sensors

MDPI

Article

Study on Multi-Model Soft Sensor Modeling Method and Its Model Optimization for the Fermentation Process of *Pichia pastoris*

Bo Wang, Xingyu Wang *, Mengyi He and Xianglin Zhu

Key Laboratory of Agricultural Measurement and Control Technology and Equipment for Mechanical Industrial Facilities, School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China; wangbo@ujs.edu.cn (B.W.); hmy1145755423@163.com (M.H.); zxl4390@126.com (X.Z.)

* Correspondence: wangxingyu79@163.com

Abstract: The problems that the key biomass variables in *Pichia pastoris* fermentation process are difficult measure in real time; this paper mainly proposes a multi-model soft sensor modeling method based on the piecewise affine (PWA) modeling method, which is optimized by particle swarm optimization (PSO) with an improved compression factor (ICF). Firstly, the false nearest neighbor method was used to determine the order of the PWA model. Secondly, the ICF-PSO algorithm was proposed to cooperatively optimize the number of PWA models and the parameters of each local model. Finally, a least squares support vector machine was adopted to determine the scope of action of each local model. Simulation results show that the proposed ICF-PSO-PWA multi-model soft sensor modeling method accurately approximated the nonlinear features of *Pichia pastoris* fermentation, and the model prediction accuracy is improved by 4.4884% compared with the weighted least squares vector regression model optimized by PSO.

Keywords: soft sensor; piecewise affine; improved compression factor; particle swarm optimization; *Pichia pastoris*

check for updates

Citation: Wang, B.; Wang, X.; He, M.; Zhu, X. Study on Multi-Model Soft Sensor Modeling Method and Its Model Optimization for the Fermentation Process of *Pichia pastoris*. *Sensors* **2021**, *21*, 7635. https://doi.org/10.3390/s21227635

Academic Editor: Klaus Stefan Drese

Received: 28 September 2021
Accepted: 11 November 2021
Published: 17 November 2021

1. Introduction

An expression system is a molecular biology technique that uses model organisms such as bacteria, yeast, animal cells or plant cells to express exogenous gene proteins. In short, the expression process is the synthesis of proteins under the guidance of genes. The *Pichia pastoris* expression system is a technically mature eukaryotic expression system, and one of the most successful foreign protein expression systems [1]. The expression system of *Pichia pastoris* has been developed for more than 30 years, and so far, nearly 1000 heterologous proteins have been successfully expressed by this system, which has a promising development [2]. In addition, the low cost but high yield of the *Pichia pastoris* expression system makes it economically valuable. *Pichia pastoris* is also a facultative anaerobe, which is very easy to carry out genetic manipulation and cultivation on [3]. The *Pichia pastoris* expression system has obvious advantages in the processing, external separation, post-translational modification, and glycosylation modification of expression products [4]. Nowadays, many regions are being plagued by the Coronavirus (COVID-19). In order to fight COVID-19, the recombinant expression of proteinase K in *Pichia pastoris* has attracted widespread attention [5,6]. Protease K has drawn attention for its following capabilities: cracking the COVID-19 virus to release nucleic acid, eliminating ribonuclease (RNase) to prevent ribonucleic acid (RNA) degradation, and inactivating the COVID-19 virus to denature the virus protein [7]. Therefore, protease K becomes an important component of nucleic acid detection kit for sample pretreatment, and plays a significant role in COVID-19 detection [8]. However, the fermentation process of recombinant expression of protease K by *Pichia pastoris* is nonlinear, coupled, uncertain, and time-varying [9]. There are few online detection instruments for the key biomass variables of the *Pichia pastoris* fermentation process (such as cell concentration, protease K concentration, etc.) [10] As for the offline assay and analysis methods, the long interval of data acquisition drags down the real-time performance, and adds to the risk of flora pollution. Moreover, the assay results are often inaccurate, due to the large errors of human operation [11]. With the development of the times, biosensors that can detect biomass on-line in real time emerge as the times require. Biosensors have the advantages of high sensitivity and a fast analysis speed; however, a highly automated, miniaturized and integrated biosensor is often very expensive [12].

The above problems give rise to the soft sensor technology. The basic idea of this technology is to model the relationship between target and auxiliary variables, and thus estimate the target variables indirectly, without causing any pollution to flora. The Soft sensor technology is powerful and cost-effective, and it can work in harmony with other software and hardware, which means that it is highly compatible [13]. However, in most cases, it is difficult for a single global soft sensor model to adequately and quickly characterize the process of a complex object. Furthermore, a single global soft sensor model is usually complex in structure and requires a lot of time to solve it. Wang and Han [14] proposed a single soft sensor model based on a recurrent wavelet neural network and Gaussian process regression methods for online prediction of whether Chlortetracycline fermentation broth is contaminated by non-target bacteria. Although the experimental results show that the proposed soft sensor model can be used to predict the occurrence of pollution during the Chlortetracycline fermentation process, and the effectiveness of the modeling method is verified based on field data, the Chlortetracycline fermentation process is a very complex and non-linear process, and most microbial fermentation processes including Chlortetracycline can be divided into four stages: adjustment, logarithmic growth, stabilization and decay phases. During these four phases, the process characteristics of Chlortetracycline are completely different, especially during the exponential growth phase, when the fermentation process is most intense. Therefore, it is impractical to try to describe the complete fermentation process by a single soft sensor model. Although the soft sensor model proposed in this paper showed good prediction results for the Chlortetracycline fermentation process, the model has limitations and cannot be extended to soft sensor modelling of other microbial fermentation processes, i.e., it is not generalizable.

Therefore, in order to address the drawbacks of a single soft sensor model, multi-model soft sensor modeling strategies have attracted a great deal of scholarly attention [15–17]. Scholars have found that using the divide and conquer modeling strategy, building a multi-local soft sensor model can greatly simplify the model structure, save calculation time, and fully mine the information of each sample, hence multi-local soft sensor modeling methods are becoming an inevitable trend in the development of soft sensor technology in the future [18]. As a result, it is highly practical to build a multi-local soft sensor model that supports the real-time online measurement of key biomass variables in *Pichia pastoris* fermentation, as well as the efficient (protease K yield) and high-quality generation of products (protease K quality).

Following the above ideas, Zhang and Cai [19] presented a multi-model soft sensor modeling method based on the improved locally weighted partial least squares (LWPLS), and adopted the soft sensor model to predict the key biomass variables in the fermentation process of alkaline protease. Their approach copes well with the multi-stage and large lag features of microbial fermentation, and realizes good prediction accuracy. However, it is difficult to identify the structural and local parameters of the extremely complex model, and it is time-consuming to complete the heavy calculations. Hence, their approach is not applicable to a wide range of fermentation processes.

In order to avoid the problems mentioned above, this paper will apply a multi-model modeling method called piecewise affine (PWA), which is also called piecewise linear (PWL). Based on the idea of multi-model "decomposition-synthesis", the PWA/PWL multi-model divides the sample space of the system into a finite number of sub-intervals, and each sub-interval is described by a PWA/PWL local model, which transforms the problem of solving a complex nonlinear system into the problem of solving several linear systems [20]. The advantages of the PWA multi-model are that the model structure is very simple; after all, the structure of linear systems is far less complex than that of nonlinear systems, and PWA multi-model is less time consuming than other multi-models because the time to solve a linear system is much less than that to solve a nonlinear system [21,22]. These advantages have made PWA systems a hot research topic in the field of modeling, and they are widely used in many engineering practices. Sun and Wu [23] applied a PWA model to speed regulation in longitudinal dynamics of intelligent vehicles. Muhammad and Michael [24] applied a PWA modeling method to optimal planning of thermal energy systems for microgrids. Sindareh-Esfahani [25] identified a discrete-time PWA model of a wind turbine during Maximum Power Point Tracking region. Mattsson and Zachariah [26] applied the PWA modeling method to the identification of discrete-time nonlinear dynamic models of cascade tanks. All these cases proved the effectiveness and feasibility of the PWA modeling method. Therefore, in this paper, a PWA modeling method was applied to soft sensor modeling of key variables in the fermentation process of *Pichia pastoris*.

In order to improve the prediction accuracy of the PWA multi-model, two model parameters must be optimized, i.e., the number of local models and the parameters of each local model. Too many local models will lead to complex models with a large computational effort and huge dimensionality of local model parameters while too few local models will lead to too simple models with poor prediction performance. In recent years, the research on intelligent optimization algorithms and its applications in model parameter optimization has been very active and has achieved encouraging results [27–29]. Inspired by the foraging behavior of birds, a PSO algorithm as an intelligent swarm optimization algorithm has been successfully applied to the optimization of model parameters [30–32]. The PSO algorithm has no crossover and variational operations, relies on particle velocity to complete the search, and only the optimal particle passes the information to other particles in the iterative evolution, which makes the search speed very fast. The PSO algorithm requires fewer parameters to be adjusted and has a simple structure, which is an effective method to solve practical engineering problems. Many numerical examples show that PSO is better than Differential Evolution (DE), Genetic Algorithm (GA) and Ant Colony Algorithm (ACO) in terms of search efficiency and convergence speed of the global optimal solution [33–35]. This is why this algorithm is used in this paper to optimize the number of local models and the parameters of each local model. However, the PSO algorithm also has some drawbacks, such as the lack of dynamic adjustment of the speed, which is easy to fall into local optimum and requires the selection of appropriate parameters for different problems to achieve the optimal results. Therefore, it is essential to improve the PSO algorithm before the PSO algorithm is used to optimize the number of local models and the parameters of each local model. The reason why the PSO algorithm easily falls into local optimum is that the particle velocity cannot be dynamically adjusted, so the particle velocity must be constrained. The most effective way to constrain the particle velocity is to introduce a compression factor into the particle velocity update formula [36]. However, the traditional compression factor has a fixed form and poor self-adaptability, which makes it difficult to achieve the desired constraint effect [37]. Therefore, this paper will study an improved compression factor (ICF) that can make the compression factor change dynamically as the number of PSO iterations increases [38]. Hence, this study eventually uses this ICF-PSO algorithm to optimize the number of local models and the parameters of each local model. The selection of specific parameters and performance analysis of the ICF-PSO algorithm will be analyzed in detail in the following. The complete PWA modeling process requires not only the optimization of the parameters of the PWA multi-model, but also the determination of the order of the PWA multi-model and the determination of dividing surfaces where each sub-model plays a role, all of which will be explained in detail in the following. To verify the feasibility of the ICF-PSO-PWA multi-model, this study obtained real and accurate original sample data through real fermentation experiments. Supported by these sample data, this study demonstrated the feasibility of applying the PWA multi-model to the *Pichia pastoris* fermentation process through MATLAB simulations, and compared the ICF-PSO-PWA multi-model with the weighted least squares support vector regression model optimized by standard PSO algorithm (PSO-WLSSVR). The simulation results showed that the ICF-PSO-PWA multi-model can effectively predict the key biological variables of the *Pichia pastoris* fermentation process online, and it outperformed the PSO-WLSSVR model.

In summary, the purpose of this study can be summarized into two aspects. The primary study focus is to construct and solve a PWA soft sensor multi-model based on the fermentation process of *Pichia pastoris*, and to demonstrate the effectiveness of the PWA soft sensor multi-model through simulations. The secondary study focus is to propose a novel ICF-PSO optimization algorithm to optimize the PWA soft sensor multi-model, which not only serves the primary study focus but also is an innovative point of this paper.

The rest of the paper is structured as follows: Section 2 consists of materials and methods, which explains PWA model basics, determination of the PWA multi-model order, proposed ICF-PSO algorithm, determination of dividing surfaces and experimental work. Section 3 includes results and discussion. The paper is concluded in Section 4.

2. Materials and Methods

2.1. Structure Diagram of Overall Modeling Algorithm

In order to clarify the components of the prediction algorithm used in this study and their respective functions, this paper first constructs a diagram of the overall modeling algorithm structure here. The functions of each component are then explained in detail in the following sections of this paper. The overall modeling algorithm structure is shown in Figure 1.

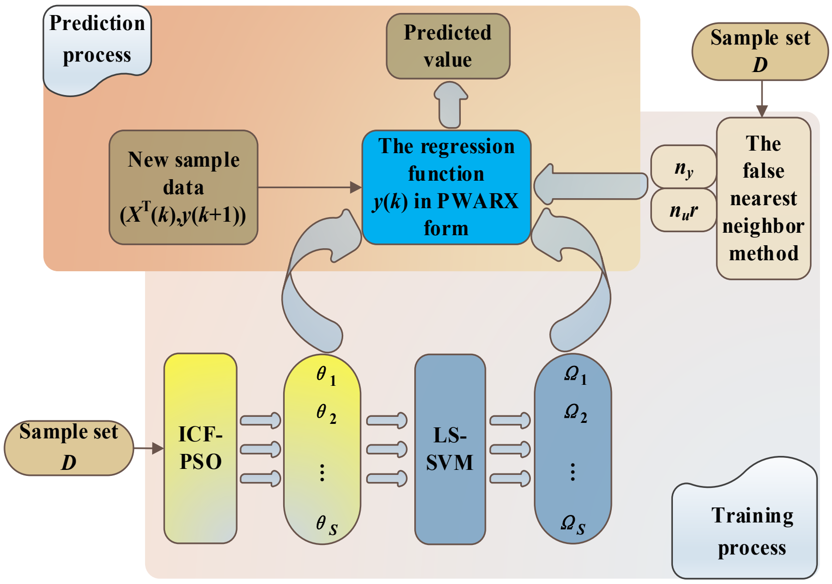


Figure 1. Overall structure diagram of modeling algorithm.

