International Journal of Innovative Computing, Information and Control
Volume **15**, Number **5**, October **2019**
 ISSN 1349-4198
pp. 1851–1864

TWIN SUPPORT VECTOR REGRESSION BASED ON FRUIT FLY OPTIMIZATION ALGORITHM

Jianwen Fang, Feng Pan and Binjie Gu*

Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education
Jiangnan University
No. 1800, Lihu Road, Wuxi 214122, P. R. China
*Corresponding author: gubinjie1980@126.com

Received January 2019; revised May 2019

Abstract. *Recently, Twin Support Vector Regression (TSVR), which determines a pair of ε-insensitive lower and upper bound functions by solving two related SVR-type problems, has become a new hot topic in machine learning field. However, at least four parameters should be appropriately specified in TSVR. In this paper, in order to obtain the optimal parameters of TSVR, we proposed a twin support vector regression based on fruit fly optimization algorithm. First, we represented the parameters to be optimized in TSVR by the locations of the fruit flies. Then, we used fitting regression precision as fitness function, and let fruit flies fly randomly to avoid trapping into local minimum. Finally, we could find the highest regression accuracy corresponding to the final position of the fruit flies within finite iterations. The experimental results on benchmark datasets and glutamic acid fed-batch fermentation process show that the proposed algorithm can be used to find suitable parameters for TSVR. Furthermore, our algorithm costs less optimization time than other state-of-the-art algorithms.*
Keywords: Twin support vector regression, Fruit fly optimization algorithm, Fitness function, Parameter selection

1. **Introduction.** Support Vector Machine (SVM) is a classical machine learning method based on statistical learning theory, and it has been successfully used to solve the problem of local maximum and dimensionality curse in classification and regression [1-5]. In general, the training time of SVM will increase rapidly with the scale of datasets. In order to shorten the training time and keep the generalization ability of SVM as much as possible, Mangasarian and Wild proposed a proximal support vector machine based on generalized eigenvalues, which optimized each hyper-plane as close as possible to its own class of samples and as far as possible from other class of samples [6]. Based on the work of Mangasarian and Wild, Jayadeva et al. proposed a Twin Support Vector Machine (TSVM), which converted the primal problems into small-scale dual problems. The training time of TSVM was only a quarter of corresponding SVM, and TSVM had shown strong generalization ability in classification [7-11]. Then, Peng proposed Twin Support Vector Regression (TSVR), which generalized the idea of TSVM to regression. Compared with the traditional support vector regression, TSVR not only had better fitting regression results but also cost less training time [12]. Based on the work of Peng, many researchers proposed related variants of TSVR [13-16]. They also obtained better results than TSVR in UCI datasets.

However, we should select several parameters first before training. The fitting regression results will deteriorate if the parameters are inappropriately selected. Therefore, the

DOI: 10.24507/ijicic.15.05.1851

parameter selection is very important for obtaining precise and stable regression model. In SVM, we should appropriately select the penalty parameter C and the insensitive loss function parameter ε. If the penalty parameter C is small, the penalty for the training samples exceeding ε will weaken, and the training error will become large. Conversely, if the penalty parameter C is large, the learning precision will improve, while the generalization capability will worsen. If the insensitive loss function parameter ε is small, the fitting regression precision will improve, while the model will become more complicated. On the contrary, if the insensitive loss function parameter ε is large, the number of support vectors will decline, which will cause lower learning precision. In TSVR, we should also appropriately select the penalty parameters C_1, C_2 and the insensitive loss function parameters ε_1, ε_2, respectively. If the penalty parameter C_1 or C_2 is small, the distance between the fitting curve and training samples will be short, so that the two hyper-planes are too close to each other, which will lead to bad fitting regression results. Conversely, if the penalty parameter C_1 or C_2 is large, the distance between training samples and their corresponding hyper-planes will be ignored by the fitting curve, which will also lead to bad fitting regression results. The effect of insensitive loss function parameters ε_1 and ε_2 are similar to those of the penalty parameters C_1 and C_2. In summary, the parameters of SVM and TSVR should not neither be too small nor too large, and they can be selected in a limited range.

At present, there are still no general guidelines for selecting the parameters of an SVM or TSVM model. The most commonly used method for parameter selection is grid search algorithm, which is based on empirical rules. However, the grid search algorithm is always time-consuming, and it costs rather long time to obtain the optimal parameters. Recently, many researchers attempt to use swarm intelligent algorithms to replace grid search algorithm. The swarm intelligent algorithms are all inspired by the foraging behaviour of natural phenomena, including Genetic Algorithm (GA) [17], Particle Swarm Optimization algorithm (PSO) [18], and Fruit fly Optimization Algorithm (FOA) [19]. The population size and the number of iterations are two important parameters associated with all swarm intelligent optimization algorithms. In addition, there are some adjustable parameters that should also be properly selected. GA requires to select six adjustable parameters, i.e., the mutation probability, the mutation mechanism, the cross-over probability, the cross-over mechanism, the selection mechanism, and the replacement mechanism. PSO requires to select four adjustable parameters, i.e., the individual learning factor, the social learning factors, the inertial weight, and the initial velocity. However, FOA only requires to select one adjustable parameter, i.e., the initial position of fruit fly population. Moreover, compared with GA and PSO, on the one hand, FOA has the advantages of being easily understood and implemented due to its shorter program code, and on the other hand, FOA can reach the global optimal solution more quickly. Therefore, FOA has gained much attention and successfully been applied to optimizing the parameters of SVM and TSVM model.

Shen et al. developed a novel SVM parameter tuning scheme based on FOA, termed as FOA-SVM [20]. The empirical results demonstrate that the proposed FOA-SVM can obtain much more appropriate model parameters as well as significantly reduce the computational time compared with grid search technique-based SVM (Grid-SVM), GA-based SVM (GA-SVM), and PSO-based SVM (PSO-SVM). Wang et al. proposed a fault diagnosis method using improved pattern spectrum and FOA-SVM [21]. FOA-SVM can help seek optimal parameters of SVR that is employed for pattern recognition. The classification accuracy of the proposed approach shows an acceptable diagnosis effect. Wu and Cao utilized FOA to optimize the parameters of Support Vector Regression (SVR) model [22,23]. They proposed a novel approach, namely FOA-SVR. The results show that the FOA-SVR approach is a viable option for tourist flow forecasting and electricity consumption forecasting applications. Fan et al. designed a novel prediction model for the number of vacant parking spaces after a specific period of time based on SVR with FOA [24]. FOA is utilized to search the optimal parameters of SVR. The experimental results show that the proposed FOA-SVR method has higher accuracy and stability than other state-of-the-art methods in all the prediction scenarios. Li et al. employed FOA to solve the problem of steel casting. The results of testing on actual production process showed that, compared with Artificial Bee Colony (ABC) algorithm and modified GA, the control parameters optimized by FOA can address actual scheduling problem more effectively under static environment [25]. Ding et al. adopted different swarm intelligence algorithms to optimize TSVM. They discovered that the parameters optimized by swarm intelligent algorithms can improve classification precision compared with traditional grid search algorithm. The FOA can obtain better classification results than other swarm intelligent algorithms, such as GA and PSO [26].

