Sensors and Materials, Vol. 35, No. 10 (2023) 4585–4596
MYU Tokyo

S & M 3416

An Enhanced Quantum Genetic Algorithm and Its Application in the Health Monitoring of a Rocket Engine

Hao Xiang*

Guangxi Colleges and Universities Key Laboratory of Complex System Optimization and Big Data Processing, Yulin Normal University, Yulin 537000, China

(Received June 5, 2023; accepted October 3, 2023)

Keywords: quantum genetic algorithm, simulated annealing algorithm, LSSVR, fault prediction, liquid-fuel rocket engine

In this study, the characteristics of different intelligent algorithms, from the simulated annealing algorithm (SA) and genetic algorithm to the quantum genetic algorithm, are analyzed. By utilizing the variety of the population and the rapidity of the convergence of the real double-chain coding objective gradient quantum genetic algorithm, this algorithm is fused with SA; the resulting real double-chain coding objective gradient quantum genetic simulated annealing algorithm is proposed. As the performance of the least squares support vector regression (LSSVR) is very sensitive to its key parameters, the proposed algorithm is applied to optimize these parameters to improve the generalization ability of LSSVR. Following that, a new hybrid nonparametric regression prediction model is put forward. The health monitoring of a liquid-fuel rocket engine is a very important topic, and all types of sensors are used to monitor the factors that can affect the engine thrust from different aspects. The relationship between the thrust and these factors is nonlinear and difficult to express using some formulas. The proposed model is used in the fault prediction of liquid-fuel rocket engine thrust, and the simulation results show that the average relative error is determined to be 0.37% using LSSVR and 0.2977% using the proposed model. Thus, this model is applicable to small samples, nonlinearity, and high dimensions of failure prediction, and is worth promoting to a certain extent.

1. Introduction

At present, the artificial neural network is widely applied in engineering practices, and most of the neural network models use the back propagation (BP) network and its variations.[1–7] The notable feature of the artificial neural network is that it achieves highly complex nonlinear mapping through its own learning, so it has been widely applied in various fields such as function approximation and pattern recognition.[8–10] Although a multilayer neural network can be applied in the fields of linear and nonlinear systems for approximating any function, an artificial neural network does not always become solvable because the optimization process is closely related to the selected initial points. On the other hand, the actual research objects are

*Corresponding author: e-mail: 15971536347@163.com
https://doi.org/10.18494/SAM4523

ISSN 0914-4935
https://myukk.org/

often dynamically changing, and it is difficult to grasp the characteristics of these samples; subject to the limited samples, the effect of artificial neural networks in practical applications is affected. The support vector machine (SVM), which was proposed by Vapnik as a new machine learning algorithm,[11,12] was developed to provide a new approach to solve these problems. This algorithm is based on the rigorous statistical learning theory, and the actual risk is obtained by using the structural risk minimization criterion. SVM can effectively improve its generalization ability, solve the problems of small sample, nonlinearity, and high dimension in the previous learning methods, and overcome shortcomings of neural network learning methods such as the difficulty in determining the network structure, slow convergence, ease of falling into local minima, overlearning and underlearning, and the need for a large number of data samples during training. Thus, SVM has widespread applications.[13–19] However, the performance of SVM is very sensitive to its key parameters, so different intelligent algorithms were used[20–23] to optimize these parameters.

The genetic algorithm (GA) has a strong search capability as well as good global optimization performance, so combining GA with BP can yield the global optimum and improve the accuracy of the result.[24,25] However, GA emphasizes the evolutionary relationship between two generations, and its mutation could lose the best solution. In SA, the state between two temperatures is irrelevant. In theory, any Markov chain of temperatures gradually reaches the stationary distribution, that is, the process from one state to another with increasing number of iterations is independent of the starting state, and the probability of each state in the Markov chain finally obeys a stationary distribution. Combining the characteristics of GA and SA, Xing and Xie presented the genetic simulated annealing (GSA) algorithm.[26]

With the acceleration of the quantum algorithm, a quantum evolutionary algorithm integrating the quantum algorithm with the genetic algorithm was proposed. Narayanan and Moore presented the quantum genetic algorithm (QGA) in 1996 and successfully solved the TSP problem by using it.[27] Han and Kim combined the concepts of the quantum bit and quantum gate in the evolutionary algorithm in 2000 and proposed the genetic quantum algorithm.[28] Because of using the quantum superposition state in QGA, the variety and convergence of the population in QGA are better than those in the common genetic algorithm. However, most of the existing quantum genetic algorithms adopt the binary coding method based on quantum bit measurement.[27–29] This method, which is used in numerical optimization problems, requires frequent coding and decoding, thereby increasing the computation amount. Therefore, the real double-chain coding objective gradient quantum genetic algorithm (DCQGA) was proposed by Li and Li.[30] In the quantum physical universe, the coherent superposition of quantum states in two-particle systems is usually known as entanglement. When a certain performance of a quantum system is measured, the measurement will cause the coherence of the quantum system to be destroyed, so that it is converted from the superposition state to a basic state. This process is called the collapse of the quantum state. After the collapse of the astrophysical universe, it underwent a long and slow process of evolution. In this study, the real double-chain coding objective gradient quantum genetic simulated annealing algorithm (DCQGSAA) based on DCQGA is proposed. Then, we used it to optimize the parameters of LSSVR and proposed a mixture model based on DCQGSAA and LSSVR.