2.2. Description of PWA Modeling Algorithm

The existing sample data evidently testifies the nonlinearity of *Pichia pastoris* fermentation. To facilitate the PWA modeling, the *Pichia pastoris* fermentation (Multi-input Multi-output Nonlinear System) was decomposed into several multiple input single output (MISO) discrete time systems, each of which can be described as a piecewise auto-regressive exogenous (PWARX) model [39]:

$$y(k) = f(X(k)) + \varepsilon(k) \tag{1}$$

where, $y(k) \in R$ is the system output; $\varepsilon(k) \in R$ is the error term.

The PWA mapping $f(\bullet)$ can be defined as:

$$f(X(k)) = \begin{cases} \left[X^T(K), 1\right]\theta_1 & X \in \Omega_1 \\ \left[X^T(K), 1\right]\theta_2 & X \in \Omega_2 \\ \vdots & \\ \left[X^T(K), 1\right]\theta_s & X \in \Omega_s \end{cases} \tag{2}$$

where, $X(k) \in R^n$ is the input vector, which consists of the output and input of the system in the past moment:

$$X(k) = \left[y(k-1), \cdots, y(k-n_y), u_1(k-p-1), \cdots, u_r(k-p-1), \cdots, u_1(k-p-n_u), \cdots, u_r(k-p-n_u)\right]^T \tag{3}$$

where, k is a time series; $y \in R$ is the system output; $u \in R^r$ is the system input; p is the system delay; n_y and n_u are the orders of the model ($n = n_y + n_u$); S is the number of models; $\theta = [\theta_1, \theta_2, \cdots, \theta_S]^T \in R^{(n_y+n_ur+1)\times S}$ is the parameter vector of each linear local model of the PWA system.

The bounded input space Ω_N can be decomposed into S closed polyhedral regions $\{\Omega_i\}_{i=1}^{S}$. Assuming that these closed polyhedral regions have no overlapping parts except the common boundary:

$$\begin{cases} \overset{S}{\underset{i}{\cup}}\Omega_i = \Omega_N \subset R^n \\ \Omega_i \cap \Omega_j = \varnothing \quad i \neq j \end{cases} \tag{4}$$

where, Ω_i is a closed polyhedron: $\Omega_i = \left\{X \in R^n \middle| H_i\left[X^T, 1\right]^T \geq 0\right\}$, with $H_i \in R^{q_i}$ which determines the boundary of a closed polyhedron Ω_i, i.e., the dividing surface dividing Ω_i from other polyhedrons (q_i is the number of linear inequalities defining the i-th polyhedron region), and $\geq$ being an inequality:

$$\text{If } H_i = \begin{bmatrix} h_{i1} \\ h_{i2} \end{bmatrix} h_{i1} \in R^{n+1} \text{ and } h_{i2} \in R^{n+1}$$

$$\text{Then } H_i\left[X^T, 1\right]^T \geq 0 \Leftrightarrow \begin{cases} h_{i,1}\left[X^T, 1\right]^T \geq 0 \\ h_{i,2}\left[X^T, 1\right]^T \geq 0 \end{cases}$$

Suppose the PWA model is unknown, and the modeling space consists of N datasets on the fermentation process:

$$Z_N = \{z(k)\}_{k=1}^{N} \text{ and } z(k) = \left[X^T(k), y(k)\right]^T \tag{5}$$

where, $Z_N \in R^{(n+1)N}$. Then, the identification problem of the PWA aims to reconstruct the PWA mapping to characterize all the features of the original system, that is, to solve the following optimization problem:

$$\begin{aligned} \min J_1 &= \min \textstyle\sum_k \left[y(k) - \hat{y}(k)\right]^2 \\ &= \min \sum_{z(k)\in Z_N} \left[y(k) - \hat{f}(X(k), \theta_i)\right]^2 \end{aligned} \tag{6}$$

where, $\hat{y}(k) \in R$ is the output predicted based on the input vector $X(k)$ at time k; $\hat{f}(\bullet, \bullet) \in R$ is the local PWA model of the current work space:

$$\text{Let } \delta_i = \begin{cases} 1 & X(k) \in \Omega_i \\ 0 & X(k) \notin \Omega_i \end{cases} \quad \Omega_i \subset \Omega_N$$

Then, Equation (2) can be represented as:

$$\hat{f}(X(k),\theta_i)=\sum_{i=1}^{S}\left[X^T(k),1\right]^T\theta_i\delta_i \text{ and } X(k)\in\Omega_N \tag{7}$$

Therefore, the solution problem of MISO nonlinear system in each fermentation process can be transformed into a mixed integer quadratic programming (MIQP) problem [40]:

$$\begin{gathered}\min_{n,\theta,S} J_1=\min_{n,\theta,S}\sum_{i=1}^{N}\left[y(k)-\hat{f}(X(k),\theta_i)\right]^2\\ s.t.\ \hat{f}(X(k),\theta_i)=\sum_{i=1}^{S}\left[X^T(k),1\right]^T\theta_i\delta_i\end{gathered} \tag{8}$$

where, $\delta_i=\begin{cases}1 & X(k)\in\Omega_i\\ 0 & X(k)\notin\Omega_i\end{cases}$, $y(k)=f(X(k))$, $X(k)\in\Omega_N$ and $\left[X^T(k),y(k)\right]^T\in Z_N$. It is not hard to see from the objective function J_1 that MIQP only identifies θ_i. However, in order to identify θ_i, several prerequisites are needed, namely, solving for the order n of the input vector $X(k)$ and determining the number of local models S. The problem of identifying n and S is a problem of identifying two hyperparameters, and in this study the optimal n and S will be solved by different algorithms respectively. When S and n are known, the objective function J_1 becomes an easier-to-solve MIQP problem containing only one decision variable θ_i. It is only necessary to find the optimal θ_i by the corresponding optimization algorithm for J_1. After all the above parameters are determined, the range Ω_i in which the local model works can be easily determined by just the corresponding classification algorithm.

Without sufficient prior information, however, it is usually difficult to determine the number of models or the parameters of each local model. Therefore, this paper modifies the goal of optimization modeling from maximizing model accuracy by single optimization to maximizing the approximation degree of the model while minimizing the number of local models. That is, the single-objective Problem (6) was transformed into a multi-objective optimization problem:

$$\min\{J_1,J_2\} \tag{9}$$

where, $J_2=S$ is an integer representing the number of local models; J_1 is given by Equation (8), reflecting the approximation of the model. The value of J_1 is negatively correlated with the approximation degree of the model. Thus, the goal of maximizing the approximation degree of the model is equivalent to minimizing the objective function J_1.

2.3. Determining Model Order

In the absence of sufficient prior information, it is a complex issue to determine the appropriate order of the PWA model. The topological delay embedding theorem [41] suggests that a nonlinear system cannot be completely modeled if the final model has a too-small order; if the model order is too large, the modeling can be complete in theory, but the actual modeling will require heavy computations, and face lots of system noises and a large rounding error [42].

The false nearest neighbor (FNN) method is suitable for determining the embedding dimension in the state space reconstruction of nonlinear time series. The core idea of this method is to iteratively determine the false and real neighboring points of a certain trajectory x_n, while increasing the embedding dimension in each iteration. The iterative process terminates when the number of false neighbors is zero, and outputs the minimum required embedding dimension. From the geometric perspective, the FNN is a very simple and easy approach to determine the embedding dimension [43]. Hence, this paper adopts this method to calculate the order of the PWA multi-model.

Step 1. Determining the false nearest neighbor

Considering the input/output data on fermentation, construct the following n-dimensional regression vector:

$$X(k) = \left[y(k-1), \cdots, y\left(k-n_y\right), u_{\,1}(k-p-1), \cdots, u_{\,r}(k-p-1), \cdots, u_{\,1}(k-p-n_{\,u}), \cdots, u_{\,r}(k-p-n_{\,u})\right]^{\,T} \quad (10)$$

Rewrite the regression vector $X^{(n)}(k)$ as:

$$x_p = \left[x_{p-1}, x_{p-2}, \cdots, x_{p-n_y}, x_{p-(n_y+1)}, \cdots, x_{p-(n_y+n_u)}, \cdots, x_{p-(n_y+n_u r)}\right]^T \quad (11)$$

$$p = N_0, N_0+1, \cdots, N \quad (12)$$

where, $N_0 = n+1$.

Define the distance from x_i to x_j in R^n by L_∞ norm:

$$\left\|x_i - x_j\right\|_\infty = \max\left|x_{i+1} - x_{j+1}\right| \quad (13)$$

Let x_p be the nearest point of $x_{\eta(p)}$:

$$\left\|x_{\eta(p)} - x_p\right\|_\infty^{(n)} = \min_{j=N_0,\cdots,N;j\neq n}\left\|x_p - x_j\right\|_\infty = \min_{j=N_0,\cdots,N;j\neq n}\ \max_{0\leq l\leq n-1}\left|x_{p+l} - x_{j+l}\right| \quad (14)$$

Then, $N_0 \leq \eta(p) \leq N$, and $\eta(p)$ is related to n, that is, $\eta(p)$ is related to n_y and n_u.

As the embedding dimension changes from n to n + 1, the distance from x_i to x_j becomes: $\left\|x_{\eta(p)} - x_p\right\|_\infty^{(n+1)} = \min_{j=N_0,\cdots,N;j\neq n}\left\|x_p - x_j\right\|_\infty = \min_{j=N_0,\cdots,N;j\neq n}\ \max_{0\leq l\leq n-1}\left|x_{p+l} - x_{j+l}\right|$. If $\left\|x_{\eta(p)} - x_p\right\|_\infty^{(n)} \ll \left\|x_{\eta(p)} - x_p\right\|_\infty^{(n+1)}$, namely, two non-adjacent points become two adjacent points when projected on the low trajectory, such adjacent points are the false neighbors.

Specifically, $x_{\eta(p)}$ is considered as the false nearest neighbor of x_p if it satisfies:

$$\left|\frac{\left\|x_{\eta(p)} - x_p\right\|_\infty^{(n+1)} - \left\|x_{\eta(p)} - x_p\right\|_\infty^{(n)}}{\left\|x_{\eta(p)} - x_p\right\|_\infty^{(n+1)}}\right| \geq R_T \quad (15)$$

where, R_T is a threshold value, $R_T \in [10, 50]$.

With the increase of dimension n, the ratio of false neighbors δ decreases, where δ is the ratio of false neighbors to total neighbors. When δ is less than a certain threshold δ_0 or δ no longer changes with the change of n, then the geometric structure of the state space has been completely opened, and the dimension n at this time is the embedding dimension.

Step 2. Calculating the minimum embedding dimension

Firstly, define the variable $\alpha(p,n)$:

$$\alpha(p,n) = \frac{\left\|x_{\eta(p)} - x_p\right\|_\infty^{(n+1)}}{\left\|x_{\eta(p)} - x_p\right\|_\infty^{(n)}} \quad (16)$$

Then, define the mean value of $\alpha(p,n)$ with respect to n as:

$$E(n) = \frac{1}{N-N_0+1}\sum_{n=N_0}^{N}\alpha(p,n) \quad (17)$$

To capture the change of embedding dimension from n to n + 1, redefine:

$$E_0(n) = \frac{E(n+1)}{E(n)} \quad (18)$$

The minimum embedding dimension of reconstructed state space can be judged according to the change of $E_0(n)$. If it is found that when the embedding dimension increases to a certain extent, $E_0(n)$ will no longer change or change to a very small extent, then $n + 1$ is the required minimum embedding dimension. At this time, the corresponding n_y and n_u are the orders of the corresponding model [44]. After using the false nearest neighbor method, the optimal order of the input vector (n_u) and output vector (n_y) of the PWA model are 1 and 2 respectively.

2.4. Collaborative Optimization of the Number and Parameters of the Models

In order to improve the prediction performance of PWA multi-models, the number of sub-models and the parameters of each sub-model must be optimized. The PSO is widely adopted for fuzzy system control, neural network training, and function optimization, because it is simple to define and easy to implement, with fast computing speed [45]. Therefore, this paper applies the PSO to determine the number of models and to optimize the parameters of local models in the PWA multi-model, and introduces the ICF to improve the ability of the standard PSO to avoid the local optimum early and perform accurate local search thereafter.