Unfortunately, to our knowledge, there is still no relevant report on the parameter selection problem for TSVR. Considering the FOA has the above-mentioned advantages over other swarm intelligent algorithms, we combine FOA with TSVR. FOA is utilized to optimize the penalty parameters C_1, C_2 and the insensitive loss function parameters ε_1, ε_2 of TSVR, simultaneously. We proposed a Twin Support Vector Regression based on Fruit fly Optimization Algorithm, termed as FOA-TSVR. In FOA-TSVR, the locations of fruit flies represent the four parameters to be optimized in TSVR. We can find the fruit fly with minimal smell concentration in the fruit fly swarm within finite iterations. The final locations of the fruit fly with minimal smell concentration correspond to the four optimal parameters.

The rest of this paper is organized as follows. Section 2 briefly presents the twin support vector regression. Section 3 elaborates our FOA-TSVR. Section 4 presents experimental results and analysis. Section 5 gives the conclusions of this paper.

2. **Twin Support Vector Regression.** Given a training sample set $T = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$ (m is the size of the training sample set) such that $x_i \in \mathbf{R}^d$ (d is the dimension of the training sample set) is an input and $y_i \in \mathbf{R}$ is a target output. The training sample is represented by m row vector $\boldsymbol{A}_i$, $i = 1, 2, \ldots, m$. And the training sample $\boldsymbol{A}_i = (\boldsymbol{A}_{i1}, \boldsymbol{A}_{i2}, \ldots, \boldsymbol{A}_{id})$ is in the d-dimensional real space $\mathbf{R}$, where $\boldsymbol{A} = (\boldsymbol{A}_1; \boldsymbol{A}_2; \ldots; \boldsymbol{A}_m)$ and $\boldsymbol{Y} = (y_1; y_2; \ldots; y_m)$. In linear case, the Twin Support Vector Regression (TSVR) determines the primal problems by the ε_1 insensitive lower bound $f_1(x) = \boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x} + b_1$ and the ε_2 insensitive upper bound $f_2(x) = \boldsymbol{w}_2^{\mathrm{T}}\boldsymbol{x} + b_2$ as follows:

$$\begin{aligned} &\min_{\boldsymbol{w}_1, b_1, \boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{e}\varepsilon_1 - (\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}b_1)\|^2 + C_1\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi} \\ &s.t. \quad \boldsymbol{Y} - (\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}b_1) \geq \boldsymbol{e}\varepsilon_1 - \boldsymbol{\xi},\ \boldsymbol{\xi} \geq \boldsymbol{0} \end{aligned} \tag{1}$$

$$\begin{aligned} &\min_{\boldsymbol{w}_2, b_2, \boldsymbol{\eta}} \frac{1}{2}\|\boldsymbol{Y} + \boldsymbol{e}\varepsilon_2 - (\boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}b_2)\|^2 + C_2\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\eta} \\ &s.t. \quad (\boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}b_2) - \boldsymbol{Y} \geq \boldsymbol{e}\varepsilon_2 - \boldsymbol{\eta},\ \boldsymbol{\eta} \geq \boldsymbol{0} \end{aligned} \tag{2}$$

where C_1 and C_2 are penalty parameters, $\varepsilon_1, \varepsilon_2 > 0$ are constants, $\boldsymbol{w}_1, \boldsymbol{w}_2 \subseteq \mathbf{R}^d$, $b_1, b_2 \subseteq \mathbf{R}$. $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are slack vectors, and $\boldsymbol{e}$ is a unit column vector of dimension $d \times 1$.

By introducing the Lagrange multiplier vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, we can obtain the following dual problems of Equations (1) and (2):

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} -\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha} + \boldsymbol{f}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{\alpha} - \boldsymbol{f}^{\mathrm{T}}\boldsymbol{\alpha} \\ &s.t. \quad \boldsymbol{0} \leq \boldsymbol{\alpha} \leq C_1\boldsymbol{e} \end{aligned} \tag{3}$$

$$\max_{\gamma} -\frac{1}{2}\gamma^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\gamma - \boldsymbol{h}^{\mathrm{T}}\boldsymbol{G}\left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}\gamma + \boldsymbol{h}^{\mathrm{T}}\gamma \quad s.t. \quad \boldsymbol{0} \leq \gamma \leq C_2\boldsymbol{e} \tag{4}$$

where $\boldsymbol{G} = [\boldsymbol{A} \quad \boldsymbol{e}]$, $\boldsymbol{f} = \boldsymbol{Y} - \varepsilon_1\boldsymbol{e}$ and $\boldsymbol{h} = \boldsymbol{Y} + \varepsilon_2\boldsymbol{e}$.

Solving Equations (3) and (4), we can obtain the following regression function:

$$\boldsymbol{f}(\boldsymbol{x}) = \frac{1}{2}\left[\boldsymbol{f}_1(\boldsymbol{x}) + \boldsymbol{f}_2(\boldsymbol{x})\right] = \frac{1}{2}(\boldsymbol{w}_1 + \boldsymbol{w}_2)^{\mathrm{T}}\boldsymbol{x} + \frac{1}{2}(b_1 + b_2) \tag{5}$$

where $\left[\boldsymbol{w}_1^{\mathrm{T}} \quad b_1\right]^{\mathrm{T}} = \left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{\alpha})$ and $\left[\boldsymbol{w}_2^{\mathrm{T}} \quad b_2\right]^{\mathrm{T}} = \left(\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}\right)^{-1}\boldsymbol{G}^{\mathrm{T}}(\mathbf{h} + \boldsymbol{\gamma})$.

By introducing kernel function $\boldsymbol{K}(\cdot,\cdot)$, we can easily extend the linear case to the non-linear case.

