x_{ij}^t - x_p^{t+1}| \cdot A^+ \cdot L, & i < n/2 \end{cases} \quad (3)$$

where  $x_p^{t+1}$  denotes the optimal position occupied by the producer of generation  $t + 1$ ;  $x_{\text{worst}}$  denotes the current position occupied by the global worst individual;  $A$  denotes a matrix of  $1 \times d$  whose elements inside are randomly assigned 1 or  $-1$ ; and  $A^+ = A^T(AA^T)^{-1}$ . When  $i > n/2$ , it indicates that the  $i$ -th scrounger with the worst fitness value does not find food and needs to go to other areas for food.

- The sparrows that sense danger are called sentinels. The formula for updating the position of the sentinel is described as

$$x_{ij}^{t+1} = \begin{cases} x_{\text{best}}^t + \beta \cdot |x_{ij}^t - x_{\text{best}}^t|, & f_i > f_g \\ x_{ij}^t + K \cdot \left(\frac{|x_{ij}^t - x_{\text{worst}}^t|}{(f_i - f_w) + \varepsilon}\right), & f_i = f_g \\ x_{ij}^t, & f_i < f_g \end{cases} \quad (4)$$

where  $x_{\text{best}}^t$  denotes the current global optimal position;  $\beta \in (-1, 1)$  denotes the step size control parameter of the sentinel position update, which is a random number obeying a normal distribution with a mean value of 0 and a variance of 1;  $K \in [-1, 1]$ , as the step size control parameter, is a random number indicating the direction in which the sentinel is moving;  $f_i$  is the fitness value of the present sparrow;  $f_g$  and  $f_w$  are the current global best and worst fitness values, respectively; and  $\varepsilon$  is the smallest constant to avoid the denominator being zero. When  $f_i > f_g$ , it indicates that the individual sparrow is at the periphery of the group, and it needs to move to obtain a higher fitness value. When  $f_i = f_g$ , it indicates that the sparrows in the center of the group sense danger and must move towards other sparrows to avoid the danger of predation.

### 2.3. Non-Dominated Sorting Genetic Algorithm II

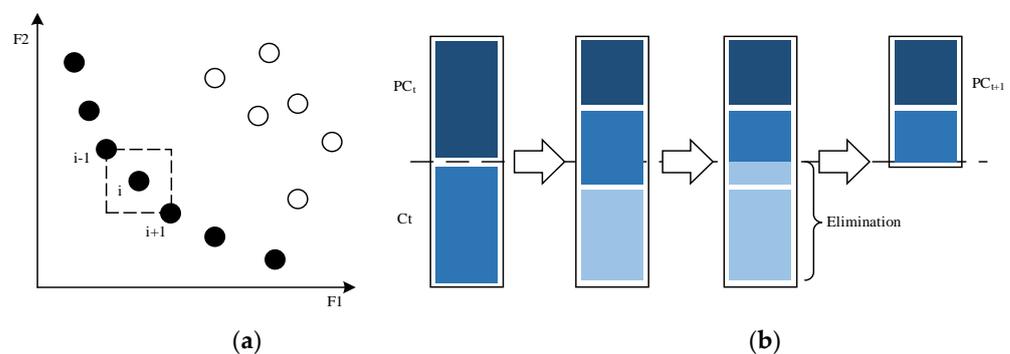
NSGAI is a typical multi-objective optimization algorithm for the solution of optimization problems with multiple conflicting objectives [30]. This algorithm simulates the process of natural evolution. The individual in the solution space is evolved and selected by genetic operators to generate a set of non-dominated solutions. The NSGAI algorithm introduces mechanisms including fast non-dominated sorting, crowding distance, and elitism strategy to maintain the diversity and convergence of the population [31].

- Population initialization: Two random generation methods, uniform distribution and Gaussian distribution, are used to generate the population. Each individual in the population is then initially evaluated. The feasibility of constraints and solutions is considered to ensure that the generated population meets the requirements of the problems.
- Fast non-dominated sorting: Each solution is assigned two attributes:  $np$  represents the number of dominant solutions  $p$ , and  $Sp$  represents the set of solutions dominated by solution  $p$ .
- Calculation of crowding distance: The crowding distance is used to measure the diversity of populations. The schematic diagram of crowding distance is shown in Figure 4a. The  $x$ -axis of the graph represents the objective function, and the  $y$ -axis represents the crowding distance. A point on the graph represents each candidate solution. The candidate solution that is closer to the left side of the graph indicates better performance on the objective function. The candidate solution that is closer to the top of the graph indicates a sparser distribution within the entire solution set, with a higher crowding distance and greater diversity. The density between candidate

solutions can be measured by the distance on the  $x$ -axis. Solutions that are closer together in terms of the  $x$ -axis distance suggest that they have similar performance on the objective function. The crowding distance of the individual in the population is calculated as

$$d_i = \sum_{k=1}^M |f_k(x_{i-1}) - f_k(x_{i+1})| \quad (5)$$

where  $d_i$  denotes the crowding distance of individual  $x_i$ , and  $f_k(x_{i+1})$  denotes the  $k$ -th objective function of individual  $x_{i+1}$ . The crowding distance between the first and last individuals is expressed as  $\infty$ . After performing fast non-dominated sorting and calculating the crowding distance, each individual in the population obtains two attributes: non-domination rank  $nrank$  and crowding distance  $d$ . With these two attributes, the dominant relationship between any two individuals in the population can be determined.



**Figure 4.** (a) Schematic diagram of crowding distance; (b) schematic diagram of the elitism strategy.

4. Elitism strategy: To prevent the loss of good individuals, the elitism strategy retains the good individuals from the parent generation directly into the offspring. Fast non-dominated sorting and the calculation of the crowding distance are used to sort the individuals in the parent generation, the offspring generation, and the synthetic populations so that the next generation can be selected. The schematic diagram of the elitism strategy is shown in Figure 4b.
5. Selection, crossover, and mutation: A binary tournament selection strategy is used to simulate binary crossover and polynomial mutation. Specifically, two individuals are randomly selected from the population in each iteration, and the better of them is chosen to join the offspring population until the new population size reaches the original population size.
6. Termination condition: During the iteration process, if the iteration reaches the maximum value, the algorithm will be stopped. Otherwise, it continues to repeat the above steps.

### 3. Improved Sparrow Search Algorithm: Non-Dominated Sorting Genetic Algorithm II

Although SSA performs better than the traditional optimization algorithm, it also has certain limitations. First, the algorithm may be trapped in the local optimum. Second, in the SSA, the positions of producers and scroungers are updated based on unchanged update formulas, which means they cannot be updated by more suitable methods. Additionally, the step size control parameters  $\beta$  and  $K$  are generated randomly and cannot be adjusted according to the actual situation of the search, which may lead to the algorithm falling into the local optimum. To address these problems, this paper introduces the Levy flight strategy, sine search strategy, and step size factor adjustment strategy to improve the basic SSA. Furthermore, considering the excellent performance of NSGAI in dealing with multi-objective problems, this paper combines NSGAI with ISSA to construct the NSGAI-ISSA model. These improvements and combinations are designed to further enhance the performance of SSA, particularly in dealing with complex problems.

### 3.1. Improved Sparrow Search Algorithm

Aiming at the weaknesses of SSA, including the tendency to fall into the local optimum, the fixed position update formulas, and the inability to adjust step size control parameters appropriately, three strategies are used for improvement.

1. Levy flight strategy: The Levy flight strategy is a random behavior strategy. Based on the Levy distribution, it simulates long-distance and random flight behavior. It is a kind of random walk with occasional large steps. In optimization, this characteristic of Levy flight enables a potentially stuck algorithm to escape from a local optimum and restart the search in a different region of the search space [32]. The formula for the Levy flight strategy is

$$\text{Levy}(d) = 0.01 \cdot \frac{r_1 \cdot \sigma}{|r_2|^{1/\beta}} \tag{6}$$

$$\sigma = \left\langle \frac{\Gamma(1 + \theta) \cdot \sin(\pi\theta/2)}{\Gamma[(1 + \lambda)/2] \cdot \theta \cdot 2^{(\theta-1)/2}} \right\rangle^{1/\theta} \tag{7}$$

where  $\Gamma$  denotes the gamma function;  $\theta$  is a constant; and  $r_1 \in [0, 1]$  and  $r_2 \in [0, 1]$  are random numbers.