2.4.1. PSO Algorithm

In the standard PSO algorithm, each particle in the swarm is of zero volume in the D-dimensional search space, and is represented by its current position. Each particle flies at a certain velocity in the search space, and keeps adjusting the velocity based on its experience and that of every other particle.

The current velocity and position of the i-th particle can be defined as:

$$V_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,D}) \text{ and } P_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,D}) \tag{19}$$

Then, the individual best-known position of the i-th particle and the global best-known position of the swarm can be respectively expressed as:

$$P_i^b = \left(P_{i,1}^b, P_{i,2}^b, \cdots, P_{i,D}^b\right) \text{ and } P_g^b = \left(P_{g,1}^b, P_{g,2}^b, \cdots, P_{g,D}^b\right) \tag{20}$$

For the generation $t + 1$, the velocity and position of the i-th particle in the j-th dimension ($1 \leq j \leq D$) can be updated by:

$$v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_{1,i}\left(p_{i,j}^b(t) - p_{i,j}(t)\right) + c_2 r_{2,i}\left(p_{g,j}^b(t) - p_{i,j}(t)\right) \tag{21}$$

$$p_{i,j}(t+1) = p_{i,j}(t) + v_{i,j}(t+1) \tag{22}$$

where, w is inertia weight; c_1 and c_2 are acceleration coefficients, both of which are constants; $r_{1,i}$ and $r_{2,i}$ are two random functions that change in the range of [0, 1].

In the position update Equation (21) of the standard PSO, if $p_{i,j} = p_{i,j}^b = p_{g,j}^b$, which means that the current position of particle i is the best position it has experienced and also the best position the whole particle population has experienced so far, then the change of $v_{i,j}(t+1)$ only depends on $wv_{i,j}(t)$. This implies that when the swarm approaches a local optimal solution, its velocity will approach zero. This means that all particles tend to stop moving near the local optimal solution. Thus, the standard PSO would converge prematurely, and get stuck in the local optimal position. However, there are many local optimal solutions for non-convex problems, and the standard PSO will converge to a local optimum very easily. For these reasons, PSO must be improved.

2.4.2. ICF-PSO Algorithm

To overcome premature convergence, this paper improves the PSO into the ICF-PSO. The position update formula was not changed, while the velocity update formula was refined by introducing the ICF μ:

$$v_{i,j}(t+1) = \mu\left(wv_{i,j}(t) + c_1 r_{1,i}\left(p_{i,j}^b(t) - p_{i,j}(t)\right) + c_2 r_{2,i}\left(p_{g,j}^b(t) - p_{i,j}(t)\right)\right) \tag{23}$$

where, $\mu = \delta^{\sigma^{\frac{G}{N+G}}-1}$ is the compression factor, which both restrains the velocity of particles and enhances the local search ability of the algorithm. Note that δ and σ are constants in μ, both of which are key to the control of the constraint effect of the compression factor; N is the total number of iterations; G is the current number of iterations. The rationale for the improved compression factor is as follows. In this paper, the characteristic curve of μ is made to be a monotonically decreasing curve in the first quadrant as the number of current iterations G increases by determining the appropriate values of δ, σ and N. This curve must contain the following characteristics, that is, the value of μ must vary between 1 and 0; the value of μ must decrease rapidly at the beginning of the iteration, and then the rate of decrease of the value of μ tends to smooth out as the iteration proceeds to the middle and later stages. The compression factor satisfying the above conditions is substituted into the velocity update formula of the standard PSO algorithm, which makes the particle velocity decrease continuously during the iteration process under the constraint of the compression factor. The particle velocity satisfies the following characteristics during the decreasing process, the particle can maintain a fast velocity at the beginning of the iteration, which can make the particle jump out of the local optimum effectively with the fastest speed, but the rate of particle velocity decreasing is fast during this time. As the iteration proceeds to the middle and later stages, the particle velocity becomes small and the particle velocity decreases very little and smoothly at this time. This kind of particle is able to search the global optimum precisely and smoothly, just like the carpet search of radar. The above is the rationale for the improved compression factor μ. This paper will describe in detail how to determine reasonable values of δ, σ and N, in the following subsections.

2.4.3. Determination of Parameter Values in ICF

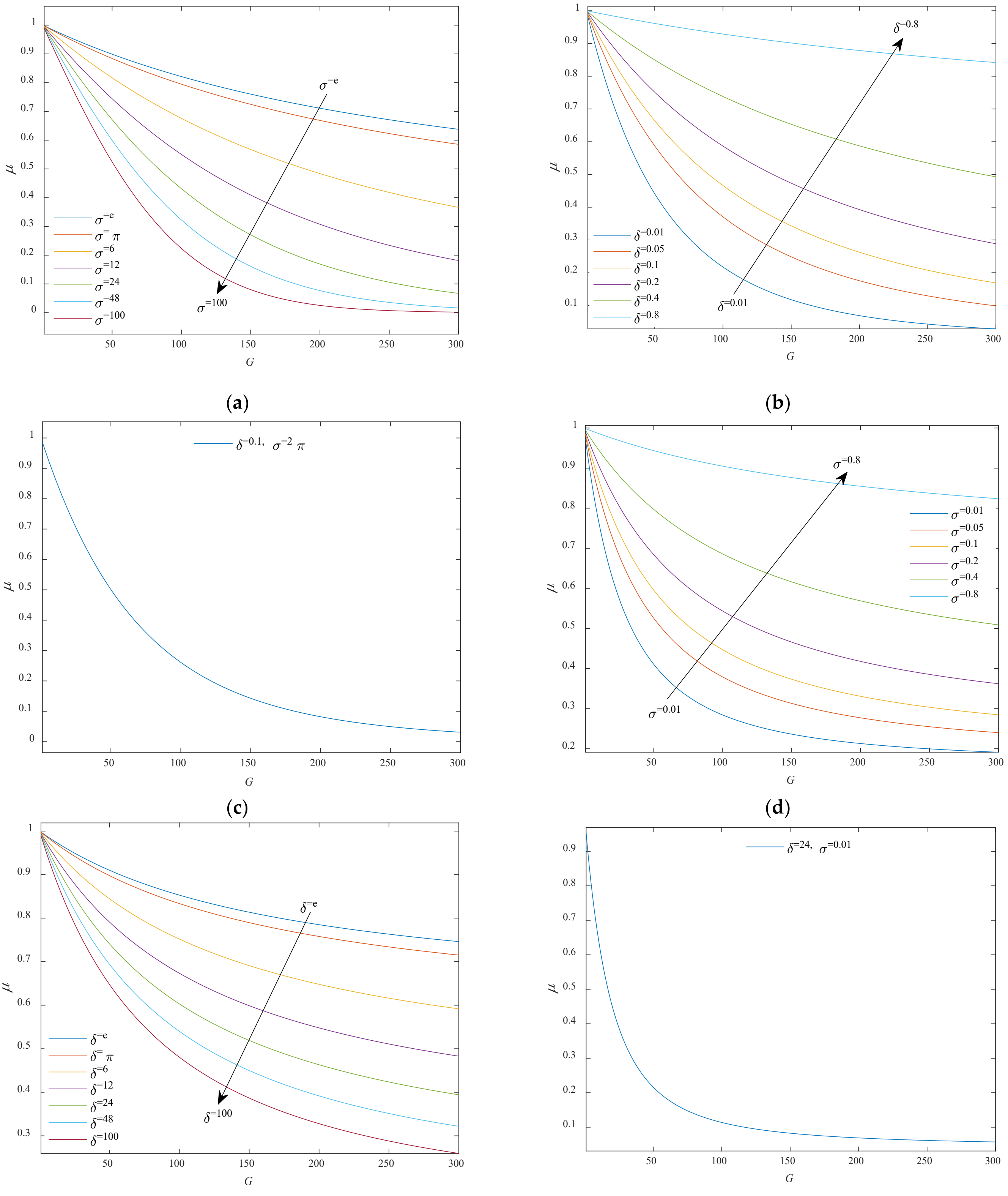
This paper tries to maximize the efficiency and optimization effect of the compression factor by determining the value range of δ and σ. Firstly, the total number of iterations N was assumed to be 300. Secondly, both δ and σ must be positive numbers, according to mathematical knowledge (otherwise, μ will be meaningless). Finally, the value of the compression factor should decrease with the increase of G, judging by its functional features. Therefore, the value range of δ and σ could only be one of the two cases: (1) $\delta \in (0,1)$ and $\sigma \in (1,+\infty)$; (2) $\delta \in (1,+\infty)$ and $\sigma \in (0,1)$. The variation curves of μ in these two cases are discussed in detail below:

(1) $\delta \in (0,1)$ and $\sigma \in (1,+\infty)$

Figure 2a shows the curve of μ with the increase of σ at δ = 0.5, and Figure 2b shows the curve of μ with the increase of δ at σ = π. It can be seen that when the value of δ was constant, the constraint performance of ICF μ was better at a larger σ value; when the value of σ was fixed, the smaller the value of δ was, the better the constraint performance was. The values of σ and δ should be considered comprehensively to optimize the constraint performance of the ICF μ. Experimental results show that, in case (1), the constraint performance of μ is the best at δ = 0.1 and σ = 2π (Figure 2c).

(2) $\delta \in (1,+\infty)$ and $\sigma \in (0,1)$

Figure 2d shows the curve of μ with the increase of σ at δ = 2π, and Figure 2e shows the curve of μ with the increase of δ at σ = 0.5. It can be seen that when the value of δ was unchanged, the constraint performance of ICF μ was better at a smaller σ value; when the value of σ was fixed, the larger the value of δ was, the better the constraint performance was. The values of σ and δ should be considered comprehensively to optimize the constraint performance of the ICF μ. Experimental results show that, in case (2), the constraint performance of μ is the best at δ = 24 and σ = 0.01 (Figure 2f).



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 2. The characteristic curve of μ when the values of δ and σ change, including (**a**) $\delta = 0.5$ and σ was increasing between $(1,+\infty)$, (**b**) $\sigma = \pi$ and δ was increasing between (0.1), (**c**) one of the best constraint performance of μ was at $\delta = 0.1$ and $\sigma = 2\pi$, (**d**) $\delta = 2\pi$ and σ was increasing between, (**e**) σ = 0.5 and μ was increasing between $(1,+\infty)$, (**f**) one of the best constraint performance of μ was at $\delta = 24$ and σ = 0.01.

To sum up, the compression factor μ has an optimal state in either case, depending on the values of δ and σ. Here, the δ and σ values in compression factor μ are set to 24 and 0.01, respectively.

In fact, the concept of compression factor φ was put forward by Clerc et al. in as early as 1999 [46]:

$$\varphi = \frac{2}{\left|2 - \rho - \sqrt{\rho^2 - 4\rho}\right|} \text{ and } \rho = c_1 + c_2 \tag{24}$$

where, $\rho > 4$ and φ is the compression factor; c_1 and c_2 are acceleration constants. The problem is that the compression factor is too rigid [47]. The value of φ only depends on the size of c_1 and c_2, failing to consider the iterative effect. With the elapse of time, such a compression factor is no longer suitable for optimizing complex nonlinear systems. Compared with the original compression factor φ, the improved compression factor μ proposed by the authors can change iteratively, and adapt its constraint effect to the specific optimization task, in the light of the combined effect of factors like δ, σ, N and G.

The details of the performance analysis of the ICF-PSO algorithm are shown in the Appendix A.

2.4.4. ICF-PSO Algorithm Applied to Optimize the Number and Parameters of PWA Local Models

The multi-objective optimization Problem (9) cannot be easily solved by the traditional phased strategy, because of the strong coupling between the two objective functions. Drawing on the idea of non-inferior optimal solution, this paper chooses to optimize objective function J_1 by the step-by-step iterative method, in view of the fact that J_1 depends heavily on J_2 but J_2 is independent of J_1: First, select the non-optimal solution S that satisfies J_2 and determine an optimal search threshold Y_{ts} for J_1. Then, S is substituted into J_1, and J_1 is optimized to obtain the optimal solution of the corresponding local model parameter; if the optimized J_1 cannot meet the preset threshold Y_{ts}, optimize S and re-optimize J_1. Repeat these steps until the threshold Y_{ts} is satisfied.

To optimize objective function J_2, it is critical to reasonably divide the input space Ω_N into S subspaces that satisfy conditional Equation (25), so as to determine the number of local models:

$$\sum_{i=1}^{S} M_i = N \tag{25}$$

where, M_i is the number of input vectors contained in the i-th subspace; $\Omega_i (i = 1, 2, \cdots, S)$ is the i-th input subspace.

After the input space dividing Ω_i is obtained, the corresponding sample dataset can be determined by:

$$Z_i = \{z(k) | X(k) \in \Omega_i,\ y(k) = f(X(k))\} \tag{26}$$

where, $\bigcup\limits_{i=1}^{S} X_i = Z_N$; Z_i is the sample dataset corresponding to Ω_i, which uniquely determines the modeling space of $f\ (X(k))$.