The non-linear TSVR determines the primal problems by the ε_1 insensitive lower bound $\boldsymbol{f}_1(\boldsymbol{x}) = K\left(\boldsymbol{x}^{\mathrm{T}}, \boldsymbol{A}^{\mathrm{T}}\right)\boldsymbol{w}_1 + b_1$ and the ε_2 insensitive upper bound $\boldsymbol{f}_2(\boldsymbol{x}) = K\left(\boldsymbol{x}^{\mathrm{T}}, \boldsymbol{A}^{\mathrm{T}}\right)\boldsymbol{w}_2 + b_2$, which determines the final regression function.

Let $\boldsymbol{H} = \left[\boldsymbol{K}\left(\boldsymbol{A} \quad \boldsymbol{A}^{\mathrm{T}}\right) \quad \boldsymbol{e}\right]$, the final regression function of non-linear TSVR can be expressed as follows:

$$\boldsymbol{f}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{K}\left(\boldsymbol{x}^{\mathrm{T}}, \boldsymbol{A}^{\mathrm{T}}\right)(\boldsymbol{w}_1 + \boldsymbol{w}_2) + \frac{1}{2}(b_1 + b_2) \tag{6}$$

where $\left[\boldsymbol{w}_1^{\mathrm{T}} \quad b_1\right]^{\mathrm{T}} = \left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{\alpha})$ and $\left[\boldsymbol{w}_2^{\mathrm{T}} \quad b_2\right]^{\mathrm{T}} = \left(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{h} + \boldsymbol{\gamma})$.

3. **FOA-TSVR.** In this section, we first give a brief overview of FOA, and then we present the procedure of FOA-TSVR and discuss its parameter setting.

3.1. **Brief overview of the FOA.** The FOA is first proposed by Pan, and it is a new swarm intelligent algorithm which simulates the food searching behaviours of the fruit fly [19]. The sensing and vision abilities of the fruit fly are far superior to other species. The fruit flies can recognize various smells floating in the air, which is originated from the food as far away as 40 kilometres. After approaching the actual location of the food, the fruit flies can lock the food by their acute visions. FOA has strong global optimization ability, and the smell mechanisms can ensure to avoid trapping into local minimal locations. Moreover, with the aid of vision, the fruit fly population can gradually update current optimal locations. The illustration of the FOA is shown in Figure 1.

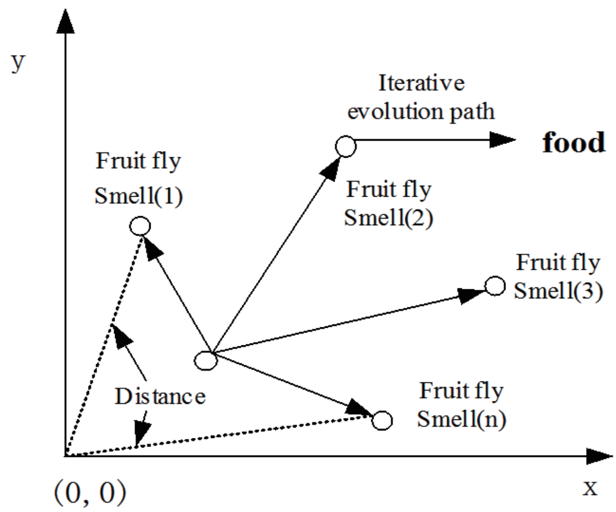

FIGURE 1. The illustration of the FOA

Due to the fact that FOA is much easier to be implemented and has faster convergence speed than other swarm intelligent algorithms, we attempt to optimize the four parameters of TSVR using FOA.

3.2. **The procedure of FOA-TSVR.** The FOA is utilized to tune the four parameters of TSVR dynamically. The idea of the proposed FOA-TSVR is originated from the existing FOA-SVR, and the key difference of our FOA-TSVR from the existing FOA-SVR is that FOA-SVR can only optimize two parameters while FOA-TSVR can optimize four parameters simultaneously.

The detailed steps of FOA-TSVR are presented as follows.

Step 1: Initialize the size of the fruit fly population ($size_{pop}$), the maximum number of generation ($gen_{\max}$), the optimal fitness function value (fit_{opt}), and the optimal coordinates $(U^j, V^j) = (0, 0)$, $j = 1, 2, 3, 4$, where U^j and V^j represent the horizontal and vertical locations of the four fruit flies within a group, respectively. To give a stochastic characteristic to our algorithm, we set the initial coordinates U_0^j and V_0^j of each fruit fly as follows:

$$\begin{cases} U_0^j = rand_j \\ V_0^j = rand_j \end{cases} \tag{7}$$

where $rand_j$ is a random number in the closed interval $[0, 1]$.

Step 2: Set the initial generation of the fruit fly population as $gen = 1$.

Step 3: Set the initial group of fruit fly population as $i = 1$.

Step 4: Calculate the locations of the ith fruit fly group as follows:

$$\begin{cases} U_i^j = U_0^j + random\_value \\ V_i^j = V_0^j + random\_value \end{cases} \tag{8}$$

where $random\_value$ is a random number in the closed interval $[-50, 50]$.

Step 5: Calculate the distance $Dist_i^j$ and the value of taste concentration judgment S_i^j as follows:

$$\begin{cases} Dist_i^j = \sqrt{\left(U_i^j\right)^2 + \left(V_i^j\right)^2} \\ S_i^j = 1/Dist_i^j \end{cases} \tag{9}$$

where $Dist_i^j$ represents the distance between the ith fruit fly group and the origin, and S_i^j is the reciprocal of $Dist_i^j$.

Step 6: Let $C_1 = S_i^1$, $C_2 = S_i^2$, $\varepsilon_1 = S_i^3$, $\varepsilon_2 = S_i^4$. Substitute the four parameters C_1, C_2, ε_1, ε_2 into TSVR to train the model. Then, we can calculate the fitness function value $smell_i$.

Step 7: If $i < size_{pop}$, let $i = i + 1$, go back to Step 4; otherwise, implement Step 8.

Step 8: Find out the fruit fly group with minimum fitness function value from the fruit fly population, i.e., $best\_smell_p = \min(smell_1, \ldots, smell_{pop\_size})$, where p is the group with minimum fitness function value.

Step 9: If $best\_smell_p > fit_{opt}$, update locations $\left(U_0^j, V_0^j\right) = \left(U_p^j, V_p^j\right)$, $j = 1, 2, 3, 4$; otherwise $fit_{opt} = best\_smell_p$, update locations $(U^j, V^j) = \left(U_p^j, V_p^j\right)$, $\left(U_0^j, V_0^j\right) = \left(U_p^j, V_p^j\right)$, $j = 1, 2, 3, 4$.