The Levy flight strategy is introduced into (4) so that the position of the sentinels can be updated according to the distance between the current position and the optimal position of the sparrows, which reduces the risk of the sparrow becoming trapped in a local optimum. It allows the algorithm to effectively perform local searches over short distances and also fully perform global searches over long distances. The improved formula for updating the position of the sentinel is

$$x_{i,j}^{t+1} = \begin{cases} \text{Levy}(d) \cdot x_{\text{best}}^t + \beta \cdot |x_{i,j}^t - \text{Levy}(d) \cdot x_{\text{best}}^t|, & f_i > f_g \\ x_{i,j}^t + K \cdot \left( \frac{|x_{i,j}^t - x_{\text{worst}}^t|}{(f_i - f_w) + \epsilon} \right), & f_i = f_g \\ x_{i,j}^t, & f_i < f_g \end{cases} \tag{8}$$

2. Sine search strategy: The sine search strategy simulates the oscillation process of the sine and cosine functions [33]. By adjusting the oscillation parameters to control the step size and direction of the search process, the global search and the local search are balanced until the solution space gradually converges to the optimal solution. The sine search strategy is introduced into (2) and (3), which are the formulas for updating the positions of the producer and the scrounger, so that the individual sparrows can be given different weight values according to their different positions. When the fitness value of the individual approaches the optimal fitness value  $f_{\text{best}}$ , the weight  $\omega$  is small, and the algorithm continues to search in the interval near the position of the current individual. When the fitness value of the individual approaches the worst fitness value  $f_{\text{worst}}$ , the weight  $\omega$  increases to  $\omega_{\text{max}}$ , and the algorithm begins to search in the interval far from the position of the current individual. Thanks to this strategy, individuals with higher fitness values in the population can search near their current positions, so the local search ability is enhanced. On the other hand, individuals with lower fitness values in the population can explore away from their own positions, thus improving global search ability. The formula for the sine search strategy is

$$\omega = \omega_{\text{min}} + (\omega_{\text{max}} - \omega_{\text{min}}) \times \left( \sin\left(\left(\frac{f_i^t - f_{\text{best}}^t}{f_{\text{worst}}^t - f_{\text{best}}^t} + 1\right) \times \frac{\pi}{2} + \pi\right) + 1 \right) \tag{9}$$

where  $\omega_{\text{min}}$  and  $\omega_{\text{max}}$  denote the minimum and maximum values of the weight range, respectively;  $f_i^t$  denotes the fitness value of the  $i$ -th sparrow in the population at iteration  $t$ ; and  $f_{\text{best}}^t$  and  $f_{\text{worst}}^t$  denote the best and worst fitness values in the population at iteration  $t$ , respectively. Figure 5 is the variation law of adaptive weight  $\omega$ .

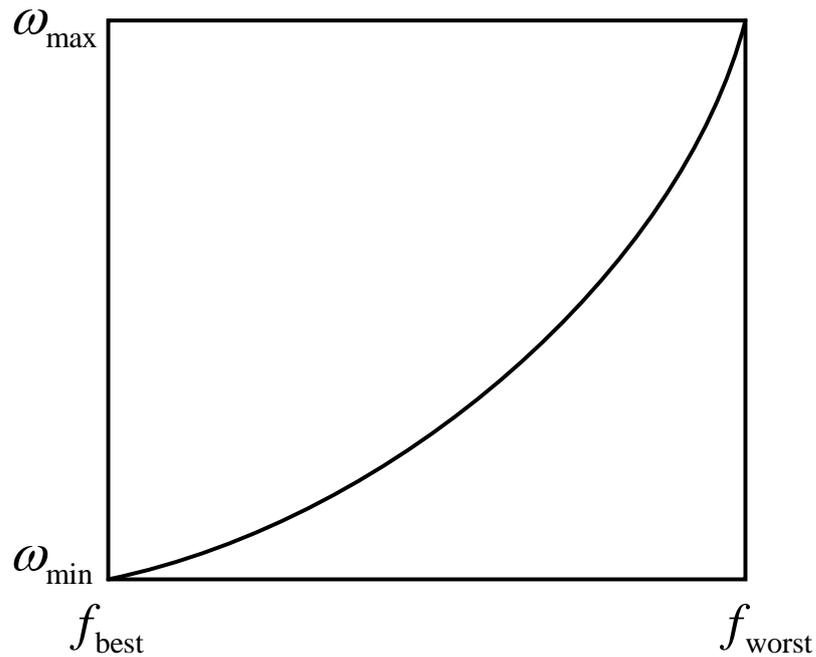


Figure 5. Variation law of adaptive weight  $\omega$ .

The sine search strategy is incorporated into the position update formula of SSA. The improved formulas for updating the position are

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t \cdot \exp\left(\frac{-1}{\alpha \cdot T_{\max}}\right), & R_2 < ST \\ x_{ij}^t + w \cdot Q \cdot L, & R_2 \geq ST \end{cases} \quad (10)$$

$$x_{ij}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{x_{\text{worst}}^t - x_{ij}^t}{i^2}\right), & i > n/2 \\ x_P^{t+1} + w \cdot |x_{ij}^t - x_P^{t+1}| \cdot A^+ \cdot L, & i \leq n/2 \end{cases} \quad (11)$$

$$x_{ij}^{t+1} = \begin{cases} x_{\text{best}}^t + w \cdot \beta |x_{ij}^t - x_{\text{best}}^t|, & f_i > f_g \\ x_{ij}^t + w \cdot K \frac{|x_{ij}^t - x_{\text{best}}^t|}{(f_i - f_w) + \epsilon}, & f_i = f_g \\ x_{ij}^t, & f_i < f_g \end{cases} \quad (12)$$

- Step size factor adjustment strategy: The step size factor adjustment strategy dynamically adjusts the step size during different stages of the search by selecting appropriate adaptive factors with their special mathematical characteristics [34]. This achieves a balance between the local search and the global search. The step size control parameters  $\beta$  and  $K$  in the (4) are improved as

$$\beta = \text{fitness}_{\text{best}} - (\text{fitness}_{\text{best}} - \text{fitness}_{\text{worst}}) \cdot \left(\frac{T-t}{T}\right)^{1.5} \quad (13)$$

$$K = (\text{fitness}_{\text{best}} - \text{fitness}_{\text{worst}}) \cdot \exp\left(-20 \cdot \tan\left(\frac{t}{T}\right)^z\right) \cdot (2 \cdot \text{rand} - 1) \quad (14)$$

where  $\text{fitness}_{\text{best}}$  and  $\text{fitness}_{\text{worst}}$  denote the best fitness value and the worst fitness value, respectively, and  $T$  denotes the maximum number of iterations.