When the system model is unknown, $y(k) \in R$ can be replaced by the actually observed output. The difficulty of J_2 is that the division of input space cannot be separated from the parameter identification of the local model. The strong coupling between the two objective functions makes it necessary to determine the number of models and estimate the model parameters simultaneously. Therefore, this paper employs the ICF-PSO algorithm, which excels in parallel search and data clustering to optimize the number of models and parameters.

During the optimization, the modeling parameters reflecting each local model are combined with the corresponding cluster center into a particle:

$$P = \{\theta_1, Z_1, \cdots, \theta_i, Z_i, \cdots, \theta_S, Z_S\} \tag{27}$$

where, $Z_i(i = 1, 2, \cdots, S)$ is the cluster center of the i-th cluster.

By the ICF-PSO algorithm, the study optimized the number of local models and the parameters of each local model at the same time [48]. The detailed steps are as follows:

(1) Step 1. Initialization

Set the number of non-optimal models S = 1, swarm size N_S, inertia weight w, acceleration coefficients c_1 and c_2, the threshold Y_{ts} of objective function J_1, maximum allowable number of iterations N_{it}, initial position P_i and initial velocity of each particle V_i, etc.

(2) Step 2. Data clustering

Because the PWA system is locally linear, the data close to each other are very likely to fall into the same class. Thus, each local subspace Z_i can be divided by the distance between the data and the given cluster center. The specific method is to calculate the norm distance from each modeling space vector $z(k)$ to a given cluster center $\hat{Z}_i$; if the modeling space vector is the closest to a given cluster center, it will be assigned to that cluster:

$$z(k) \in Z_i \text{ and } I = \underset{k}{\operatorname{argmin}} \|z(k) - \hat{Z}_i\|_2 \tag{28}$$

where, $i = 1, 2, \cdots, S$.

(3) Step 3. Initial fitness evaluation

Evaluate the initial fitness of each particle according to the objective function J_1.

(4) Step 4. Evolution of particle swarm

Update the position and velocity of particles according to Equations (22) and (23), respectively.

(5) Step 5. Fitness update

Cluster the data by Step 2, and update the fitness of each particle by objective function J_1.

(6) Step 6. Termination conditions

(1) If the maximum number of iterations is not reached and the threshold of the objective function is not met, return to Step 3.
(2) If the maximum number of iterations is not reached, but the threshold of the objective function is met, terminate the algorithm.
(3) If the maximum number of iterations is reached and the threshold of the objective function is met, terminate the algorithm.
(4) If the maximum number of iterations is reached, but the threshold of the objective function is not met, go to Step 7.

(7) Step 7. Make J_1 = S + 1, return to Step 1, and restart the search

The above steps can be organized into the following flow chart in Figure 3.

2.5. Determination of Dividing Surfaces

So far, the sample space has been divided into S local spaces Ω_i (i = 1, 2, ⋯, S). According to the definition of the PWA system in Equation (2), the convex polyhedrons with different scopes have no overlapping parts except the common boundary. Therefore, the problem of finding the optimal dividing surface is directly transformed into the linear division of cluster data. The least square support vector machine (LS-SVM) algorithm was selected to solve the optimal dividing surface between local PWA models. The selection was made because the LS-SVM is a classical classifier to handle small samples, and high-dimensional nonlinear data. The training speed of the model is accelerated, owing to the combination between the standard SVM and the least squares method.

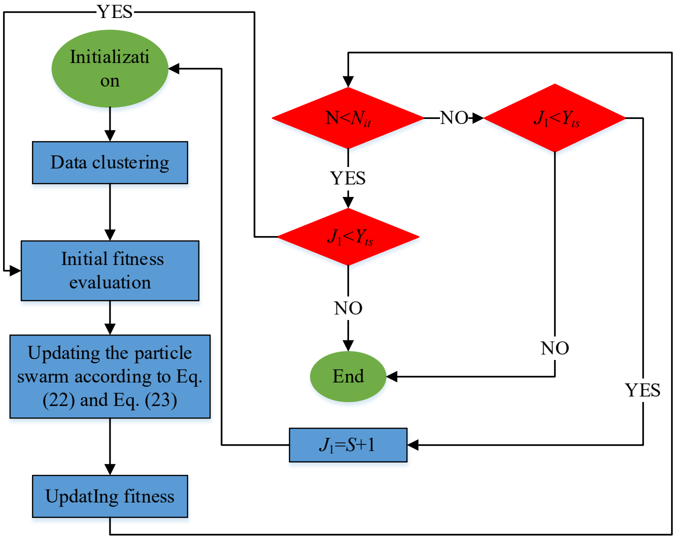


Figure 3. Flow chart of ICF-PSO for optimizing the number and parameters of PWA local models.

2.5.1. Least Square Support Vector Machine

The core idea of LS-SVM is that in a given sample data set $\{x_k, y_k\}_{k=1}^{N}$, in which $x_k \in R^n$ and $y_k \in \{-1, 1\}$. To find the optimal classification hyperplane $w^T x + b = 0$, that is, to solve the following optimization problems [49], namely

$$\begin{array}{l} \min J(w,e) = \min\limits_{w,b,e} \frac{1}{2} w^T w + \mu \sum\limits_{k=1}^{N} e_k^2 \\ s.t. y_k \left[w^T \varphi(x_k) + b \right] = 1 - e_k \quad k = 1, 2, \cdots, N \end{array} \tag{29}$$

where, $\varphi(g) : R^n \to R^{n_H}$ is a function that maps input data to a high-dimensional feature space; weight vector $w \in R^{n_H}$; error variables and offset values satisfy $e_k \in R$ and $b \in R$; $\mu > 0$ is the weight coefficient, which can make LS-SVM find the optimal hyperplane and ensure the minimum deviation [50].

Introducing Lagrange function into Equation (29), then

$$L(w, b, e; \alpha) = J(w, e) - \sum_{k=1}^{N} \alpha_k \left\{ \left[y_k \left(w^T \varphi(x_k) + b \right) - 1 + e_k \right] \right\} \tag{30}$$

where, $\alpha_k \geq 0$ is Lagrange multiplier; x_k is the support vector.

By deriving w, b, e_k, α_k in the above formula respectively, the following linear equation system about α and b can be obtained:

$$\begin{bmatrix} 0 & y^T \\ y & \Omega + I/\mu \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_v \end{bmatrix} \tag{31}$$

where, $y = [y_1, \cdots, y_N]$; $\alpha = [\alpha_1, \cdots, \alpha_N]$; $1_v = [1, \cdots, 1]$; Ω is called a kernel matrix, and its expression is

$$\Omega_{i,j} = y_i y_j \varphi(x_i)^T \varphi(x_j) = y_i y_j K(x_i, x_j) \tag{32}$$

where, $i,j = 1, \cdots, N$. A set of α and b can be obtained by solving Equation (31). Finally, the classification hyperplane obtained by LS-SVM is shown below:

$$f(x) = sgn\left[\sum_{k=1}^{N} \alpha_K y_k K(x, x_k) + b \right] \tag{33}$$

where, the kernel function must satisfy Mercer condition. The kernel function of this paper uses RBF kernel function. The expression of the RBF kernel function is shown below.

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right) \tag{34}$$

where, σ is the width parameter of the function, which controls the radial range of action of the function.

2.5.2. LS-SVM Applied to Determine the Parameters of Dividing Surfaces

To begin with, a sample space is defined:

$$Z_i = \{z(k)|X(k) \in \Omega_{\ i},\ y(k) = f(x(k))\}_{k=1}^{M_i} \tag{35}$$

where, $Z_N = \bigcup_{i=1}^{S} Z_i$.

The LS-SVM determines the dividing surface in two steps: First, find out the adjacent subspaces; Then, determine the parameters of dividing surfaces between adjacent subspaces. The specific steps are as follows:

Step 1. Determine the modeling subspaces adjacent to each other by the nearest neighbor rule of cluster center, namely:

$$\{Z_i, Z_j\} = \min_{2\leq i,j\leq s, i\neq j}\left\{\|Z_i - Z_j\|_2\right\} \tag{36}$$

Step 2. Determine the parameters of the dividing surface between two adjacent subspaces.

According to Equation (4), the dividing surface of two adjacent subspaces can be described as: $h_{i,j} = \left\{w_{i,j}^T X + v_{i,j} = 0\right\}$. Suppose *y*(*i*) = 1 for the sample belonging to Ω_i subspace, and *y*(*j*) = −1 for the samples belonging to Ω_j subspace. Then, kernel matrix Ω in Equation (31) can be transformed by LS-SVM into:

$$\Omega_{i,j} = y_i y_j \varphi(X_i)^T \varphi(X_j) = y_i y_j K(X_i, X_j) \tag{37}$$

Without changing other parameters in Equation (31), *N* groups of α and *b* can be obtained by solving the new linear equation set. Finally, the dividing surface can be obtained by LS-SVM as:

$$f(x) = sgn\left[\sum_{i=1}^{N} \alpha_i y_i K(X, X_i) + b\right] \tag{38}$$

where, *N* = M_i + M_j is the total number of vectors in subspace Ω_i and subspace Ω_j.

Accordingly, the parameters of dividing surfaces can be obtained by:

$$w_{i,j} = \sum_{i=1}^{N} \alpha_i y_i X_i \ \text{ and } v_{i,j} = y_i - w_{i,j}^T X_i \tag{39}$$

Therefore, the parameters of dividing surfaces can be expressed as $h_{i,j} = \left[w_{i,j}^T, v_{i,j}\right]$.

2.6. Modeling Steps of the PWA Soft Sensor Model

Let *D* be the sample dataset of the fermentation process. The sample dataset is a kind of MISO Problem (1). Then, the model identification problem is transformed into an MIQP optimization Problem (8). The optimal solution of the MIQP problem can be obtained by the following steps:

(1) Step 1. Determining model order

The FNN is introduced to determine the embedding dimension in the state space reconstruction of nonlinear time series in the fermentation process. The false nearest neighbor can be identified by Equation (15). When the dimension n increases continuously, the ratio δ of the false nearest neighbor will approach a threshold. The dimension at this very moment is taken as the embedding dimension. The change of embedding dimension is evaluated by $E_0(n)$ in Equation (18). When the embedding dimension $n = n_y + n_u$ increases to a certain extent, $E_0(n)$ tends to be stable. At this moment, the n_y and n_u are the order of the corresponding model.

(2) Step 2. Collaborative optimization of the number and parameters of models

Because of their strong coupling, J_1 and J_2 must be determined synchronously. Therefore, the powerful parallel search ability of ICF-PSO algorithm is utilized to optimize the multi-objective optimization Problem (9). By the ICF-PSO, Equations (22) and (23) are updated, such that the particles can avoid the local optimum.

(3) Step 3. Determining dividing surfaces

The adjacent modeling subspaces $\{Z_i, Z_j\}$ are determined by the nearest neighbor rule of cluster center, and the dividing surfaces of adjacent subspaces are determined by $h_{i,j} = \left\{ w_{i,j}^T X + v_{i,j} = 0 \right\}$. By solving the new linear equation set, N groups of α and b are obtained. Then, the parameters $w_{i,j}^T$ and $v_{i,j}$ of dividing surfaces can be solved by Equation (38), namely, $h_{i,j} = \left[w_{i,j}^T, v_{i,j} \right]$.

The modeling process of the PWA algorithm is illustrated in Figure 4.