Step 10: Let $gen = gen + 1$. If $gen < gen_{\max}$, go back to Step 3; otherwise implement Step 11.

Step 11: Obtain the optimal parameters $C_1 = S^1$, $C_2 = S^2$, $\varepsilon_1 = S^3$ and $\varepsilon_2 = S^4$, where $S^j = 1/\sqrt{U^j + V^j}$, $j = 1, 2, 3, 4$.

Step 12: Substitute the optimal parameters into TSVR to test the samples.

In order to better understand the process of the proposed algorithm, we present the flowchart of FOA-TSVR in Figure 2.

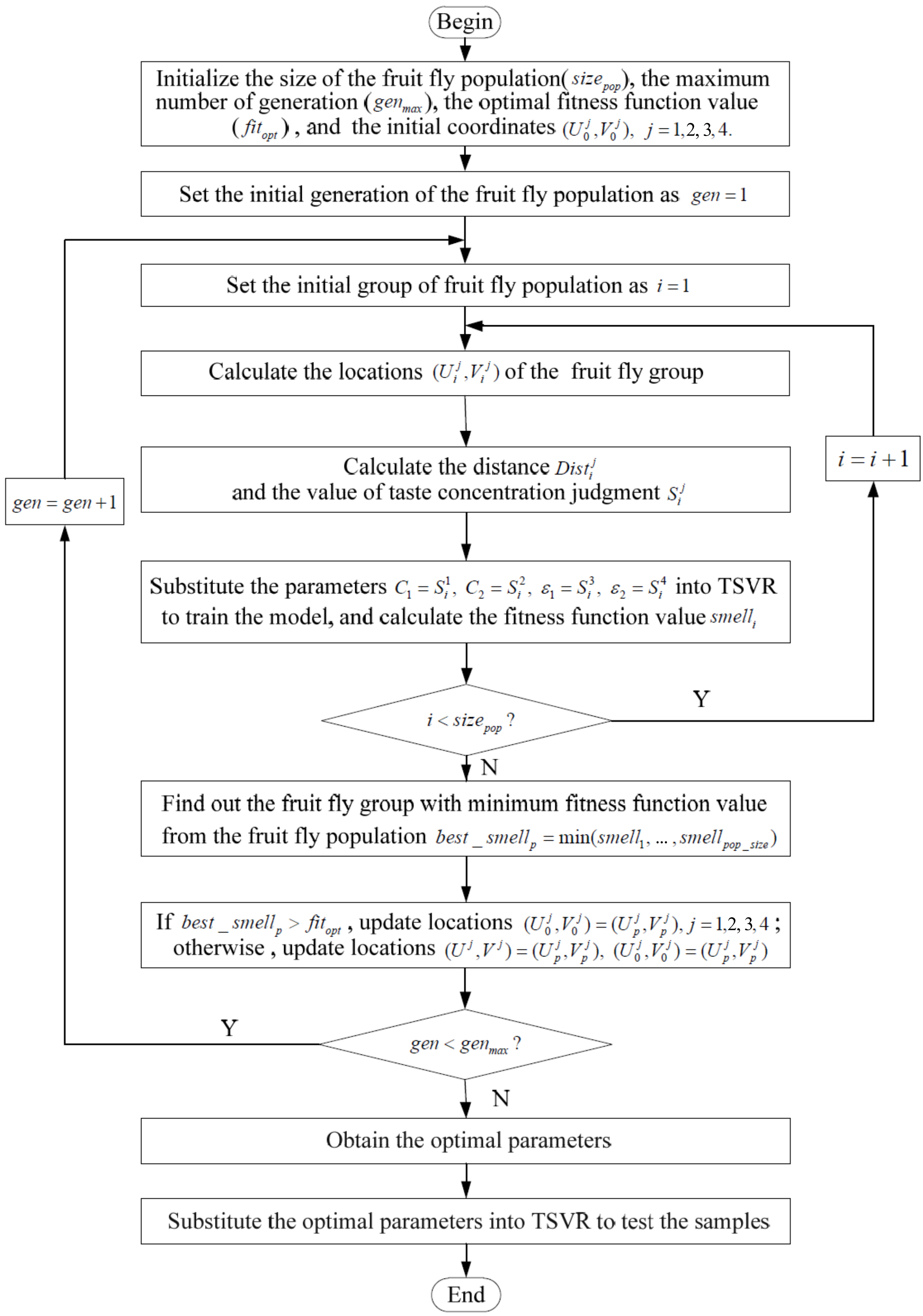

Figure 2. The flowchart of FOA-TSVR

3.3. **The parameter setting of FOA-TSVR.** It is very important to set suitable parameters for FOA-TSVR. In FOA-TSVR, the initial size of the fruit fly population ($size_{pop}$) determines the exploration ability and the computation efficiency of FOA. If the $size_{pop}$ is small, the exploration ability of FOA will weaken. Conversely, if the $size_{pop}$ is large, the computation efficiency of FOA will also decrease. In general, the $size_{pop}$ is selected in the range of 10 to 50 [19,26]. Therefore, in our FOA-TSVR, the $size_{pop}$ is set as 20. In addition, the $gen_{\max}$ is set as 100, the initial positions of fruit fly population are set within the closed interval $[0, 10]$, and the random direction and the distance of fruit flies are set within the closed interval $[-50, 50]$. The C_1 and C_2 are selected from the set $\{2^i|i = -6, \ldots, 0, \ldots, 6\}$, and ε_1 and ε_2 are selected from the set $\{0.01, 0.03, \ldots, 0.20\}$.

4. Experiments.

4.1. **Experimental design.** In order to show the prediction performance of FOA-TSVR, and to distinguish the strengths of FOA-TSVR in terms of optimization time, we conduct a detailed experimental research on benchmark datasets and glutamic acid fed-batch fermentation process, respectively.

To show the prediction performance of FOA-TSVR, the Root Mean Square Error (RMSE) and the Absolute Average Error (AAE) are used as fitness functions. We calculate the RMSE and AAE respectively, in 20 independent trials, using different state-of-the-art algorithms like Grid-SVR (i.e., Support Vector Regression based on Grid search algorithm), Grid-TSVR (i.e., Twin Support Vector Regression based on Grid search algorithm), PSO-TSVR (i.e., Twin Support Vector Regression based on Particle Swarm Optimization algorithm), and CS-TSVR (i.e., Twin Support Vector Regression based on Cuckoo Search algorithm). And the results of RMSE and AAE are averaged. The RMSE and AAE can be calculated by Equations (10) and (11), respectively.