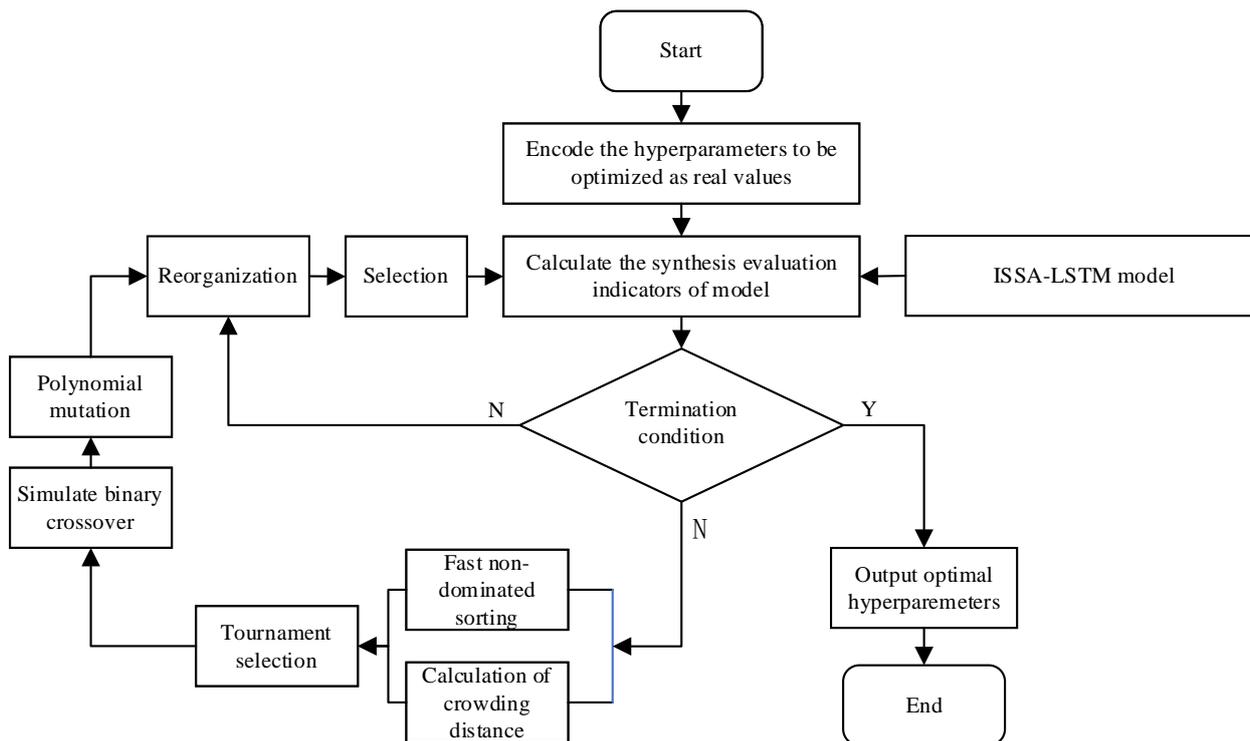
From (13), it can be seen that the improved step size control parameter  $\beta$  varies nonlinearly and incrementally. In the early iterations of SSA, the population has high diversity, and the algorithm has a great ability to search the global space but a weak

ability to explore the local space. Therefore, the control parameter  $\beta$  is set to a small value to enhance the local search ability. On the other hand, at the later iterations of SSA, all individual sparrows are attracted by the current global optimum, and there is not enough space to search, which may lead to premature convergence. Therefore, the control parameter  $\beta$  is set to a larger value in order to avoid becoming trapped in the local optimum. As can be seen from (14), the improved step size control parameter  $K$  increases first and then decreases, which promotes SSA to thoroughly explore the search space in the early iterations and improve convergence speed in the later iterations. By introducing the step size factor adjustment strategy, the global search and local search abilities of SSA are balanced. The optimization accuracy is improved while preventing the algorithm from becoming trapped in the local optimum.

### 3.2. ISSA-NSGAI Algorithm

SSA was originally designed for single-objective optimization problems and has no ability to handle multi-objective problems directly. By introducing NSGAI, the capability of the algorithm can be extended to handle multi-objective problems, and a solution set with better diversity and balance can be found. The flow chart of ISSA-NSGAI is shown in Figure 6. The process for ISSA-NSGAI is as follows:

1. Define the hyperparameter space: The range of hyperparameters is determined for the LSTM network.
2. Initialize the sparrow population: The random initialization method of SSA is used to generate the initial sparrow population, where each sparrow represents a hyperparameter configuration.
3. Evaluate the fitness value of the sparrow: For each sparrow, the LSTM network is trained using the training set divided by cross-validation, and its performance is evaluated on the validation set. According to the specific indicators of the problem, such as accuracy and loss function, the fitness value of each sparrow is calculated.
4. Fast non-dominated sorting and calculation of crowding distance: NSGAI is used to perform fast non-dominated sorting on the sparrow population. The sparrows are divided into different levels, and the crowding distance of each sparrow on the Pareto front is calculated.
5. Selection: Based on fast non-dominated sorting and crowding distance, the elitism strategy is used to select sparrows with higher fitness values in the population as parents for the next generation.
6. Genetic operation: Crossover and mutation operations are performed on selected parent sparrows to generate the next generation of sparrow populations. The crossover operation is performed using the cross strategy of the NSGAI, which can be adapted according to the characteristics of the hyperparameters.
7. Updated sparrow population: The newly generated sparrow population is merged with the original population to form an updated sparrow population.
8. Termination condition: According to the predefined termination condition, such as the maximum number of iterations or the threshold of the fitness value, it is determined whether to terminate the optimization process. If the termination condition is not satisfied, return to step 4.
9. Output optimal hyperparameters: At the end of the optimization, the sparrow with the best fitness value, which represents the optimal hyperparameters, is selected from the final sparrow population as the best configuration for the LSTM network.



**Figure 6.** Flow chart of the ISSA-NSGAI algorithm.

#### 4. Experimental Results

To verify the effectiveness of the improved strategies and the excellent performance of the improved algorithm, the SSA, ISSA, and ISSA-NSGAI algorithms were tested under the same conditions. For the tests, this study focused on several evaluation indicators of the intelligent population optimization algorithm, including convergence time, evolutionary algebra, and global search ability. However, it should be noted that these evaluation indicators usually cannot be optimized at the same time because of the trade-off relationship between them. Usually, as the population size increases, the change in direction of each evaluation indicator will be contradictory. In order to accelerate the convergence speed of the algorithm on the basis of improving global search ability, the population size of all algorithms was set to 10 to ensure a fair performance comparison under the same population size. This setting can better test the performance of the algorithm in different aspects and help find the algorithm that is most suitable for the specific problems.