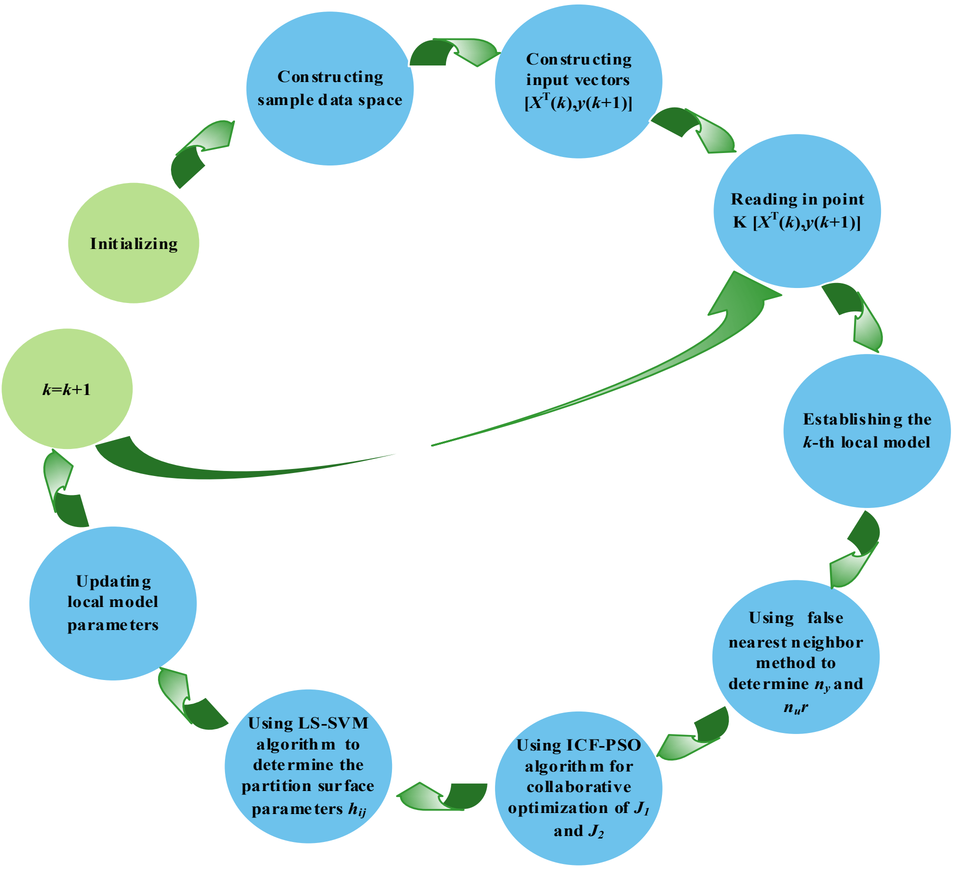

Constructing sample data space
Constructing input vectors $[X^T(k),y(k+1)]$
Reading in point K $[X^T(k),y(k+1)]$
Establishing the k-th local model
Using false nearest neighbor method to determine n_y and n_ur
Using ICF-PSO algorithm for collaborative optimization of J_1 and J_2
Using LS-SVM algorithm to determine the partition surface parameters h_{ij}
Updating local model parameters
k=k+1
Initializing

Figure 4. Flow chart of ICF-PSO-PWA soft sensor modeling algorithm.

2.7. Introduction of Experimental Work

The online control of *Pichia pastoris* fermentation is a difficult problem, due to the complexity of the fermentation process, and the cost of online detection instruments for key biomass variables being very high. To provide information for the online control and to optimize the fermentation process, it is significant to establish real-time soft sensor models for cell concentration X and protease K concentration P.

The sample data used in this paper all came from the microbial fermentation lab of Zhenjiang Yangzhong Weikete Bioengineering Equipment Co., Ltd. (Zhenjiang, China), and the fermentation experiment equipment adopted RTY-C−100 L model. Taking *Pichia pastoris* fermentation as the object, the *Pichia pastoris* KM71, Mut^S constructed by the Key Laboratory of Animal Husbandry and Veterinary Institute of Shanghai Academy of Agricultural Sciences, was selected as the strain, and the expression vector and foreign genes were pPICZαA and IFNαcDNA [51]. The experiment was carried out according to the schematic diagram of the *Pichia pastoris* fermentation process (Figure 5). After deeply analyzing the fermentation process, the temperature of fermentation broth T, dissolved oxygen concentration DO, pH of fermentation broth, air flow q, stirring speed v, and the pressure of fermentation tank p were selected as auxiliary variables. In order to clarify the auxiliary and key biomass variables of the ICF-PSO-PWA model at a glance, a model expression with a generalized form for the input and output variables is given in Equation (39).

$$\varphi(X, P) = f(T, Do, \mathrm{pH}, q, v, p) \tag{40}$$

where, $f(\mathrm{g})$ is a generalized form of the complex nonlinear relationship between the auxiliary variables in the ICF-PSO-PWA model; $\varphi(X, P)$ is any one of X and P. That is, the auxiliary variables (T, Do, pH, q, v, p) are the input variables of the model we use, and the key biomass variables (X, P) are the output variables we need.

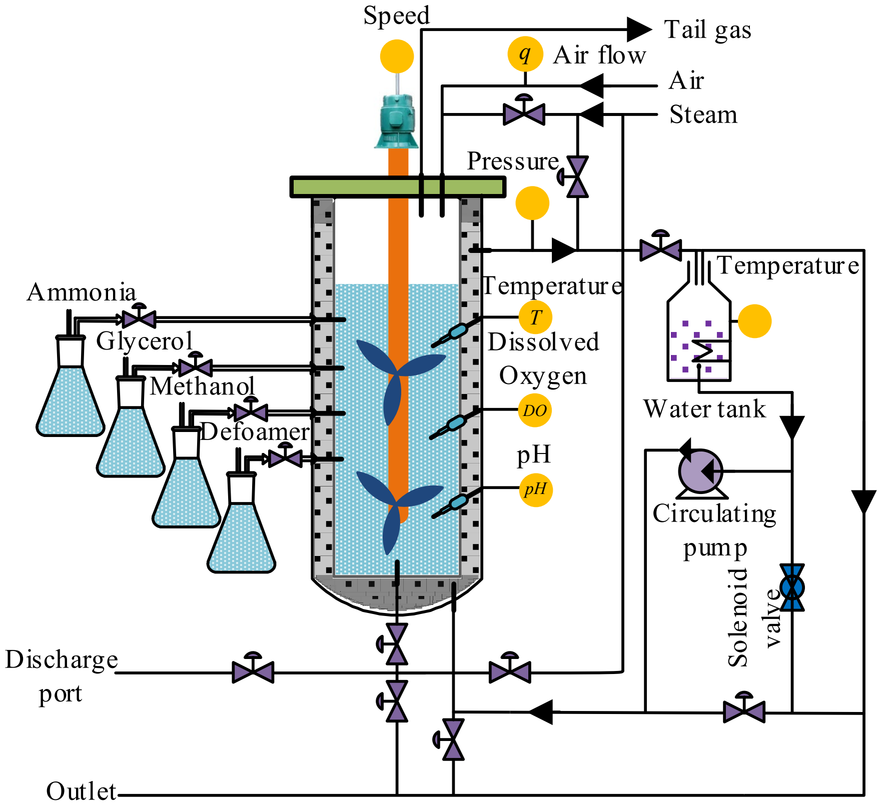


Figure 5. Schematic diagram of *Pichia pastoris* fermentation process.

The initial values of the input variables in the fermentation process were set as follows: the pressure of the fermentation tank was controlled at 0.04 Mpa, the stirring speed of the motor was about 250 rpm, the fermentation temperature was controlled at about 28 ± 0.5 °C, and the air flow was controlled at the range of 1000–1200 L/h. The oxygen content was maintained between 35% and 45%, and the pH was 7.3.

Under normal fermentation conditions, the environmental variables of the fermentation processes were monitored by sensors every 5 min and uploaded to the host computer. The cell concentrations and protease K concentrations were measured offline every 1 h. The cell concentration was measured by the American Lehman Lamotte 1200 photoelectric colorimeter, and the concentration of proteinase k was measured by the UV−260 model automatic spectrophotometer of Shimadzu Corporation, Japan. The adjustment, logarithmic growth and stabilization phases in the fermentation processes were selected as the data collection time period, which was approximately the first 80 h of the fermentation process. Finally, 10 batches of sample data were extracted to apply our algorithm. Subsequently, 6 out of these 10 batches were used for training, the 7th and 8th batch for online correction of the initial PWA model, and the 9th and 10th batch for verifying the effectiveness and prediction accuracy of the PWA model. As the output variables are measured offline once every 1 h and the input variables once every 5 min, there is no one-to-one correspondence between the two, so the interpolation methods are used to convert the output variables offline every 1 h to the input variables every 5 min, forming a one-to-one correspondence. The data are then pre-processed by a coordinate transformation, which normalizes all sample data to between −1 and 1, making the calculation easier and faster. Through coordinate transformation and interpolation operation on 10 batches of sample data, a total of 960 sets of sample data were obtained at last, including 576 sets of sample data in training set, 204 sets of sample data in verification set and 180 sets of sample data in test set.

3. Results and Discussion

After determining that the order of input vector (n_u) is 1 and the order of output vector (n_y) is 2, $X^T(k) = [u^T(k), y^T(k), y^T(k+1)]$ is selected to form the input vector and the local model is chosen to be of the form $y(k) = [y^T(k-1), u^T(k-1), 1]\theta$. The dividing surface equation is $h_{i,j}[y^T(k-1), u^T(k-1), 1]^T = 0$, which separates the i-th local model from the region where the j-th local model is located. After using the ICF-PSO algorithm to optimize the PWA model, the number of local models is $S = 3$, which exactly corresponds to the three phases of the sample data in this paper: adjustment, logarithmic growth and stable phase. The local model parameters θ_i ($i = 1, 2, 3$) and the parameters of dividing surfaces $h_{i,j}$ are shown in the Table 1.

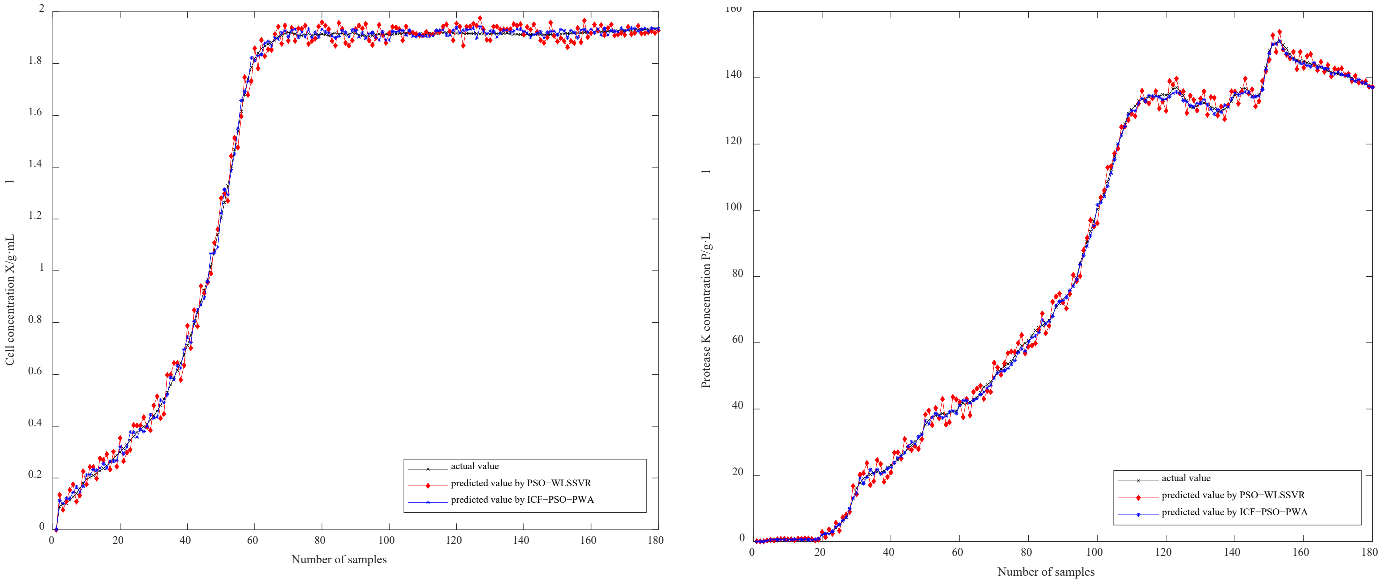
Table 1. Local moel modeling parameters.

	The Local Model Parameters	**The Parameters of Dividing Surfaces**
Local model I	θ_1 = [0.5957, −0.1847, −0.0964, 0.4447, 0.3635, 0.0464, 0.5567, −0.2888] ^{T}	h_{13} = [1, 1.0445, −0.0045, 0.9854, 0.5421, −0.2347, −1.0786, 0.0864]
Local model II	θ_2 = [0.0962, 0.9649, 0.0434, 0.0762, 0.9564, 0.0615, −0.6332, 1] ^{T}	
Local model III	θ_3 = [0.1629, 0.9885, −0.0003, 1, 0.1005, −0.6575, 0.3209, 0.6887] ^{T}	h_{23} = [1, −1.8753, 0.9753, −0.8656, 1.9531, 0.6542, −0.5423, 1.0112]

To verify its performance, the ICF-PSO-PWA model was compared with the PSO-WLSSVR model based on global modeling, which used Gaussian radial basis kernel (RBF) function $k(x_1, x_2) = \exp(-\|x_1 - x_x\|^2/\sigma^2)$, with σ = 0.2, and penalty coefficient C = 10. The WLS-SVR model introduces the least squares algorithm and weights on the basis of the SVR model, which greatly enhances the prediction accuracy and generalization ability of the SVR model. It is a very good comparator. All simulation results were obtained based on MATLAB R2019b software.