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2} \tag{10}$$

$$\mathrm{AAE} = \frac{1}{m}\sum_{i=1}^{m}|(\hat{y}_i - y_i)| \tag{11}$$

where m is the number of samples, y_i is the actual output of the ith input, and $\hat{y}_i$ is the predicted value of the ith input.

Furthermore, to distinguish the superiority that FOA-TSVR costs less optimization time, we compared our algorithm with Grid-TSVR, PSO-TSVR, and CS-TSVR.

All experiments are implemented in MATLAB 2014a platform. We use LIBSVM and QUADPROG function in the quadratic programming toolbox to solve SVR and TSVR, respectively. The seven benchmark datasets used in our experiments are presented in Table 1, whose sizes vary from 102 to 5375. They can be downloaded from UCI machine learning repository. For each dataset, half samples are randomly selected for training and the remaining samples are selected for testing.

4.2. **Parameter setting.** In Grid-SVR, C and ε are selected from the same set as in FOA-TSVR. In Grid-TSVR, C_1 and C_2 are selected from the same set as in FOA-TSVR. Due to the fact that the setting of ε_1 and ε_2 cannot greatly influence the fitting regression precision [9-16], and to make the comparison fair with Grid-SVR, we set $\varepsilon_1 = 0.2$ and $\varepsilon_2 = 0.2$. In PSO-TSVR, we use default settings, i.e., $c_1 = 0.6$, $c_2 = 0.6$ and $w = 0.6$. The $gen_{\max}$ is set as 100, the initial population size is set as 20, and the search space

TABLE 1. The seven benchmark datasets used in our experiments

Dataset	**Maximum number of training samples**	**Attributions**
Concrete Slump	102	10
Auto-Mpg	398	7
Boston-Housing	506	13
Energy Efficiency	768	8
Airfoil-Noise	1500	6
Abalone	4376	8
Page Blocks	5375	10

TABLE 2. The results of average RMSE and AAE on benchmark datasets in linear case

Datasets	**Algorithms**	**Average RMSE**	**Average AAE**
Concrete Slump	Grid-SVR	4.5041	3.0158
	Grid-TSVR	2.8580	2.2315
	PSO-TSVR	2.7287	2.1853
	CS-TSVR	2.7009	**2.1620**
	FOA-TSVR	**2.6837**	2.1883
Auto-Mpg	Grid-SVR	0.6799	0.4869
	Grid-TSVR	0.6306	0.4697
	PSO-TSVR	0.5953	0.4597
	CS-TSVR	**0.5868**	**0.4459**
	FOA-TSVR	0.5897	0.4548
Boston-Housing	Grid-SVR	5.9186	4.1822
	Grid-TSVR	5.1376	3.6715
	PSO-TSVR	5.0454	3.4480
	CS-TSVR	**4.7511**	**3.3746**
	FOA-TSVR	4.9611	3.4227
Energy Efficiency	Grid-SVR	2.3593	1.5264
	Grid-TSVR	1.9968	1.4744
	PSO-TSVR	2.0419	1.4878
	CS-TSVR	1.9811	**1.4677**
	FOA-TSVR	**1.9772**	1.4708
Airfoil-Noise	Grid-SVR	5.0829	4.1056
	Grid-TSVR	**4.8282**	**3.7678**
	PSO-TSVR	4.8473	3.7970
	CS-TSVR	4.8473	3.8112
	FOA-TSVR	4.8337	3.8043
Abalone	Grid-SVR	**2.1283**	**0.7365**
	Grid-TSVR	2.2967	0.7653
	PSO-TSVR	2.3107	0.7608
	CS-TSVR	2.2640	0.7586
	FOA-TSVR	2.2546	0.7618
Page Blocks	Grid-SVR	0.7672	0.2768
	Grid-TSVR	0.9331	0.3116
	PSO-TSVR	0.8243	0.2756
	CS-TSVR	0.7517	0.2779
	FOA-TSVR	**0.6895**	**0.2678**

dimension is set as 4. The C_1 and C_2, ε_1 and ε_2 are selected from the set as in FOA-TSVR. In CS-TSVR, the nests are set as 20, the probability of being found is set as 0.2, the upper and lower bounds are set within the closed interval $[0, 100]$, and the $gen_{\max}$ is set as 100. The C_1 and C_2, ε_1 and ε_2 are selected from the same set as in FOA-TSVR.

4.3. **Experimental results and analysis.**

4.3.1. *The linear test.* We conduct experiment on benchmark datasets. The average RMSE and AAE in 20 independent trials on seven benchmark datasets are presented in Table 2.

We employ standard 5-fold cross-validation technique. The minimum average RMSE and AAE on each dataset are marked with bold font. From Table 2, we can find that FOA-TSVR achieves one best prediction pair on Page Blocks datasets. However, the prediction performances of FOA-TSVR, PSO-TSVR, and CS-TSVR on the remaining six datasets are very close. This means that FOA-TSVR can be used to find suitable parameters for TSVR.

Figure 3 presents the average optimization time of Grid-SVR, Grid-TSVR, PSO-TSVR, CS-TSVR, and FOA-TSVR on seven benchmark datasets. From this figure, we can find that FOA-TSVR costs the shortest optimization time on five benchmark datasets. Unfortunately, on Airfoil-Noise and Abalone datasets, the average optimization time cost by FOA-TSVR is slightly longer than CS-TSVR. This might be explained by the fact that TSVR has reached saturation for Airfoil-Noise and Abalone datasets. Therefore, FOA-TSVR can finish parameter optimization within the shortest time especially for large-scale datasets.

4.3.2. *The non-linear test.* We use radial basis function, i.e., $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-(\boldsymbol{x}_i, \boldsymbol{x}_j)^2/\sigma)$ in nonlinear case, where the kernel width parameter is set as $\sigma = 2$. The average RMSE and AAE in 20 independent trials on seven benchmark datasets are presented in Table 3.