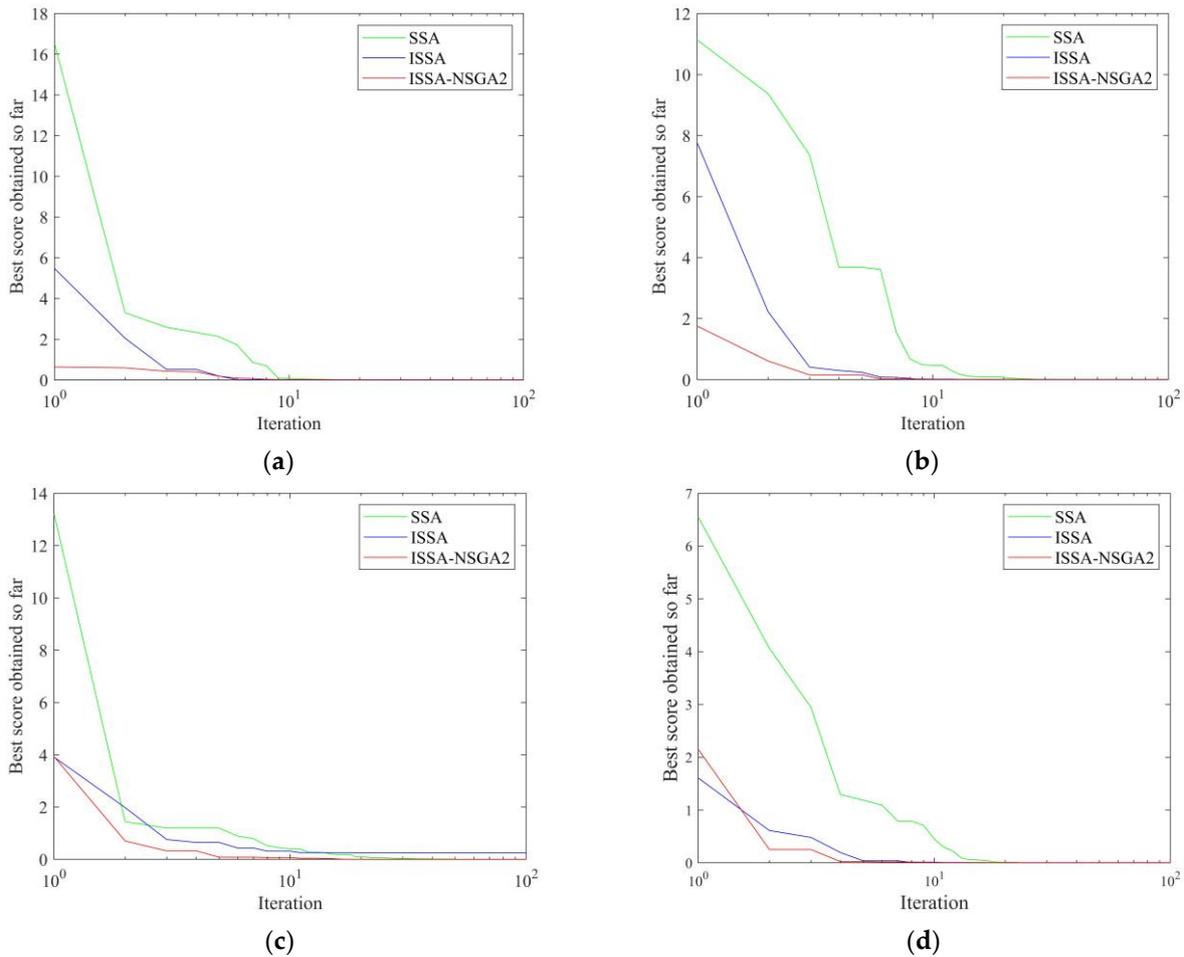
##### 4.1. Benchmark Function

We selected the four benchmark functions in Table 1. Among them, ZDT1 is a multi-dimensional unimodal flat-bottom function with random interference. ZDT2 is a multimodal test function. The distance between the global minimum point and the next local minimum point of ZDT2 is a geometric distance, which is deceptive, and may cause the algorithm to converge in the wrong direction. ZDT3 is also a multimodal test function and has a large search space and many local minima. ZDT4 is a test function for global optimization. It has six local minima, two of which are global minima. Through these four benchmark functions, the SSA, ISSA, and ISSA-NSGAI algorithms were tested in terms of anti-interference, the ability to global search, and so on.

Considering that the randomness of the initial position of the population will affect the optimization process, the average values of 25 tests were taken as the final results to avoid contingencies and make the test results more authentic. The test results of the benchmark functions iterated 500 times are shown in Figure 7.

**Table 1.** Benchmark function.

Number	Benchmark Function
ZDT1	$\sum_{i=1}^n i \times x_i^4 + random[0,1)$
ZDT2	$\sum_{i=1}^n -x_i \times \sin(\sqrt{ x_i })$
ZDT3	$\sum_{i=1}^n [x_i^2 - 10 \cos 2\pi x_i + 10]$
ZDT4	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$



**Figure 7.** Test results of (a) ZDT1; (b) ZDT2; (c) ZDT3; (d) ZDT4.

The performance of the algorithms was tested under the same parameters, iterations, and test environment. Because the initial value was randomly selected, the results of each algorithm at the first iteration are different. Specifically, thanks to the more homogenous initial distribution of the population, the initial value of ISSA-NSGAII is the closest to the optimal feasible solution. It can be seen from Figure 7a that ISSA-NSGAII had strong anti-interference performance under the ZDT1 with fast convergence speed and a smooth convergence curve. The performance of ISSA is poor, and the performance of SSA is the worst. It can be seen in Figure 7b that ISSA and ISSA-NSGAII converge at fewer than 10 iterations, while SSA converges at more than 10 iterations. It can be seen from Figure 7c that ISSA-NSGAII converges to the optimal value the fastest. ISSA falls into the local optimum, and SSA converges to the optimal value more slowly. As shown in Figure 7d, ISSA-NSGAII converges the fastest. ISSA converges more slowly, and SSA converges at more than 10 iterations. To summarize, ISSA-NSGAII converges the most smoothly without obvious jump changes. Its advantages

are mainly reflected in its convergence speed, anti-interference ability, and high global search ability. ISSA has a slightly weaker performance, and SSA has the worst performance.

In order to evaluate the performance of the multi-objective algorithm, inverted generational distance (IGD) was chosen as the performance indicator of the algorithm, which represents the average value of the distance from each reference point to the nearest solution [35]. IGD can reflect the convergence of the algorithm. The smaller the IGD, the more the algorithm converges to the true Pareto frontier and the better the comprehensive performance of the algorithm. The calculation formula of IGD is

$$\text{IGD}(P, Q) = \frac{\sum_{v \in P} d(v, Q)}{|P|} \quad (15)$$

where  $P$  denotes the set of points distributed on the true Pareto frontier;  $|P|$  denotes the number of points in  $P$ ;  $Q$  denotes the set of Pareto optimal solutions solved by the algorithm; and  $d(v, Q)$  denotes the minimum Euclidean distance from  $v$  to the points in  $Q$ .

The IGD values of each algorithm under the benchmark function are shown in Table 2. The mean values of SSA under ZDT1, ZDT2, ZDT3, and ZDT4 are 1.5364, 0.8428, 51.364, and 40.309, respectively. The mean values of ISSA are 1.7228, 0.7523, 0.8152, and 40.211. The mean values of ISSA-NSGAI are 0.1971, 0.1524, 0.03242, and 40.024, which are the optimal values. This shows that ISSA-NSGAI is better in terms of search performance and stability compared to ISSA and SSA under the same experimental conditions.

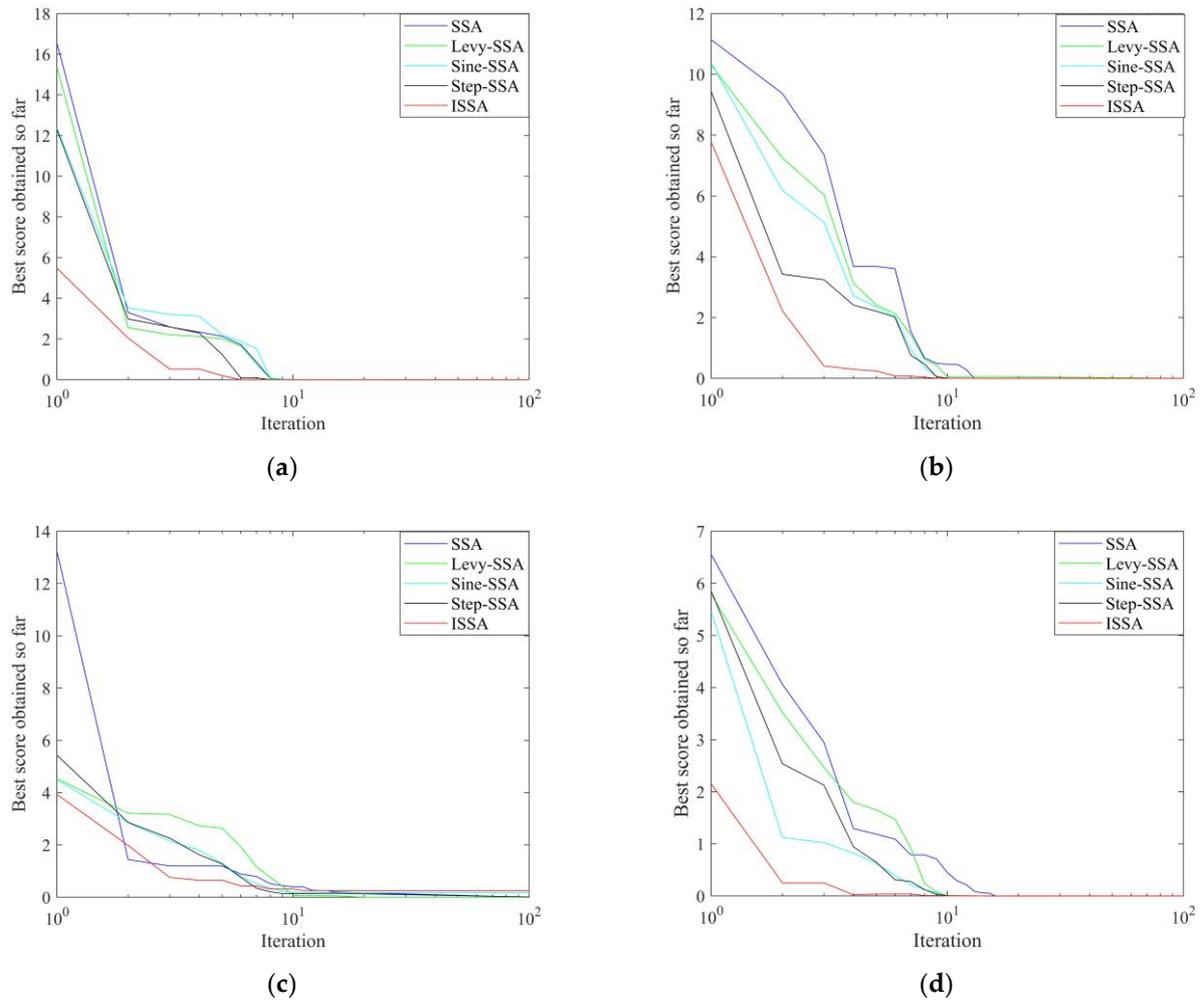
**Table 2.** IGD of each algorithm under the benchmark function.