Figure 6a,b present the prediction results of soft sensor models based on ICF-PSO-PWA and PSO-WLSSVR, respectively. Obviously, the fitting effect of ICF-PSO-PWA soft sensor model is better than that of the PSO-WLSSVR model, regardless of the bacterial concentration or the concentration of protease K. In particular, the fitting effect of the

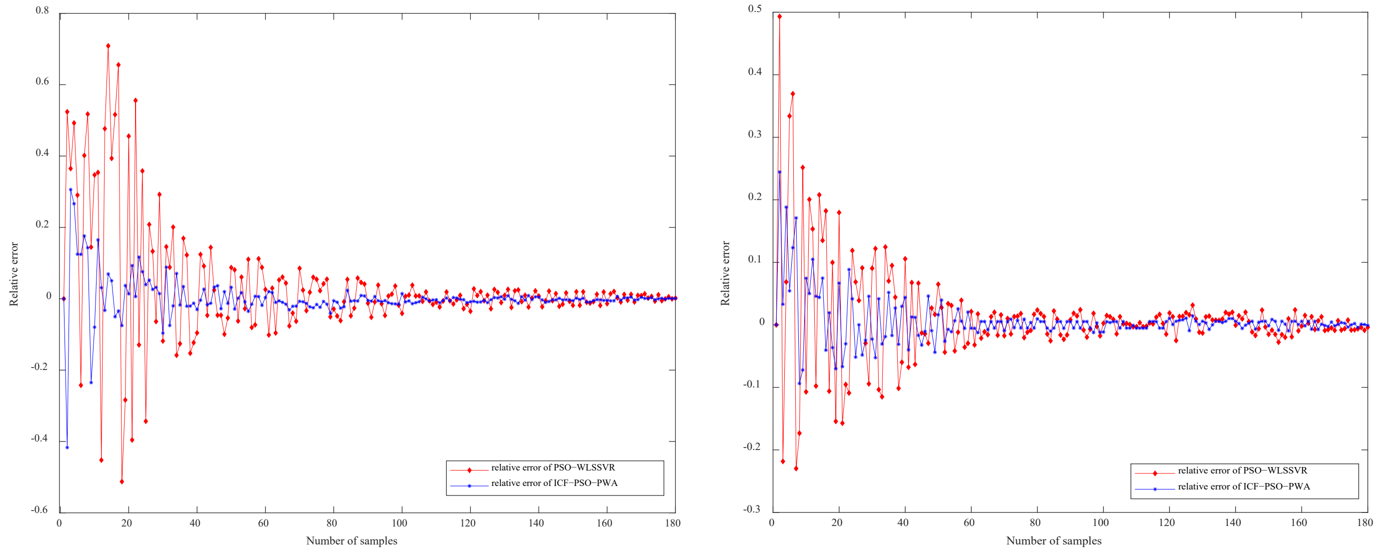
PSO-WLSSVR model is clearly inferior to the ICF-PSO-PWA model in terms of the cell concentration in the later stage.



(**a**) (**b**)

Figure 6. Comparison curve between the soft sensor value and the actual value of (**a**) cell concentration and (**b**) protease K concentration.

Figure 7a,b compare the relative error of cell concentration and protease K concentration based on ICF-PSO-PWA and PSO-WLSSVR, respectively. For both ICF-PSO-PWA and PSO-WLSSVR soft sensor models, the prediction errors gradually decrease with time and are eventually stabilized. However, the ICF-PSO-PWA soft sensor model has a much smaller error range, a relatively high accuracy in the whole sample interval, and an obviously smaller final error than the PSO-WLSSVR model.



(**a**) (**b**)

Figure 7. Comparison curve of relative error of (**a**) cell concentration and (**b**) protease K concentration.

Table 2 shows the mean absolute percentage error (MAPE) of cell concentration and proteinase K concentration based on ICF-PSO-PWA and PSO-WLSSVR. It can be seen from

Table 2 that compared with the PSO-WLSSVR soft sensor model, the ICF-PSO-PWA soft sensor model has an improved accuracy by 2.4642% when predicting the cell concentration, and an improved accuracy by 6.5127% when predicting the proteinase K concentration. The mean prediction accuracy of ICF-PSO-PWA soft sensor model has been improved by 4.4884%.

Table 2. Comparison of prediction accuracy of soft sensor models.

	PSO-WLSSVR	**ICF-PSO-PWA**	**Accuracy Improvement**
MAPE of cell concentration	0.043573488	0.018931913	2.4642%
MAPE of protease K concentration	0.091100816	0.025974078	6.5127%
Mean accuracy improvement		4.4884%	

Furthermore, the root mean square errors (RMSEs) of the ICF-PSO-PWA and PSO-WLSSVR soft sensor models were tested on different scale datasets. The test results in Table 3 show that, with the growing number of test samples, the RMSE of the PSO-WLSSVR soft sensor model increases, while that of the ICF-PSO-PWA model does not change significantly. This is because the latter model divides the working interval of the system into several subintervals, and models each subinterval separately; besides, multiple local models are used in place of the global single model, making full use of the information of each sample. As time goes by, the ICF-PSO-PWA model can adapt to the changing working conditions, and achieve excellent generalization performance and self-adaptability.

Table 3. RMSE of different test sample sets.

Number	**PSO-WLSSVR**		**ICF-PSO-PWA**	
	X	***P***	***X***	***P***
20	0.034753378	0.242421228	0.013863265	0.057610602
40	0.041227412	1.726108115	0.017189875	0.537857149
60	0.043646458	2.086190315	0.021352118	0.598752504
80	0.041167894	2.272430942	0.019573828	0.635771991
100	0.039357834	2.402638746	0.018463627	0.710257895
120	0.036679143	2.35714544	0.017360173	0.70555966
140	0.035563714	2.38677746	0.016815066	0.745174526
160	0.035008022	2.372930488	0.016138098	0.716623673
180	0.033457354	2.278762339	0.015402352	0.711287476

4. Conclusions

In order to solve the problem that the key biological variables in the fermentation process of *Pichia pastoris* are difficult to directly measure online, a PWA multi-model soft sensor modeling method based on the idea of "decomposition-synthesis" was proposed to measure the key variables in the fermentation process of *Pichia pastoris* online. In order to identify the PWA multi-model, the FNN was used to solve the order of the input and output vectors of the model. Then, a novel ICF-PSO algorithm was proposed for collaborative optimization of the number of local models and the parameters of each local model. Finally, the LS-SVM was used for the determination of the parameters of the dividing surfaces in the modeling process, so as to delineate the regions where each local model plays a role. MATLAB simulation results show that the prediction accuracy of the ICF-PSO-PWA model for X and P reach 1.8932% and 2.5974%, respectively, compared with 4.3573% and 9.1101% of the prediction accuracy of the PSO-WLSSVR model for X and P, the prediction accuracy of the ICF-PSO-PWA model improved by 2.4642% and 6.5127%. The RMSEs of the ICF-PSO-PWA model for X and P prediction results remain at 0.015402352 and 0.711287476, respectively, which are smaller than 0.033457354 and 2.278762339 of the PSO-WLSSVR model, indicating that the prediction results of the ICF-PSO-PWA model are more stable. The simulation results above demonstrate the effectiveness of the ICF-PSO-PWA soft sensor method in predicting the key biological variables in the fermentation process of *Pichia pastoris*. The proposed model provides a feasible theoretical approach to solve the soft sensing of key biological variables in the *Pichia pastoris* fermentation process.

The PWA modeling method, as a special form of modeling method, partitions the nonlinear complex system into several linear systems, so that the system can be solved using the existing mature linear theory, breaking through the bottleneck that the traditional single model is extremely complex and cannot meet the needs of actual engineering when the global description of increasingly complex controlled objects is performed. The strong nonlinear approximation capability and simple model structure of PWA are not only loved by scholars in the field of modeling, but also a hot research topic in the control community. In the future, we are interested in studying the optimal control of PWA models and the design of online monitoring systems for the whole fermentation process.

Author Contributions: Conceptualization, X.W.; methodology, X.W.; software, X.W.; validation, X.W. and B.W.; formal analysis, X.W.; investigation, X.W. and M.H.; resources, B.W.; data curation, B.W. and X.W.; writing—original draft preparation, X.W.; writing—review and editing, B.W. and M.H.; visualization, X.W. and B.W.; supervision, B.W. and X.Z.; project administration, B.W.; funding acquisition, B.W. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was carried out by the National Natural Science Foundation of China (Grant No. 41376175), The Innovative Research Group Project of the National Natural Science Foundation of China (CN) (Grant No. BK20140568), and The key R & D project of Zhenjiang City (Grant No. (SH2020005)).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are thankful to the Key Laboratory of Agricultural Measurement and Control Technology and Equipment for Mechanical Industry Facilities, School of Electrical and Information Engineering, Jiangsu University, for their technical support of our experiment. The authors are also thankful to the anonymous reviewers whose comments helped improve and clarify.

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

Appendix A. Performance Analysis of ICF-PSO Algorithm

In order to study the influence of the algorithm improvement scheme on the algorithm performance, this paper selects six well-known classic test functions for verification. Since the ICF-PSO algorithm is used for the collaborative optimization of the number of PWA models and the parameters of PWA sub-models, three classic test functions for multi-objective optimization problems similar to the PWA model proposed in this paper are selected in the test function. The expressions of the functions are shown in Table A1, where $f_2(x$ (De Jong's f2), $f_4(x)$ (Shaffer's f6) and $f_5(x)$ (Levy N.13) are test functions for multi-objective optimization problems. This is because both values of x_1 and x_2 must be searched for simultaneously in order to determine the final minimum values of $f_2(x)$, $f_4(x)$ and $f_5(x)$. The parameter settings of the 6 classic test functions are shown in Table A2.

Table A1. Functions used to test the performance of different optimization algorithms.

Function Name	Expression
Ackley	$f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{d=1}^{D} x_d^2}\right) - \exp\left(\frac{1}{D}\sum_{d=1}^{D}\cos(2\pi x_d)\right) + 20 + e$
De Jong's f2	$f_2(x) = 100\left(x_1^2 - x_2\right)^2 + (1 - x_1)^2$
Griewank	$f_3(x) = \frac{1}{4000}\sum_{d=1}^{D} x_d^2 - \prod_{d=1}^{D}\cos\left(\frac{x_d}{d}\right) + 1$
Shaffer's f6	$f_4(x) = 0.5 + \frac{\sin\sqrt{x_1^2+x_2^2} - 0.5}{\left(1.0+0.001\left(x_1^2+x_2^2\right)\right)^2}$
Levy N.13	$f_5(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2\left[1 + \sin^2(3\pi x_2)\right] + (x_2 - 1)^2\left[1 + \sin^2(2\pi x_2)\right]$
Rosenbrock	$f_6(x) = \sum_{d=1}^{D}\left(100\left(x_{d+1} - x_d^2\right)^2 + (x_d - 1)^2\right)$

Table A2. Function parameters for the test problems.

Function	Dimension	Initial Range of x_i	Global Minimum
Ackley	10	±32.768	0
De Jong's f2	2	±50	0
Griewank	10	±300	0
Shaffer's f6	2	±100	0
Levy N.13	2	±10	0
Rosenbrock	10	±15	0

In order to further verify the performance of the ICF-PSO algorithm, compare it with the standard PSO, Bat Algorithm (BA) and Firefly Algorithm (FA) [52,53]. To facilitate quick comparison of performance, setting the maximum number of iterations to 100 is enough, and calculate the average (AVE) and standard deviation (SD) of the calculation results. The comparison results are shown in Table A3, and the convergence curve of each algorithm test function is shown in Figure A1.

Table A3. Function test results.

Function	ICF-PSO		PSO		BA		FA	
	AVE	SD	AVE	SD	AVE	SD	AVE	SD
$f_1(x)$	0.35633	1.70326	0.60577	2.45382	0.70429	2.00445	1.24317	2.57169
$f_2(x)$	0.21804	0.79072	0.46414	1.04782	0.96809	1.82061	1.08155	2.41802
$f_3(x)$	0.08603	0.18551	0.15912	0.37573	0.11013	0.22398	0.14648	0.24683
$f_4(x)$	0.04314	0.12191	0.08433	0.09664	0.09812	0.09255	0.06048	0.08848
$f_5(x)$	0.83762	2.86836	2.74258	7.55932	3.10593	8.56497	1.97337	5.36548
$f_6(x)$	0.32771	3.19435	0.73393	4.53503	1.28882	6.83131	1.95052	7.77723

It can be seen from the test results in Table A3 that the AVE and SD of ICF-PSO are significantly smaller than the standard PSO, BA and FA, indicating that ICF-PSO can find the minimum zero stably. The optimization accuracy of ICF-PSO is also significantly higher than the other three algorithms. Summarized from the six curves in Figure A1, the convergence curve of ICF-PSO is obviously below the other curves, indicating that ICF-PSO has stronger optimization ability and the convergence speed is significantly higher than the other three algorithms, especially in the Figure A1e, ICF-PSO quickly converges to the minimum in less than ten steps, and the convergence speed is at least twice that of the other three algorithms, and the stability is stronger. It can also be seen in Figure A1b,d that the convergence speed of ICF-PSO is at least ten steps ahead of other algorithms, and the stability is stronger.