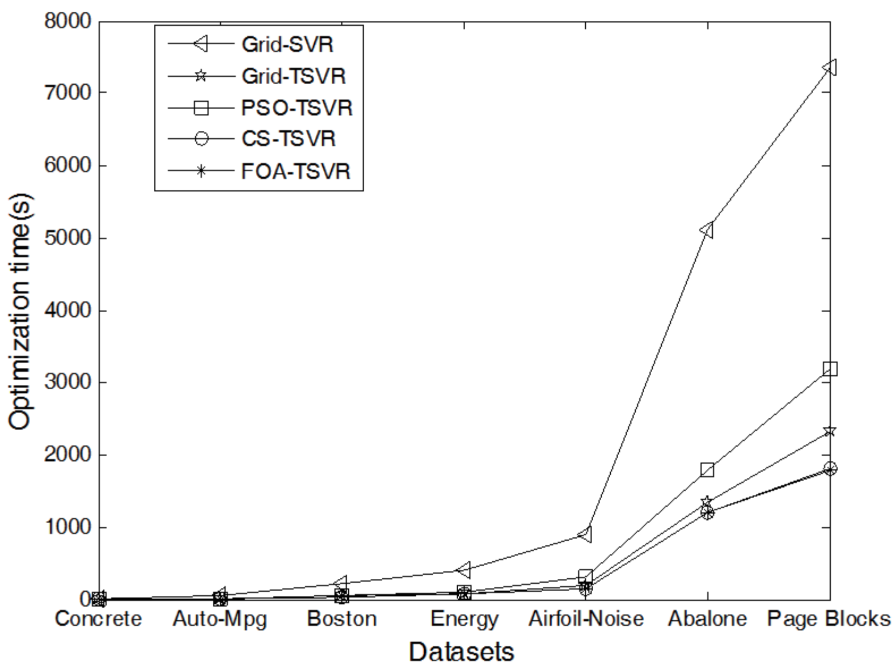

Figure 3. Comparison of average optimization time for different algorithms on benchmark datasets in linear case

TABLE 3. The results of average RMSE and AAE on benchmark datasets in nonlinear case

Datasets	**Algorithms**	**Average RMSE**	**Average AAE**
Concrete Slump	Grid-SVR	2.9837	3.5677
	Grid-TSVR	1.3697	1.0123
	PSO-TSVR	1.3040	1.0008
	CS-TSVR	1.3483	0.9002
	FOA-TSVR	**1.2919**	**0.8996**
Auto-Mpg	Grid-SVR	0.6574	0.6861
	Grid-TSVR	0.5826	0.4471
	PSO-TSVR	0.5868	0.4401
	CS-TSVR	0.5614	0.4220
	FOA-TSVR	**0.5614**	**0.4160**
Boston-Housing	Grid-SVR	4.9324	3.3155
	Grid-TSVR	4.9611	3.4961
	PSO-TSVR	5.0788	3.2604
	CS-TSVR	4.7670	3.2918
	FOA-TSVR	**4.6111**	**3.1408**
Energy Efficiency	Grid-SVR	2.0693	1.4338
	Grid-TSVR	2.0360	1.5002
	PSO-TSVR	2.0027	1.4548
	CS-TSVR	1.9537	1.4400
	FOA-TSVR	**1.9380**	**1.4100**
Airfoil-Noise	Grid-SVR	5.9729	4.9558
	Grid-TSVR	6.2796	5.4257
	PSO-TSVR	5.8908	5.2898
	CS-TSVR	**5.5019**	**4.8976**
	FOA-TSVR	5.9948	5.2953
Abalone	Grid-SVR	**1.9459**	**0.6036**
	Grid-TSVR	2.3201	0.7713
	PSO-TSVR	2.1330	0.7489
	CS-TSVR	2.0956	0.7539
	FOA-TSVR	2.1611	0.7802
Page Blocks	Grid-SVR	0.8191	0.2667
	Grid-TSVR	0.8606	0.2812
	PSO-TSVR	0.7672	0.2589
	CS-TSVR	0.7880	0.2588
	FOA-TSVR	**0.6325**	**0.2368**

The minimum average RMSE and AAE on each dataset are also marked with bold font. Figure 4 presents the average optimization time of Grid-SVR, Grid-TSVR, PSO-TSVR, CS-TSVR, and FOA-TSVR on seven benchmark datasets.

As can be seen from Table 3, FOA-TSVR achieves five best prediction pairs on seven benchmark datasets. The prediction performances of FOA-TSVR and CS-TSVR on the Airfoil-Noise and Page Blocks datasets are very close. Moreover, from Tables 2 and 3, we can find that the fitting regression precision in the nonlinear case is smaller than the linear case. This further implies that our FOA-TSVR can be used to find most suitable parameters of TSVR. From Figure 4, we can see that our FOA-TSVR almost costs the shortest optimization time compared with other state-of-the-art algorithms.

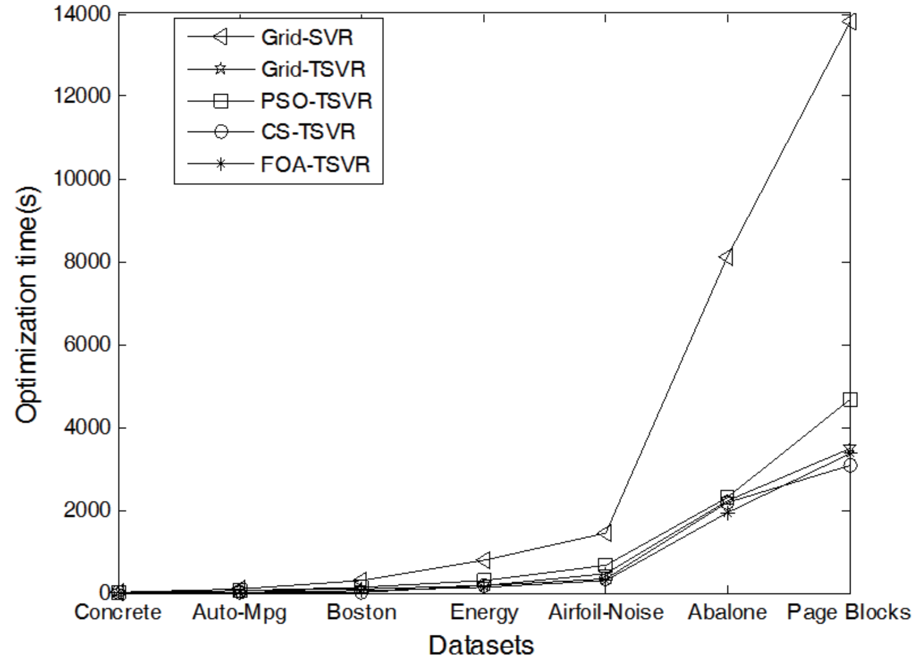

FIGURE 4. Comparison of average optimization time for different algorithms on benchmark datasets in nonlinear case

4.3.3. *Experiment on glutamic acid fed-batch fermentation process.* To further highlight the performance of our FOA-TSVR, we conduct experiment on actual glutamic acid fed-batch fermentation process. The related data are offered by the Key Laboratory of Industrial Biotechnology, Ministry of Education, Jiangnan University [27]. It is well-known that the glutamic fed-batch fermentation process is a rather complicated nonlinear process. There are three different kinds of variable in glutamic acid fed-batch fermentation process: physical variable, chemical variable and biological variable. To maximize enterprise profits, the managers always care more about biological variable, such as biomass concentration, glutamic acid concentration and substrate concentration. However, these three variables cannot be measured online. Fortunately, with the help of soft sensor modelling, such as FOA-TSVR, they can be easily measured and tuned online.