Benchmark Function		SSA	ISSA	ISSA-NSGAI
ZDT1	Mean value	1.5364	1.3228	0.1971
	Standard deviation	0.06954	0.0551	0.0375
ZDT2	Mean value	0.8428	0.7523	0.1524
	Standard deviation	0.7628	0.5241	0.03241
ZDT3	Mean value	51.364	47.742	42.284
	Standard deviation	72.653	53.651	52.413
ZDT4	Mean value	40.309	40.211	40.024
	Standard deviation	65.577	65.354	65.586

#### 4.2. Ablation Experiment

In order to demonstrate the effectiveness of the three improvement strategies for SSA, the benchmark functions ZDT1, ZDT2, ZDT3, and ZDT4 were used to test the five algorithms: SSA, SSA improved by the Levy flight strategy (Levy-SSA), SSA improved by the sine search strategy (Sine-SSA), SSA improved by the step size factor adjustment strategy (Step-SSA), and SSA improved by the combination of the three strategies (ISSA). The results of ablation experiments are shown in Figure 8.

The results show that ISSA converges the fastest and most smoothly. The IGD values of each improvement strategy under the benchmark functions are shown in Table 3. Under the same experimental conditions, it is evident that the IGD values of ISSA are all the optimal values. This indicates that ISSA, which is improved by the three improvement strategies, has better optimization performance than the algorithms improved by a single strategy. Additionally, it also indicates that all three strategies can improve the SSA.



**Figure 8.** Results of (a) ZDT1 ablation experiment; (b) ZDT2 ablation experiment; (c) ZDT3 ablation experiment; (d) ZDT4 ablation experiment.

**Table 3.** IGD of each improvement strategy under the benchmark function.

Benchmark Function		SSA	Levy-SSA	Sine-SSA	Step-SSA	ISSA
ZDT1	Mean value	1.5364	1.5653	1.5376	1.6638	1.3228
	Standard deviation	0.06954	0.04459	0.06975	0.05743	0.0551
ZDT2	Mean value	0.8428	0.7579	0.7794	0.8281	0.7523
	Standard deviation	0.7628	0.7334	0.6461	0.6983	0.5241
ZDT3	Mean value	51.364	50.685	51.587	51.257	47.742
	Standard deviation	72.653	67.972	66.221	65.785	53.651
ZDT4	Mean value	40.309	40.163	40.272	40.046	40.211
	Standard deviation	65.577	65.415	65.569	65.422	65.354

### 4.3. Experiments Based on Real Industrial Operation Data

At the BF ironmaking site, the operation status of the BF is monitored in real time, and the operation data is recorded by the information management system. Some BF parameters can be directly measured by the monitoring devices, and some BF parameters need to be calculated by relevant formulas based on the directly measured parameters. Additionally, some BF parameters require offline analysis before being imported into the information management system, such as silicon content ([Si]), sulfur content ([S]), and phosphorus content ([P]), which are all MIQ indicators. Considering the current production conditions

and equipment, there are 19 input variables to be selected, such as furnace top pressure, pressure drop, and feed blast ratio, and 4 output variables to be selected, including [Si], [S], [P], and molten iron temperature (MIT). Among the 19 input variables, some cannot be directly measured and some cannot be controlled. Because of the high coupling among the variables, not all of them are suitable as control parameters. Considering the large number of variables and the fact that not all of them are suitable as control or controlled variables, it is necessary to select input and output variables that reduce the dimension of the model. Because BF operators typically focus on [Si] and MIT, these two MIQ indicators were selected as the output variables of the model. Then, six variables that have a high correlation with the output variables were selected by the canonical correlation analysis. Furthermore, the correlation analysis was used to find out the combinations of variables with a higher correlation among these six variables. The results of canonical correlation analysis and correlation analysis are shown in Table 4. The measurable and controllable variables among the combinations were selected as the input variables of the model. Finally, the flow rate of cold air, pressure drop, volume of coal injection, and flow rate of rich oxygen were determined as the input variables of the model, and [Si] and MIT were determined as the output variables of the model.

**Table 4.** Results of canonical correlation analysis and correlation analysis.

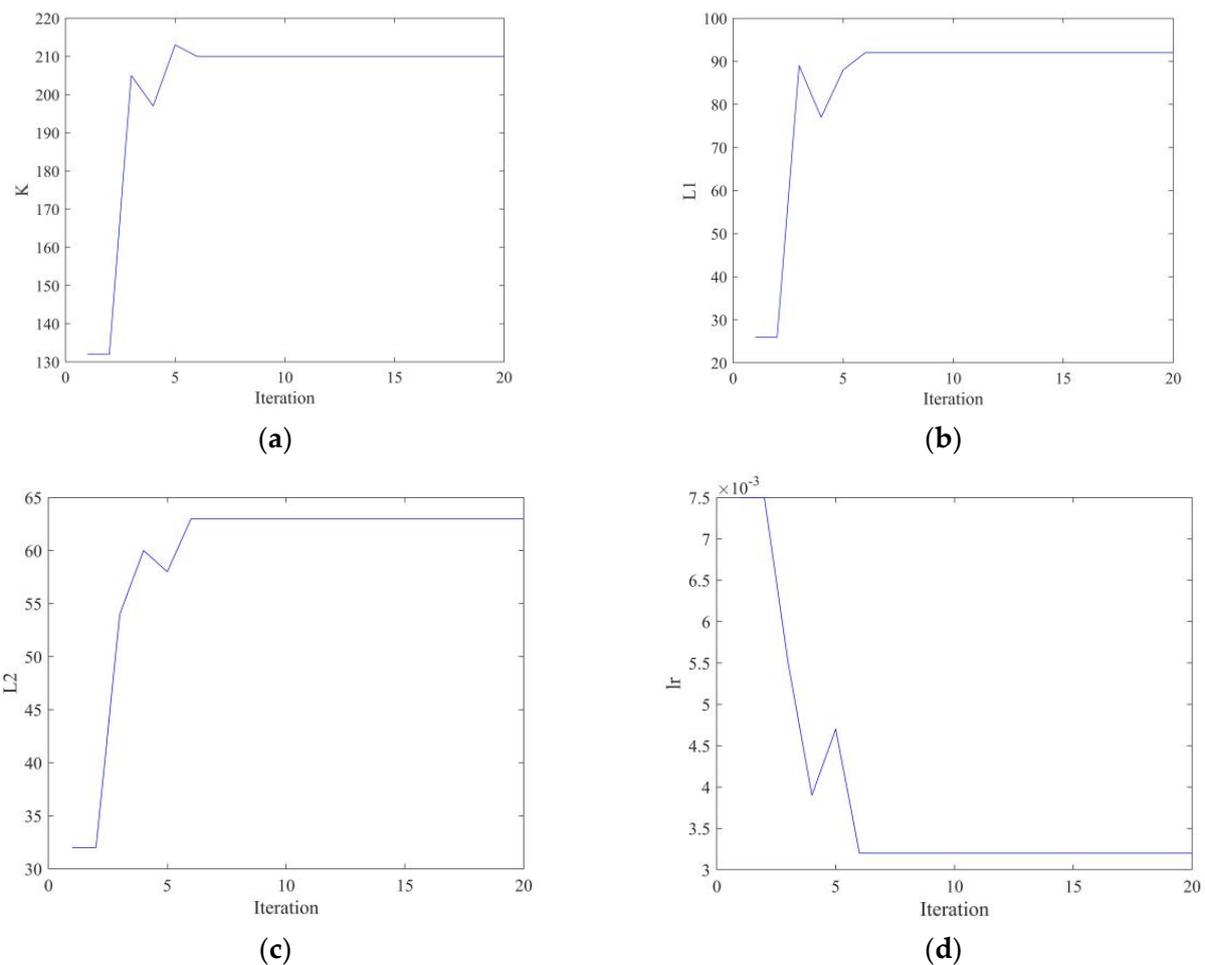
Variable	Typical Variable	
	$-0.191[\text{Si}] - 0.912\text{MIT}$	$-1.063[\text{Si}] + 0.578\text{MIT}$
Flow rate of cold air (m <sup>3</sup> /min)	<b>-1.618</b>	<b>9.488</b>
Feed blast ratio (%)	-0.295	-2.416
Blast pressure (kPa)	0.072	0.778
Furnace top pressure (kPa)	-0.220	-0.140
Pressure drop (kPa)	<b>-2.342</b>	<b>2.687</b>
Top pressure air volume ratio (kPa/m <sup>3</sup> × min)	-0.299	1.252
Gas permeability (m <sup>3</sup> /min × kPa)	-0.475	-2.435
Resistance coefficient (-)	0.744	-5.188
Blast temperature (°C)	-0.293	-3.298
Flow rate of rich oxygen (m <sup>3</sup> /h)	<b>-2.587</b>	<b>-0.450</b>
Oxygen enrichment rate (%)	1.590	-3.421
Volume of coal injection (Kg/t)	<b>0.959</b>	<b>3.781</b>
Blast humidity (g/m <sup>3</sup> )	0.170	1.528
Theoretical burning temperature (°C)	<b>0.991</b>	<b>8.740</b>
Standard wind speed (m/s)	0.479	1.450
Actual wind speed (m/s)	-0.284	2.270
Blast kinetic energy (KJ/s)	0.062	-2.939
Gas volume of bosh (m <sup>3</sup> /min)	<b>2.186</b>	<b>-12.154</b>
Bosh gas index (m <sup>3</sup> /min/m <sup>2</sup> )	0.231	0.175