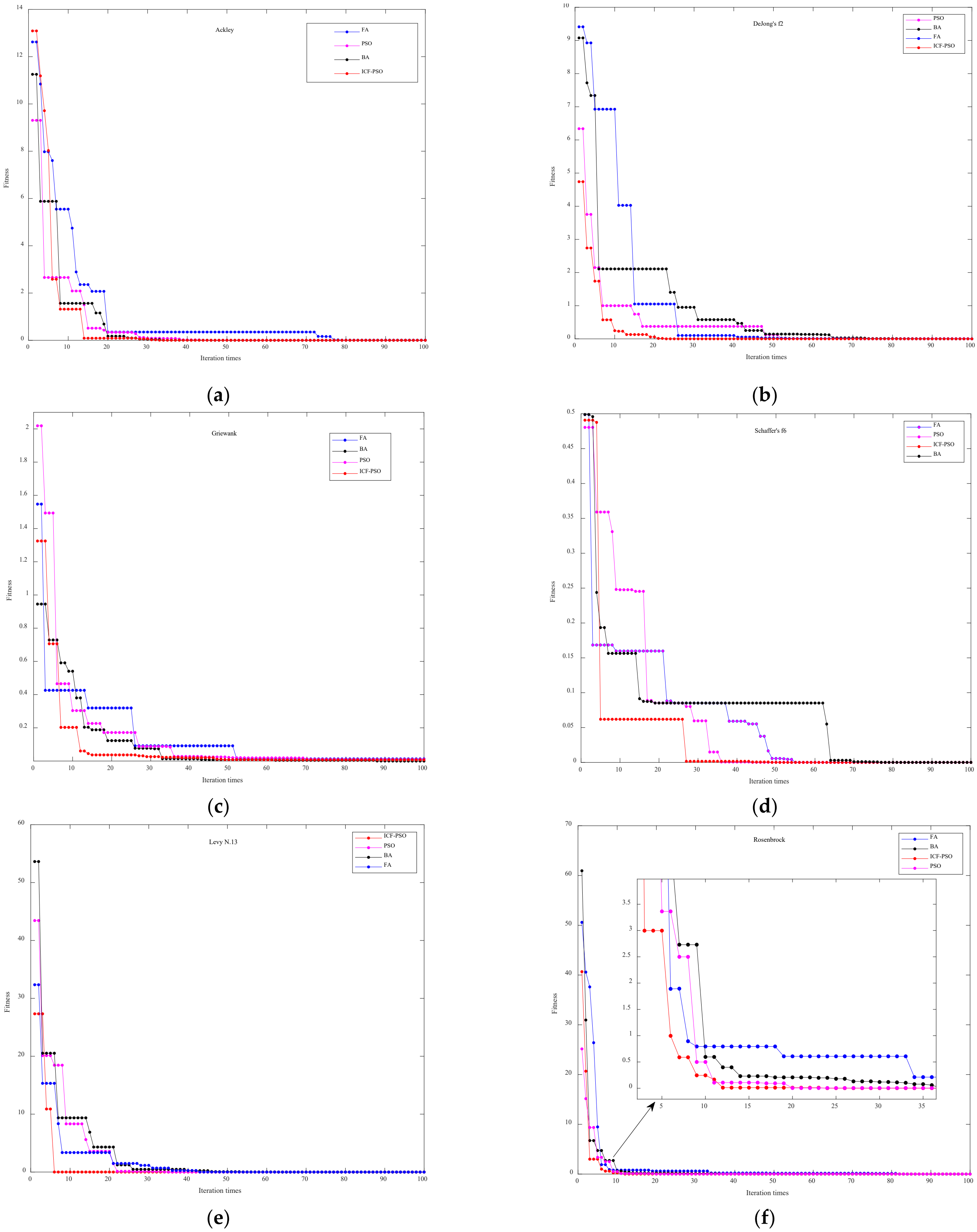
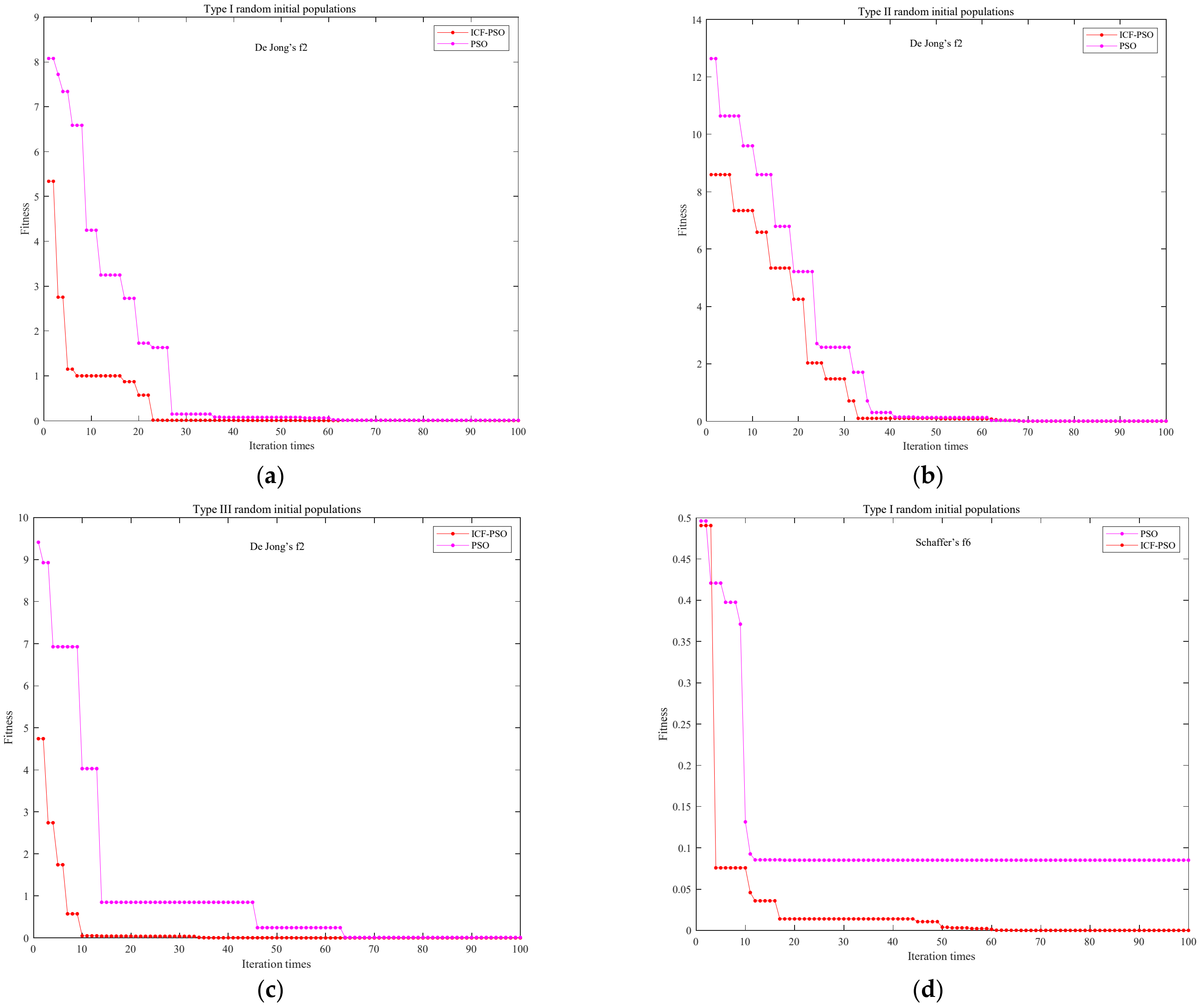


Figure A1. The fitness curves of different optimization algorithms on different test functions (**a**) Ackley; (**b**) De Jong's f2; (**c**) Griewank; (**d**) Schaffer's f6; (**e**) Levy N.13; (**f**) Rosenbrock.

Besides, as can be seen from Figure A1 and Table A3, the PSO algorithm performs second only to ICF-PSO in terms of the outperformance of these six test functions. Therefore, in order to assert that the ICF-PSO algorithm performs better, converges faster, and is more robust, the PSO algorithm is again compared with the ICF-PSO algorithm. De Jong's f2 and Shaffer's f6 were chosen as the test functions and three different initial populations were chosen for each function, i.e., the optimization performance of the two optimization algorithms was compared separately for different initial populations of each test function. The comparison results are shown in Table A4, and the convergence curve of each algorithm test function is shown in Figure A2.

Table A4. Function test results.

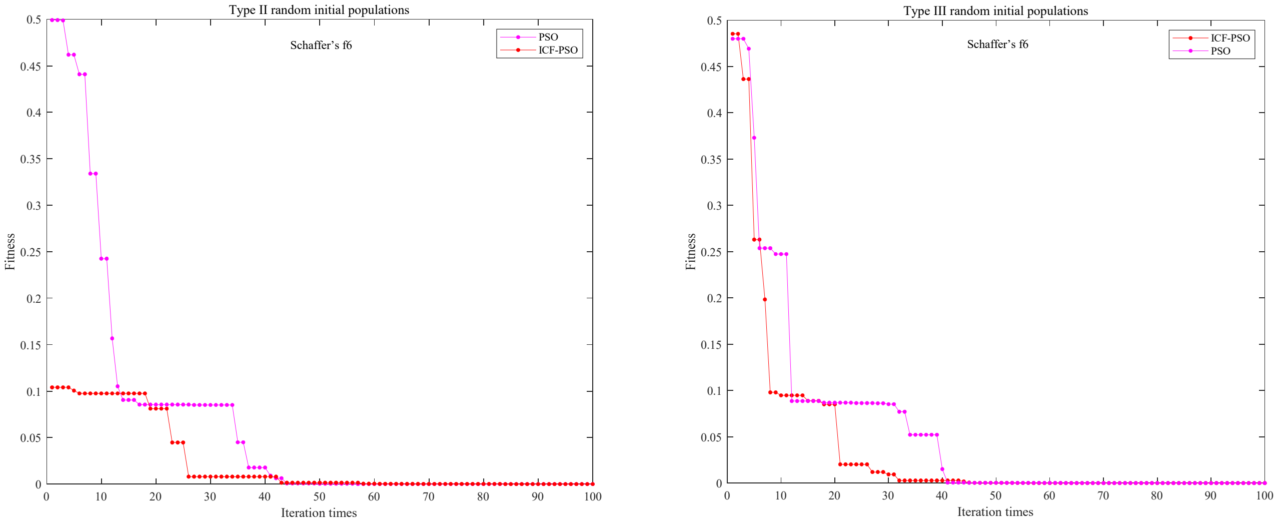
Function	Initial Populations	ICF-PSO		PSO	
		AVE	SD	AVE	SD
De Jong's f2	Type I	0.33380	0.87704	1.10774	2.13176
	Type II	1.58682	2.72450	2.25954	3.64831
	Type III	0.21251	0.79077	1.16916	2.20947
Shaffer's f6	Type I	0.02700	0.08391	0.11621	0.09779
	Type II	0.02409	0.03907	0.06715	0.12842
	Type III	0.03974	0.09922	0.06017	0.11168



(**a**) (**b**)

(**c**) (**d**)

Figure A2. *Cont.*



(**e**) (**f**)

Figure A2. The fitness curves optimized for De Jong's f2 under three different initial populations (**a**) Type I; (**b**) Type II; (**c**) Type III; The fitness curves optimized for Schaffer's f6 under three different initial populations (**d**) Type I; (**e**) Type II; (**f**) Type III.

It can be seen from the test results in Table A4 that the AVE and SD of ICF-PSO are significantly smaller than the standard PSO, indicating that ICF-PSO can find the minimum stably. The optimization accuracy of ICF-PSO is also significantly higher than the PSO algorithm. Summarized from the six curves in Figure A2, the convergence curve of ICF-PSO is obviously below the curves of PSO, indicating that ICF-PSO has stronger optimization ability and the convergence speed is significantly higher than the PSO algorithm.

Therefore, considering the above performance comparison results, the ICF-PSO algorithm with better convergence speed, convergence accuracy and convergence stability was chosen to optimize the PWA model in this paper.

The ICF-PSO algorithm can be implemented in the following steps:

(1) Step 1. Initialize the particle swarm.
(2) Step 2. Calculate the initial fitness of each particle by the objective function *F* and evaluate particle fitness.
(3) Step 3. Update the position and velocity of each particle in the swarm by Equations (22) and (23), respectively.
(4) Step 4. Continue to evaluate the initial fitness of each particle according to the objective function *F*.
(5) Step 5. Terminate the algorithm if the termination condition is met; otherwise, jump back to Step 3.