In this paper, we focus on glutamic acid concentration. Six batch experiments are carried out under the condition of keeping 10%, 20%, 30% and 50% Dissolved Oxygen (DO) concentration, respectively. Each batch of data can represent the whole fermentation process. The data include offline analyzed data and online measured data, and they are normalized to eliminate the influence of dimension. Five batches are used to train the model and the remaining one batch is used to test the model. We use the same parameter setting as in the nonlinear test.

The iterative RMSE trend of glutamic acid concentration using FOA-TSVR is shown in Figure 5. Figure 6 presents the RMSE of glutamic acid concentration in 20 independent trials. We can see from Figure 5 that the RMSE of glutamic acid concentration using our FOA-TSVR will approximate 0 within finite iterations. This implies that FOA-TSVR can quickly find the four optimal parameters of TSVR. Figure 6 further proves that our FOA-TSVR is stable.

Table 4 lists the results of average RMSE, AAE of glutamic acid concentration and optimization time in glutamic acid fed-batch fermentation process using different algorithms. It can be seen that the average RMSE and AAE of glutamic acid concentration

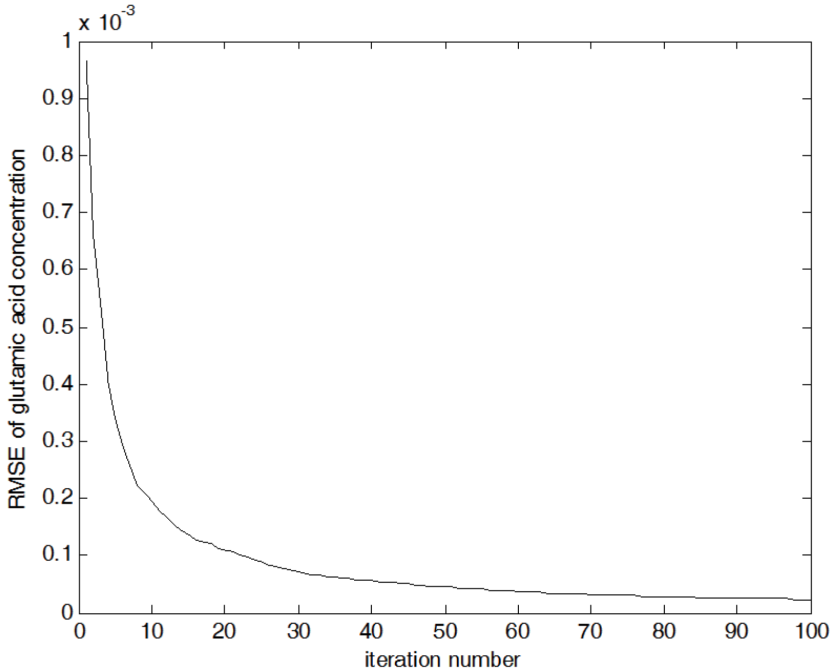

FIGURE 5. The iterative RMSE trend of glutamic acid concentration using FOA-TSVR model

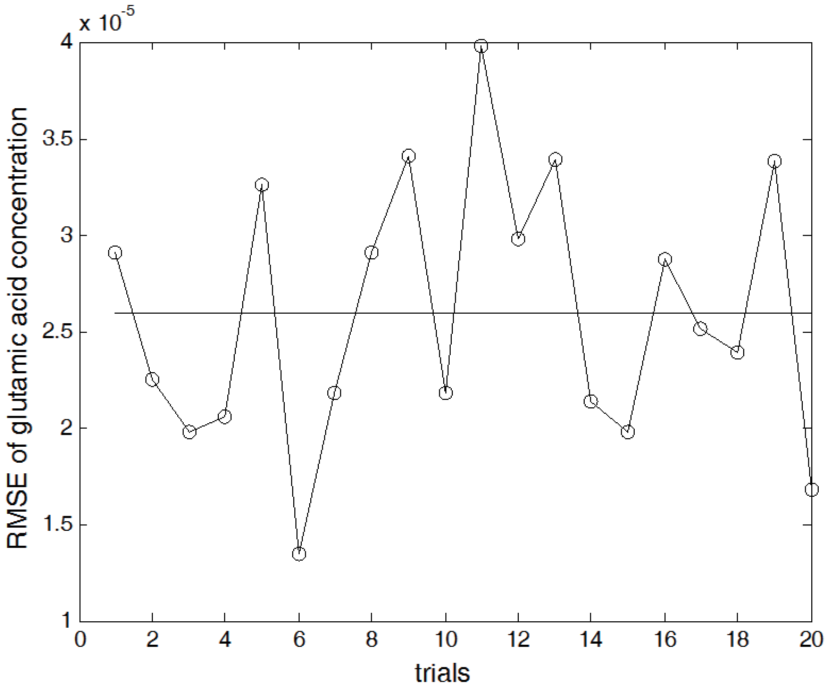

FIGURE 6. The RMSE of glutamic acid concentration in 20 independent trials

using FOA-TSVR are the smallest compared with other swarm intelligent optimization algorithms. The optimization time using FOA-TSVR is almost half that of Grid-TSVR. Moreover, the optimization time using FOA-TSVR is shorter than PSO-TSVR but slightly longer than CS-TSVR. This means that PSO-TSVR, CS-TSVR and FOA-TSVR can find the four optimal parameters of TSVR rapidly. In a word, we can conclude that our

Table 4. The results of average RMSE, AAE of glutamic acid concentration and optimization time in glutamic acid fed-batch fermentation process using different algorithms

Algorithms	**Average RMSE**	**Average AAE**	**Optimization time (s)**
Grid-TSVR	1.03×10^{-2}	7.20×10^{-3}	205.87
PSO-TSVR	5.03×10^{-3}	2.88×10^{-3}	142.86
CS-TSVR	3.64×10^{-4}	5.09×10^{-4}	**108.59**
FOA-TSVR	$\mathbf{2.64 \times 10^{-4}}$	$\mathbf{2.31 \times 10^{-4}}$	121.25

FOA-TSVR can ensure to find most suitable parameters for TSVR within the shortest optimization time.