The experiments were conducted based on the BF body data and the MIQ data of the 2# blast furnace of the Liuzhou Steel Group. Among the collected industrial data, the sampling frequency of the process input variables was more uniform at around 10 s, while the sampling frequency of the MIQ indicators was not uniform and the timing of the inputs and outputs was not consistent. Therefore, time-matching of the input variables and MIQ indicators was required prior to conducting the experiment. Before time-matching, outliers and missing values were removed. Considering that the offline analysis time for the MIQ indicators is around 1 h, it was necessary to bring forward the MIQ data by 1 h and then select the MIQ data that correspond to the sampling interval of approximately 1 h. Because it is not possible to strictly satisfy the sampling interval of 1 h, the sampling interval of the MIQ data obtained will have an error of approximately 10 min. Using the approximately uniformly sampled MIQ data as the reference, input variable data whose sampling time difference was less than a certain threshold (1 min was taken in this experiment) was selected, and the average value of the multiple process variables obtained was taken as

the final process variables. The data were normalized to facilitate further processing. The aim was to predict [Si] and MIT for a future period of sampling. The ironmaking number represents the sampling time at 1 h intervals. Using the preprocessed data, the LSTM, SSA-LSTM, ISSA-LSTM, and ISSA-NSGAIL-LSTM models were used to model the MIQ parameters of BF. The processed BF body data were grouped. Of them, 255 groups were selected as the training set for modeling, and 100 groups were used as the test set for modeling.

To ensure fairness of the test results, the ISSA-NSGAIL-LSTM, ISSA-LSTM, SSA-LSTM, and unoptimized LSTM models were compared and analyzed with the same test set under the same experimental environment. The population size was set to 10, the maximum number of iterations was 20, the initial position was random, and the early warning value was set to 0.5. The proportions of producers, scroungers, and sentinels were 0.2, 0.6, and 0.2, respectively. The hyperparameters, including the number of nodes in the first hidden layer ( $L_1$ ), the number of nodes in the second layer ( $L_2$ ), the learning rate ( $lr$ ), and the number of iterations ( $K$ ) of the model, were optimized by the algorithms.

The search processes of the hyperparameters are shown in Figure 9. It can be seen that the main changes in the four hyperparameters are mainly concentrated in the third to sixth iterations. The number of iterations  $K$  finally converges to 210. The number of nodes in the first hidden layer  $L_1$  converges to 92. The number of nodes in the second hidden layer  $L_2$  converges to 63. The learning rate  $lr$  converges to 0.0032.

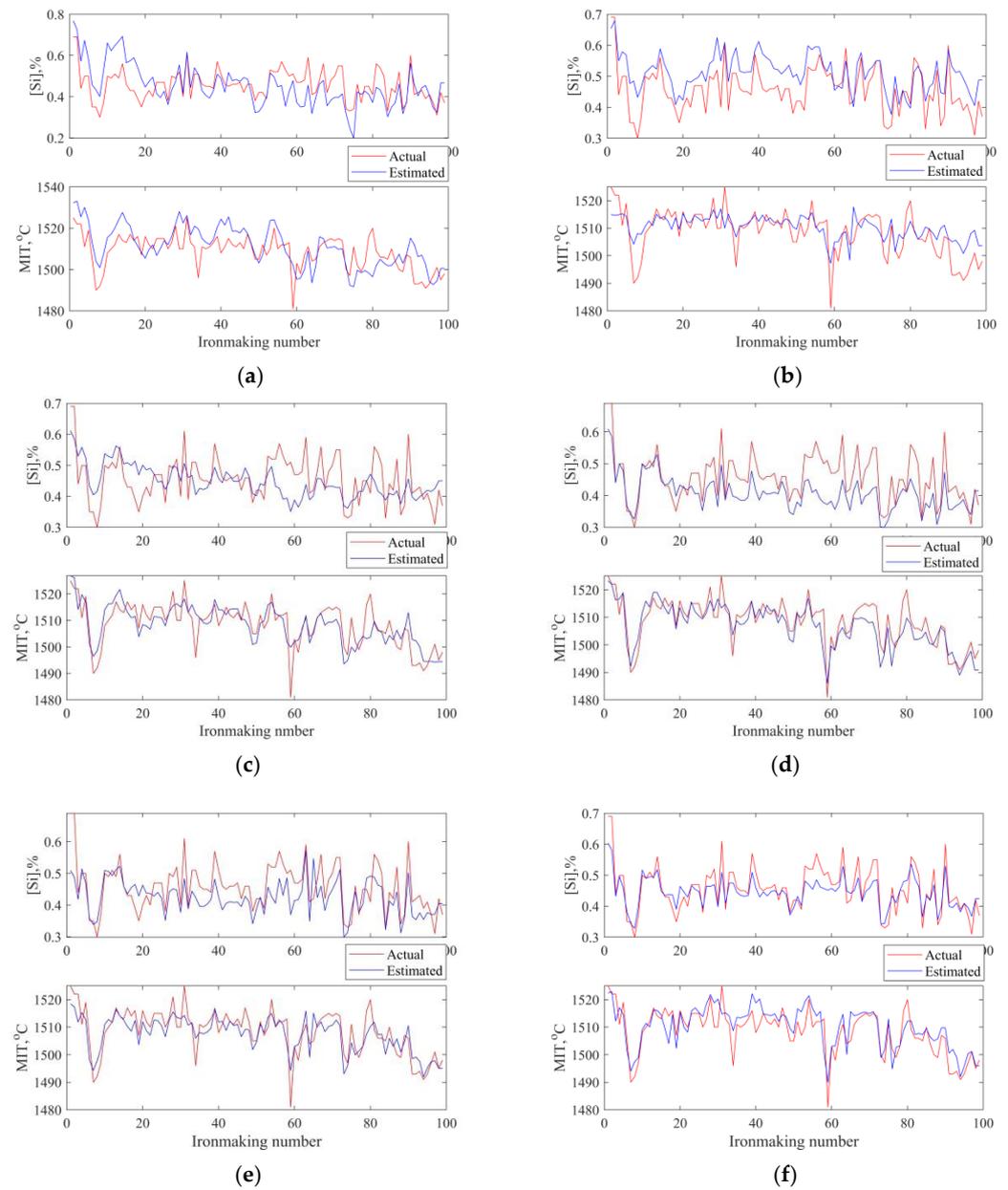


**Figure 9.** Search processes of (a)  $K$ ; (b)  $L_1$ ; (c)  $L_2$ ; (d)  $lr$ .

Based on the screened and processed BF operation data and the four hyperparameters in Table 5, a dataset was established, and the four different models were used to predict the MIQ parameters [Si] and MIT. The prediction results are shown in Figure 10.