References

1. Hori, K.; Okano, S.; Sato, F. Efficient microbial production of stylopine using a Pichia pastoris expression system. *Sci. Rep.* **2016**, *6*, 22201. [CrossRef] [PubMed]
2. Jie, W.; Zhang, X.X.; Yu, H.S.; Li, W.; Jia, Y.P.; Guo, J.Y.; Zhang, L.J.; Song, X.B. Research progress of high density fermentation process of pichia pastoris. *China Biotech.* **2016**, *36*, 108–114.
3. Kalinina, A.N.; Gordeeva, T.L.; Sineoky, S.P. Expression of the xylanase gene from paenibacillus brasilensis X1 in pichia pastoris and characteristics of the recombinant enzyme. *Appl. Biochem. Microbiol.* **2019**, *55*, 797–804. [CrossRef]
4. Wang, L.; Hu, T.; Jiang, Z.; Yan, Q.; Yang, S. Efficient production of a novel alkaline cold-active phospholipase C from Aspergillus oryzae by molecular chaperon co-expression for crude oil degumming. *Food Chem.* **2021**, *350*, 129212. [CrossRef] [PubMed]
5. Chiniadis, L.; Bratsos, I.; Bethanis, K.; Karpusas, M.; Giastas, P.; Papakyriakou, A. High-resolution crystal structures of a "half sandwich"-type Ru(II) coordination compound bound to hen egg-white lysozyme and proteinase K. *JBIC J. Biol. Inorg. Chem.* **2020**, *25*, 635–645. [CrossRef] [PubMed]

6. Han, M.; Wang, W.; Zhou, J.; Gong, X.; Xu, C.; Li, Y.; Li, Q. Activation of the unfolded protein response via co-expression of the HAC1i gene enhances expression of recombinant elastase in pichia pastoris. *Biotechnol. Bioprocess. Eng.* **2020**, *25*, 302–307. [CrossRef]
7. Yao, Z.; Wang, J.H.; Liu, X.Q. Selective quenching detection of proteinase K by croconaine based organic sensor. *World Sci. Res. J.* **2021**, *7*, 472–479.
8. Martens, T. Urgent cardiac surgery and COVID-19 infection: Uncharted territory: Reply. *Ann. Thorac. Surg.* **2021**, *111*, 1735. [CrossRef]
9. Babbal, A.; Mohanty, S.; Khasa, Y.P. Nitrogen supplementation ameliorates product quality and quantity during high cell density bioreactor studies of Pichia pastoris: A case study with proteolysis prone streptokinase. *Int. J. Biol. Macromol.* **2021**, *180*, 760–770. [CrossRef]
10. Abdulrachman, D.; Thongkred, P.; Kocharin, K.; Nakpathom, M.; Somboon, B.; Narumol, N.; Champreda, V.; Eurwilaichitr, L.; Suwanto, A.; Nimchua, T.; et al. Heterologous expression of Aspergillus aculeatus endo-polygalacturonase in Pichia pastoris by high cell density fermentation and its application in textile scouring. *BMC Biotechnol.* **2017**, *17*, 15. [CrossRef]
11. Hu, X.Q.; Chu, J.; Zhang, S.L.; Zhuang, Y.P.; Wu, X.; Chen, H.X.; Lv, Z.Y.; Yuan, Z.Y. An alkaline pH control strategy for methionine adenosyltransferase production in Pichia pastoris fermentation. *Biotech. Bioprocess. Eng.* **2014**, *19*, 900–907. [CrossRef]
12. Adem, S.; Jain, S.; Sveiven, M.; Zhou, X.H.; O'Donoghue, A.J.; Hall, D.A. Giant magnetoresistive biosensors for real-time quantitative detection of protease activity. *Sci. Rep.* **2020**, *10*, 7941. [CrossRef]
13. Alakent, B. Soft-sensor design via task transferred just-in-time-learning coupled transductive moving window learner. *J. Process. Control* **2021**, *101*, 52–67. [CrossRef]
14. Wang, M.C.; Han, X.; Sun, Y.M.; Sun, Q.Y.; Chen, X.G. Study on soft sensor modeling method for sign of contaminated fermentation broth in Chlortetracycline fermentation process. *Prep. Biochem. Biotechnol.* **2021**, *51*, 76–85. [CrossRef]
15. López-Martínez, J.; Arzola-Sotelo, E.A.; Nevárez-Martínez, M.O.; Álvarez-Tello, F.J.; Morales-Bojórquez, E. Modeling growth on the cannonball jellyfish Stomolophus meleagris based on a multi-model inference approach. *Hydrobiologia* **2020**, *847*, 1399–1422. [CrossRef]
16. Chakraborty, S.; Jain, S.; Nandy, A.; Venture, G. Pathological gait detection based on multiple regression models using unobtrusive sensing technology. *J. Signal Process. Syst.* **2021**, *93*, 1–10. [CrossRef]
17. Wang, B.; Yu, M.F.; Zhu, X.L.; Zhu, L. Soft-sensing modeling based on ABC-MLSSVM inversion for marine low-temperature alkaline protease MP fermentation process. *BMC Biotech.* **2020**, *20*, 343–354. [CrossRef]
18. Lu, B.; John, S.; Thomas, E.F. Data-driven adaptive multiple model system utilizing growing self-organizing maps. *J. Process. Control* **2018**, *67*, 56–68. [CrossRef]
19. Zhang, W.G.; Cai, K. Soft sensor modeling of alkaline protease fermentation process based on improved LWPLS. *Transducer Microsyst. Technol.* **2020**, *39*, 108–114.
20. Zhu, X.J.; Song, C.Y.; Zhao, J.; Xu, Z.H. A Semi-continuous PWA model based optimal control method for nonlinear systems. *Proceses* **2020**, *8*, 170. [CrossRef]
21. Sun, X.Q.; Hu, W.W.; Wu, P.C.; Wong, P.K.; Chen, L. Modeling of tire longitudinal-slip mechanical characterisics based on piecewise affine identification method. *J. Xi'an Jiaotong Univ.* **2021**, *55*, 52–60.
22. Wang, J.-H. Zonotope parameter identification for piecewise affine system. *WSEAS Trans. Power Syst.* **2020**, *19*, 36–46. [CrossRef]
23. Sun, X.; Wu, P.; Cai, Y.; Wang, S.; Chen, L. Piecewise affine modeling and hybrid optimal control of intelligent vehicle longitudinal dynamics for velocity regulation. *Mech. Syst. Signal. Process.* **2021**, *162*, 108089. [CrossRef]
24. Muhammad, M.; Michael, S.; Michael, Z.; Klaus, L.; Hans, A.; Armin, C. Optimal planning of thermal energy systems in a microgrid with seasonal storage and piecewise affine cost function. *Energy* **2021**, *215*, 119095.
25. Sindareh-Esfahani, P.; Pieper, J. Machine learning–based piecewise affine model of wind turbines during maximum power point tracking. *Wind. Energy* **2019**, *23*, 404–422. [CrossRef]
26. Mattsson, P.; Zachariah, D.; Stoica, P. Identification of cascade water tanks using a PWARX model. *Mech. Syst. Signal Process.* **2018**, *106*, 40–48. [CrossRef]
27. Muhammad, A.R.; Reham, R.M.; Ozgur, K.; Mundher, Y.Z.; Shamsuddin, S.; Mohammad, Z.K. Improving streamflow prediction using a new hybrid ELM model combined with hybrid particle swarm optimization and grey wolf optimization. *Knowl. Based Syst.* **2021**, *230*, 107379.
28. Han, K.; Huang, T.; Yin, L. Quantum parallel multi-layer Monte Carlo optimization algorithm for controller parameters optimization of doubly-fed induction generator-based wind turbines. *Appl. Soft Comput.* **2021**, *112*, 107813. [CrossRef]
29. Yuan, J.; Xie, J.; Huang, M.; Fan, H.; Feng, E.; Xiu, Z. Robust optimal control problem with multiple characteristic time points in the objective for a batch nonlinear time-varying process using parallel global optimization. *Optim. Eng.* **2020**, *21*, 905–937. [CrossRef]
30. Jiang, X.; Zhang, X.; Zhang, Y. Establishment and optimization of sensor fault identification model based on classification and regression tree and particle swarm optimization. *Mater. Res. Express* **2021**, *8*, 085703. [CrossRef]
31. Hu, X.; Chen, F.; Zhang, Z. Model construction and optical properties investigation for multi-sectioned compound parabolic concentrator with particle swarm optimization. *Renew. Energy* **2021**, *179*, 379–394. [CrossRef]
32. Ren, M.; Song, Y.; Chu, W. An improved locally weighted pls based on particle swarm optimization for industrial soft sensor modeling. *Sensors* **2019**, *19*, 4099. [CrossRef] [PubMed]

33. Javed, M.S.; Ma, T.; Jurasz, J.; Ahmed, S.; Mikulik, J. Performance comparison of heuristic algrithms for optimization of hybrid offgrid renewable energy systems. *Energy* **2020**, *210*, 118599. [CrossRef]
34. Wihartiko, F.D.; Wijayanti, H.; Virgantari, F. Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *332*, 012020. [CrossRef]
35. Pandey, H.M. Performance review of harmony search, differential evolution and particle swarm optimization. *IOP Conf. Ser. Mater. Sci. Eng.* **2017**, *225*, 012221. [CrossRef]
36. Liu, Y.D.; Zhou, M.R.; Xie, J.Y.; Tong, Y.L. Research on simulation model for data assimilation in solar radiation prediction based on particle swarm optimizer with time varying constrict factor. *Acta Ener. Sol. Sin.* **2021**, *42*, 181–185.
37. Zhang, C.X. Comprehensive informed particle swarm optimizer based on constrict factor. *J. Front. Comput. Sci. Technol.* **2014**, *8*, 506–512.
38. Wu, Z.; Zhang, S.; Wang, T. A cooperative particle swarm optimization with constriction factor based on simulated annealing. *Computing* **2018**, *100*, 861–880. [CrossRef]
39. Olfa, Y.; Zeineb, L.; Kamel, A. Predictive control based on fuzzy supervisor for PWARX hybrid model. *Int. J. Autom. Comput.* **2019**, *16*, 683–695.
40. Maximo, M.R.O.A.; Afonso, R.J.M. Mixed-integer quadratic programming for automatic walking footstep placement, duration, and rotation. *Optim. Control Appl. Methods.* **2020**, *41*, 1928–1963. [CrossRef]
41. Han, M.; Ren, W.J.; Li, B.S.; Feng, S.B. Review of chaotic time series analysis and prediction. *Inf. Control* **2020**, *49*, 24–35.
42. Vignesh, R.; Jothiprakash, V.; Sivakumar, B. Streamflow variability and classification using false nearest neighbor method. *J. Hydrol.* **2015**, *531*, 706–715. [CrossRef]
43. Zhao, L.; Fang, J.; Wang, Y.; Huang, Z.M. Output state analysis and classification of semiconductor fiber ring lasers. *Acta Photonica Sin.* **2009**, *38*, 2449–2457.
44. Jia, Q.L. Time series prediction model of power load based on probability statistics. *Mod. Electron. Technol.* **2020**, *43*, 179–186.
45. Xia, B.; Liu, T.; Ding, T.; Wang, Z. An improved PSO localization algorithm for UWB sensor networks. *Wirel. Pers. Commun.* **2020**, *117*, 2207–2223. [CrossRef]
46. Huang, G.; Cai, Y.; Cai, H. A time varying constrict factor PSO algorithm research. *J. Comput. Methods Sci. Eng.* **2018**, *18*, 725–735. [CrossRef]
47. Zhang, Z.; Wang, Y.P. Particle swarm optimization algorithm with compression factor for complex shading MPPT problem. *Mod. Electron. Tech.* **2021**, *44*, 102–105.
48. Zhang, H.; Jin, Y. Particle swarm cooperative optimization algorithm based on geometric algebra. *J. Discret. Math. Sci. Cryptogr.* **2017**, *20*, 913–930. [CrossRef]
49. Zhu, X.L.; Chen, W.; Ding, Y.H.; Wang, B.; Zhu, L.; Jiang, Z.Y. Soft sensing of photosynthetic bacteria fermentation based on IBA-LSSVM. *Comput. Meas. Control.* **2019**, *27*, 41–54.
50. Zhu, X.L.; Hua, T.Z. Soft sensor modeling of straw fermentation process based on chaos drosophila optimized least squares support vector machine. *J. Beijing Univ. Technol.* **2016**, *42*, 1468–1474.
51. Gao, M.J.; Li, Z.; Yu, R.S. Methanol/sorbitol co-feeding induction enhanced porcine interferon-α production by P. pastoris asso-ciated with energy metabolism shift. *J. Bioprocess. Biosyst. Eng.* **2020**, *35*, 1125–1136. [CrossRef] [PubMed]
52. Cui, Y.B.; Huang, C.C.; Cui, Y.P. A novel compound wind speed forecasting model based on the back propagation neural network optimized by bat algorithm. *Environ. Sci. Pollut. Res.* **2020**, *27*, 7353–7365. [CrossRef] [PubMed]
53. Jalili, A.; Taheri, B. Optimal sizing and sitting of distributed generations in power distribution networks using firefly algorithm. *Technol. Econ. Smart Grids Sustain. Energy* **2020**, *5*, 33–44. [CrossRef]