5. **Conclusions.** This paper investigates the parameter selection problem of twin support vector regression. The main contribution of this paper is that we first attempt to use a new swarm intelligent algorithm, named fruit fly optimization algorithm, to optimize the penalty parameters and the insensitive loss function parameters of twin support vector regression simultaneously. The results on benchmark datasets and glutamic acid fed-batch fermentation process validate that our FOA-TSVR not only can greatly shorten the optimization time but also can be used to find most suitable parameters for TSVR. Therefore, the proposed FOA-TSVR can be used in scenarios which requires fast parameter selection.

However, it should be pointed out that the fruit fly optimization algorithm may trap into local minimum like other swarm intelligent algorithms such as GA, PSO and CS. Therefore, improving FOA algorithm by introducing other approaches with better global searching capabilities may be a solution we need to focus. Furthermore, combining the advantages of different swam intelligent algorithms to form new optimization algorithm is also our future research directions. We hope these questions will be successfully addressed in the near future.

Acknowledgment. This research was funded by the National Natural Science Foundation of China under Grant Numbers 21878124, 31771680 and 61773182.

REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, Inc., New York, 1998.
[2] J. Ruan, Y. Shi and J. Yang, Forest fires burned area prediction based on support vector machines with feature selection, *ICIC Express Letters*, vol.5, no.8(A), pp.2597-2603, 2011.
[3] J. Ruan, X. Wang and Y. Shi, Developing fast predictors for large-scale time series using fuzzy granular support vector machines, *Applied Soft Computing*, vol.13, no.9, pp.3981-4000, 2013.
[4] C. Cortes and V. Vapnik, Support vector networks, *Machine Learning*, vol.20, no.3, pp.273-297, 1995.
[5] N. Y. Deng and Y. J. Tian, *Support Vector Machine: Theoretical Algorithm and Extension*, Science Press, Beijing, 2009.
[6] O. L. Mangasarian and E. W. Wild, Multisurface proximal support vector classification via generalized eigenvalues, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.28, no.1, pp.69-74, 2006.
[7] Jayadeva, R. Khemchandani and S. Chandra, Twin support vector machines for pattern classification, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.29, no.5, pp.905-910, 2007.
[8] S. F. Ding, *Twin Support Vector Machine: Theory Algorithm and Extension*, Science Press, Beijing, 2017.
[9] S. F. Ding, F. L. Wu and Z. Z. Shi, Wavelet twin support vector machines, *Neural Computing and Applications*, vol.25, pp.1241-1247, 2014.

[10] J. Z. Yu, S. F. Ding and F. Jian, Twin support vector machines based on rough sets, *International Journal of Digital Content Technology & Its Applications*, vol.6, no.20, pp.493-500, 2012.

[11] Z. Q. Qi, Y. J. Tian and Y. Shi, Structural twin support vector machine for classification, *Knowledge-Based Systems*, vol.43, pp.74-81, 2013.

[12] X. J. Peng, TSVR: An efficient twin support vector machine for regression, *Neural Networks*, vol.23, no.3, pp.365-372, 2010.

[13] Y. H. Shao, C. H. Zhang, Z. M. Yang et al., An ε-twin support vector machine for regression, *Neural Computing & Applications*, vol.23, no.1, pp.175-185, 2013.

[14] Y. T. Xu and L. S. Wang, A weighted twin support vector regression, *Knowledge-Based Systems*, vol.33, pp.92-101, 2012.

[15] X. B. Chen, J. Yang, J. Liang et al., Smooth twin support vector regression, *Neural Computing and Applications*, vol.21, no.3, pp.505-513, 2012.

[16] X. B. Chen, J. Yang and L. Chen, An improved robust and sparse twin support vector regression via linear programming, *Applied Soft Computing*, vol.18, no.12, pp.2335-2348, 2014.

[17] K. Deb, A. Pratap, S. Agarwal et al., A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evolutionary Computation*, vol.6, no.2, pp.182-197, 2002.

[18] B. Alatas and E. Akin, Multi-objective rule mining using a chaotic particle swarm optimization algorithm, *Knowledge-Based Systems*, vol.22, pp.455-460, 2009.

[19] W. T. Pan, A new fruit fly optimization algorithm: Taking the financial distress model as an example, *Knowledge-Based Systems*, vol.26, pp.69-74, 2012.

[20] L. M. Shen, H. L. Chen, Y. Zhe et al., Evolving support vector machines using fruit fly optimization for medical data classification, *Knowledge-Based Systems*, vol.96, pp.61-75, 2016.

[21] B. Wang, H. Xiong, W. Wang et al., Fault diagnosis using improved pattern spectrum and fruit fly optimization algorithm-support vector machine, *Advances in Mechanical Engineering*, vol.10, no.11, pp.1-10, 2018.

[22] L. J. Wu and G. H. Cao, Seasonal SVR with FOA algorithm for single-step and multi-step ahead forecasting in monthly inbound tourist flow, *Knowledge-Based Systems*, vol.110, pp.157-166, 2016.

[23] G. H. Cao and L. J. Wu, Support vector regression with fruit fly optimization algorithm for seasonal electricity consumption forecasting, *Energy*, vol.115, pp.734-745, 2016.

[24] J. K. Fan, Q. Hu and Z. Z. Tang, Predicting vacant parking space availability: An SVR method with fruit fly optimization, *IET Intelligent Transport Systems*, vol.12, no.10, pp.1414-1420, 2018.

[25] J. Q. Li, Q. K. Pan, K. Mao et al., Solving the steelmaking casting problem using an effective fruit fly optimization algorithm, *Knowledge-Based Systems*, vol.72, pp.28-36, 2014.

[26] S. F. Ding, X. K. Zhang and J. Z. Yu, Twin support vector machines based on fruit fly optimization algorithm, *International Journal of Machines Learning and Cybernetics*, vol.7, no.2, pp.193-203, 2016.

[27] B. J. Gu and F. Pan, A soft sensor modelling of biomass concentration during fermentation using accurate incremental online ν-support vector regression learning algorithm, *American Journal of Biochemistry and Biotechnology*, vol.11, no.3, pp.149-159, 2015.