**Table 5.** LSTM hyperparameter settings and optimization results.

Model	$K$	$L_1$	$L_2$	$lr$
LSTM	100	50	50	0.0020
SSA-LSTM	134	62	48	0.0057
ISSA-LSTM	167	79	54	0.0072
ISSA-NSGAIL-LSTM	210	92	63	0.0032



**Figure 10.** Prediction results of (a) BP; (b) RVFLNs; (c) LSTM; (d) SSA-LSTM; (e) ISSA-LSTM; (f) ISSA-NSGAIL-LSTM.

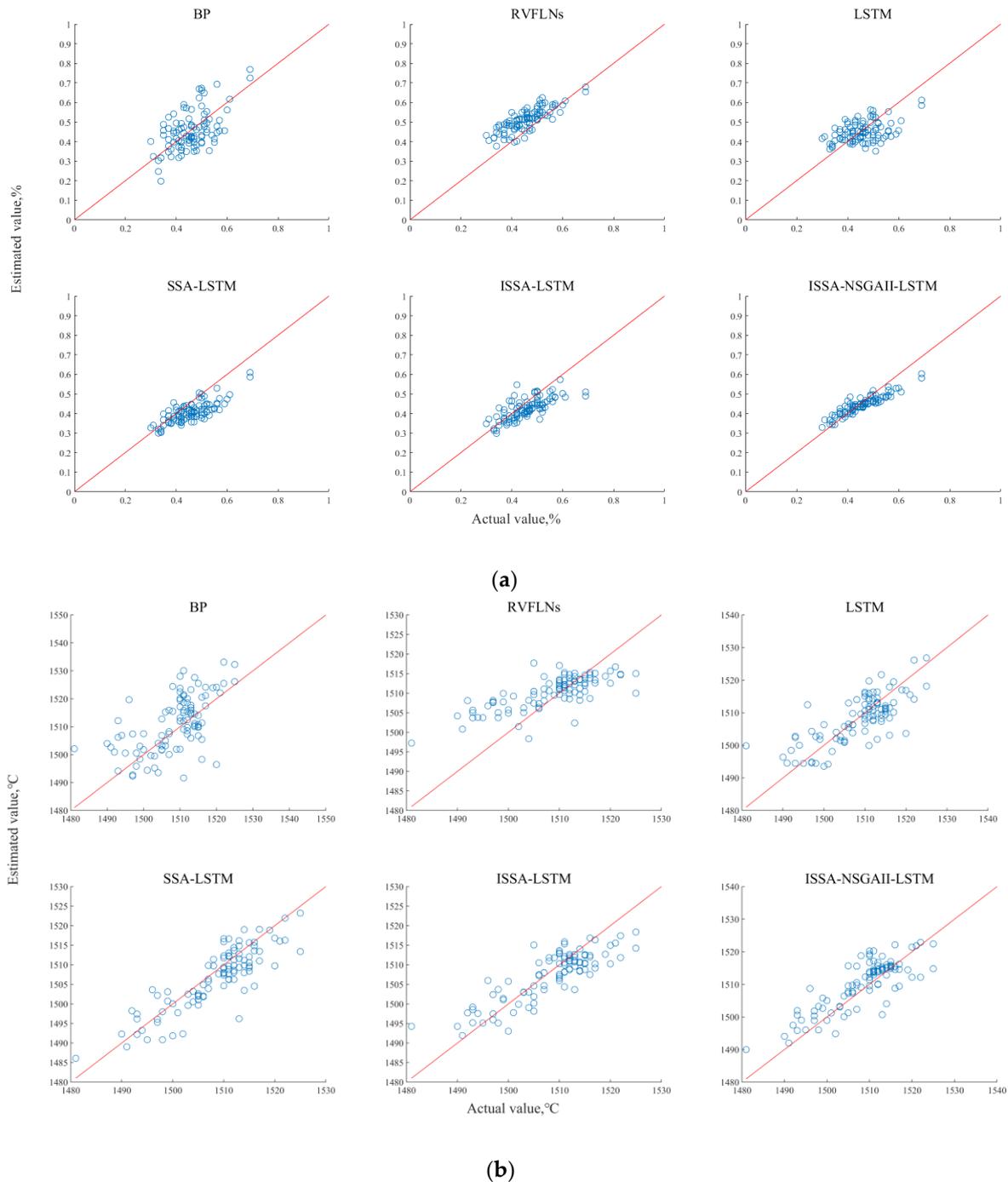
In molten iron quality modeling, the root mean square error (RMSE) is commonly used as a model evaluation indicator [36]. The calculation formula for RMSE is

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{Y}_i - Y_i)^2} \quad (16)$$

where  $\hat{Y}_i$  and  $Y_i$  denote the predicted value and the true value of the  $i$ -th sample, respectively, and  $i = 1, 2, \dots, N$ . The smaller the result of the RMSE, the better the performance of the model. Figure 10a–f show the prediction results of the six different models. Scatter diagrams of the sampling points of the MIQ parameters versus the prediction results are given in Figure 11, with each diagram comparing the prediction effectiveness of each algorithm on the MIQ parameters [Si] and MIT. The x-axis of each subplot is the actual value of the parameters, and the y-axis is the estimated value using the different algorithms. If the scatter is distributed closer to the red diagonal, it indicates a better prediction. If the estimated value matches the actual value, the scatter is distributed along the diagonal. Table 6 lists the RMSEs of each model for [Si] and MIT. As shown in Figure 11, conventional prediction models have scatters far from the diagonal. The scatters of the four LSTM-based models are closer to the diagonal. The scatters of the LSTM model based on the proposed algorithm are closest to the diagonal, which indicates the proposed algorithm has better prediction performance. According to Figure 10 and Table 6, the RMSEs of the backpropagation and random vector functional link networks are larger than those of the models based on the LSTM network, suggesting that the LSTM network has better performance in predicting time series data. Among the results of the four models based on LSTM, it can be seen that ISSA-NSGAIL-LSTM predicts [Si] and MIT better than the others. ISSA-LSTM ranks second, SSA-LSTM is worse, and the unoptimized LSTM is the worst. This indicates that the proposed algorithm improves the performance of the original algorithm, especially in predicting time series data. It can be seen that the RMSE of LSTM for [Si] is 0.0702, and the RMSE of MIT is 5.6069. The RMSE of LSTM for [Si] is 0.0692, and the RMSE of MIT is 4.5841. The RMSE of ISSA-LSTM for [Si] is 0.0613, and the RMSE of MIT is 4.5703. The RMSE of ISSA-NSGAIL-LSTM for [Si] is 0.0388, and the RMSE of MIT is 4.3859, both of which are the optimal values. Compared to ISSA-LSTM, SSA-LSTM, and LSTM, the RMSEs of ISSA-NSGAIL-LSTM for [Si] are lower by 44.73%, 43.93%, and 36.70%, and the RMSEs of MIT are lower by 21.78%, 4.324%, and 4.035%, respectively. To summarize, the above experimental results show that the ISSA-NSGAIL-LSTM model is more stable and has a smaller error than the other models.

**Table 6.** Prediction errors of the models.

Model	RMSE([Si])	RMSE(MIT)
BP	0.0844	8.7118
RVFLNs	0.0701	6.1552
LSTM	0.0702	5.6069
SSA-LSTM	0.0692	4.5841
ISSA-LSTM	0.0613	4.5703
ISSA-NSGAIL-LSTM	0.0388	4.3859



**Figure 11.** (a) Scatter diagram of [Si] estimations with the different models; (b) scatter diagram of MIT estimations with the different models.

## 5. Conclusions

The aim of this study was to optimize the hyperparameters of the LSTM model in order to improve its ability to predict molten iron quality indicators. Aiming at the problems of the basic SSA easily falling into the local optimum and the imbalance between the global and local search abilities, this paper examined several strategies to improve the basic SSA. The tests based on the benchmark functions show that the convergence curve of the improved algorithm is apparently smoother and converges faster, without obvious jumps. This indicates that the improved algorithm balances global and local search abilities, speeds up convergence, and finds the global optimal solution more accurately. The IGD, which reflects the optimization ability, also indicates that the improved algorithm has

better comprehensive optimization performance. The ablation experiments show that the convergence curves of the algorithm improved by a single strategy are better than those of the basic SSA, proving the effectiveness of each improvement strategy. Finally, the validity of the model based on the proposed algorithm was verified using real industrial operation data of the 2# blast furnace of the Liuzhou Steel Group. The results show that the proposed algorithm has better modeling performance with a lower prediction error. In the future, aiming to address the problems of a lack of diversity in the initial population and the premature convergence of SSA, the algorithm should be improved by chaotic mapping and the Tabu search strategy.

**Author Contributions:** Methodology, Z.Z., R.Z. and P.Z.; software, Z.Z.; validation, Z.Z.; formal analysis, P.Z.; writing—original draft, Z.Z. and R.Z.; writing—review and editing, R.Z. and P.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Key R&D Program of China under Grant No. 2022YFB3304903 and the National Natural Science Foundation of China under Grant No. U22A2049.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the author did not have permission to share the data.

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

## References

- Geerdes, M.; Chaigneau, R.; Lingardi, O. *Modern Blast Furnace Ironmaking: An Introduction (2020)*; Ios Press: Amsterdam, The Netherlands, 2020.
- Yang, Y.; Holappa, L.; Saxen, H.; Van der Stel, J. Ironmaking. In *Treatise on process metallurgy*; Elsevier: Amsterdam, The Netherlands, 2024; pp. 7–88.
- Cameron, I.; Sukhram, M.; Lefebvre, K.; Davenport, W. *Blast Furnace Ironmaking: Analysis, Control, and Optimization*; Elsevier: Amsterdam, The Netherlands, 2019.
- Liu, Y.; Zhang, J.; Jiao, K.; Huang, W. *The Operation of Contemporary Blast Furnaces*; Springer: Singapore, 2021.
- Proctor, D.M.; Fehling, K.A.; Shay, E.C.; Shay, E.C. Physical and chemical characteristics of blast furnace, basic oxygen furnace, and electric arc furnace steel industry slags. *Environ. Sci. Technol.* **2000**, *34*, 1576–1582. [[CrossRef](#)]
- Vapnik, V.N.; Lerner, A.Y. Recognition of patterns with help of generalized portraits. *Avtom. I Telemekhanika* **1963**, *24*, 774–780.
- Pisner, D.A.; Schnyer, D.M. Support vector machine. In *Machine Learning*; Academic Press: Cambridge, MA, USA, 2020; pp. 101–121.
- Genuer, R.; Poggi, J.M. *Random Forests*; Springer International Publishing: Cham, Switzerland, 2020.
- Priyanka; Dharmender, K. Decision tree classifier: A detailed survey. *Int. J. Inf. Decis. Sci.* **2020**, *12*, 246–269. [[CrossRef](#)]
- Jian, L. Application of Support Vector Machine in Blast Furnace Temperature Forecasting. Master's Thesis, Zhejiang University, Hangzhou, China, 2006.
- Chen, J. A predictive system for blast furnaces by integrating a neural network with qualitative analysis. *Eng. Appl. Artif. Intell.* **2001**, *14*, 77–85. [[CrossRef](#)]
- Zhang, L. Blast Furnace Molten Iron Quality Parameters Modeling Methods Based on Modified Random Vector Functional-Link Networks. Master's Thesis, Northeastern University, Shenyang, China, 2016.
- Lv, Y.B. Data-Driven Robust Modeling for Molten Iron Quality Parameters Based on RVFLNs. Master's Thesis, Northeastern University, Shenyang, China, 2016.
- Dai, P. Bilinear Subspace Modeling and Nonlinear Predictive Control of Molten Iron Quality Indices in Blast Furnace. Master's Thesis, Northeastern University, Shenyang, China, 2018.
- Zhou, P.; Guo, D.W.; Wang, H.; Chai, T.Y. Data-driven robust M-LS-SVR-based NARX modeling for estimation and control of molten iron quality indices in blast furnace ironmaking. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 4007–4021. [[CrossRef](#)] [[PubMed](#)]
- Hochreiter, S. Untersuchungen zu dynamischen neuronalen netzen. Master's Thesis, Technische Universität München, München, Germany, 1991.
- Chen, L.; Lei, C. Deep learning basics. In *Deep Learning and Practice with Mindspore*; Springer: Singapore, 2021; pp. 17–28.
- Houdt, V.G.; Mosquera, C.; Nápoles, G. A review on the long short-term memory model. *Artif. Intell. Rev.* **2020**, *53*, 5929–5955. [[CrossRef](#)]
- Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
- Abbe, E.; Boix-Adsera, E.; Brennan, M.S. The staircase property: How hierarchical structure can guide deep learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 26989–27002.

21. Bengio, Y. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 437–478.
22. Abbasimehr, H.; Shabani, M.; Yousefi, M. An optimized model using LSTM network for demand forecasting. *Comput. Ind. Eng.* **2020**, *143*, 106435. [[CrossRef](#)]
23. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; Volume 2.
24. Gorgolis, N.; Hatzilygeroudis, I.; Istenes, Z.; Gyenne, L.G. Hyperparameter optimization of LSTM network models through genetic algorithm. In Proceedings of the 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Achaia, Greece, 15–17 July 2019; pp. 1–4.
25. Xue, J.K.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [[CrossRef](#)]
26. Das, S.; Tariq, A.; Santos, T.; Kantareddy, S.S. Recurrent neural networks (RNNs): Architectures, training tricks, and introduction to influential research. In *Machine Learning for Brain Disorders*; Humana: New York, NY, USA, 2023; pp. 117–138.
27. Bautista, L.M.; Alonso, J.C.; Alonso, J.A. Foraging site displacement in common crane flocks. *Anim. Behav.* **1998**, *56*, 1237–1243. [[CrossRef](#)]
28. Lendvai, A.Z.; Barta, Z.; Liker, A.; Bokony, V. The effect of energy reserves on social foraging: Hungry sparrows scrounge more. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* **2004**, *271*, 2467–2472. [[CrossRef](#)] [[PubMed](#)]
29. Xue, J.K. Research and Application of a Novel Swarm Intelligence Optimization Technique. Master’s Thesis, Donghua University, Shanghai, China, 2020.
30. Verma, S.; Plant, M.; Snel, V.A. A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems. *IEEE Access* **2021**, *9*, 57757–57791. [[CrossRef](#)]
31. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T.A.M.T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
32. Iacca, G.; dos Santos Junior, V.C.; de Melo, V.V. An improved Jaya optimization algorithm with Lévy flight. *Expert Syst. Appl.* **2021**, *165*, 113902. [[CrossRef](#)]
33. Singh, N.; Kaur, J. Hybridizing sine–cosine algorithm with harmony search strategy for optimization design problems. *Soft Comput.* **2021**, *25*, 11053–11075. [[CrossRef](#)]
34. Han, M.; Zhong, J.; Sang, P.; Liao, H. A combined model incorporating improved SSA and LSTM algorithms for short-term load forecasting. *Electronics* **2022**, *11*, 1835. [[CrossRef](#)]
35. Ma, H.; Zhang, Y.; Sun, S.; Liu, T.; Shan, Y. A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artif. Intell. Rev.* **2023**, *56*, 15217–15270. [[CrossRef](#)]
36. Xie, J.; Zhou, P. Robust stochastic configuration network multi-output modeling of molten iron quality in blast furnace ironmaking. *Neurocomputing* **2020**, *387*, 139–149. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.