The establishment of an accurate and reliable fault prediction model for liquid-fuel rocket engine thrust is of great significance for the health monitoring and fault diagnosis of liquid-fuel rocket engines.[31] All types of sensors are used to monitor the factors that can affect the engine thrust from different aspects. The relationship between the thrust and these factors is nonlinear and difficult to express using some formulas. Consequently, the above presented model is adopted for the failure prediction of liquid-fuel rocket engine thrust. The prediction results show that this model works well.

2. DCQGA

DCQGA is presented in Ref. 30. Xiang and Wang[32] used this algorithm to optimize the weights and thresholds of the BP neural network that was applied in the field of intelligent fault diagnosis.

2.1 Describing the problem of continuous optimization

The common continuous optimization problem is expressed as

$$\begin{cases} \min f(x) = f(x_1, \cdots, x_n) \\ \text{s.t. } a_i \le x_i \le b_i;\ i = 1, 2, \cdots, n, \end{cases} \tag{1}$$

where $x \in \mathbb{R}^n$. To evaluate the quality of the approximate solution, the fitness function is specified as

$$fit(x) = c_{max} - f(x), \tag{2}$$

where c_{max} is a proper constant.

2.2 Double-chain coding plan

In DCQGA, the probability amplitude of the quantum bit is encoded. The double-chain coding plan is defined as

$$p_i = \begin{bmatrix} \cos(t_{i1}) & \cos(t_{i2}) & \cdots & \cos(t_{in}) \\ \sin(t_{i1}) & \sin(t_{i2}) & \cdots & \sin(t_{in}) \end{bmatrix}, \tag{3}$$

where $t_{ij} = 2\pi \times \boldsymbol{rand}$, with ***rand*** being a random number from 0 to 1, $i = 1, 2, ..., m$, $j = 1, 2, ..., n$, m the population size, and n the number of quantum bits. In DCQGA, the probability amplitude of each quantum bit is considered as two balanced pieces of genes, and each chromosome contains two balanced gene chains representing two different optimization solutions.

2.3 Solution space transformation

Each chromosome in the group contains the probability amplitudes with $2n$ quantum bits, and these probability amplitudes are mapped from the n-dimensional unit space of $I^n = [-1, 1]^n$ to the solution space of the optimization problem by linear transformation. Assuming that the i-th quantum bit of chromosome p_j is $[\alpha_i^j, \beta_i^j]^T$, the corresponding solution space is

$$\begin{aligned} X_{ic}{}^j &= 0.5 \cdot \left[b_i \left(1 + \alpha_i{}^j\right) + \left(a_i (1 - \alpha_i{}^j)\right)\right], \\ X_{is}{}^j &= 0.5 \cdot \left[b_i \left(1 + \beta_i{}^j\right) + a_i \left(1 - \beta_i{}^j\right)\right]. \end{aligned} \tag{4}$$

In the above formula, the probability amplitude α_i^j of the quantum state |0> is related to X_{ic}^j; the probability amplitude β_i^j of the quantum state |1> is related to X_{is}^j; $i = 1, 2, ..., n$, $j = 1, 2, ..., m$.

2.4 Angle direction for quantum rotation gate

In DCQGA, the quantum rotation gate, which is used to update the phase of the quantum bit, is defined as

$$U(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix}. \tag{5}$$

The updating way is described as

$$\begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \cos(t) \\ \sin(t) \end{bmatrix} = \begin{bmatrix} \cos(t + \Delta\theta) \\ \sin(t + \Delta\theta) \end{bmatrix}. \tag{6}$$

To define the direction of the rotation angle $\Delta\theta$, the principle is depicted as

$$A = \begin{vmatrix} \alpha_0 & \alpha_1 \\ \beta_0 & \beta_1 \end{vmatrix}. \tag{7}$$

If $A \neq 0$, the direction is set as $-\mathrm{sgn}(A)$. If $A = 0$, the direction can be positive or negative.

2.5 Size of rotation angle in quantum rotation gate

The step size function of the rotation angle, using the gradient's definition, is defined as

$$\Delta\theta_{ij} = -\mathrm{sgn}(A) \cdot \Delta\theta_0 \cdot \exp\left(-\frac{\left|\nabla f\left(x_i{}^j\right)\right| - \nabla f_{jmin}}{\nabla f_{jmax} - \nabla f_{jmin}} \right), \tag{8}$$

where $\Delta\theta_0$ is the original number of iterations, $\nabla f\left(x_i{}^j\right)$ is the gradient with the evaluation function $f(x)$ at point $x_i{}^j$, and ∇f_{jmax} and ∇f_{jmin} are defined as

$$\nabla f_{jmax} = \max\left\{\left|\frac{\partial f(x_1)}{\partial x_1{}^j}\right|,\cdots,\left|\frac{\partial f(x_m)}{\partial x_m{}^j}\right|\right\},$$
$$\nabla f_{jmin} = \min\left\{\left|\frac{\partial f(x_1)}{\partial x_1{}^j}\right|,\cdots,\left|\frac{\partial f(x_m)}{\partial x_m{}^j}\right|\right\}. \quad (9)$$

Here, $x_i{}^j$ (i = 1, 2, ..., m; j = 1, 2, ..., n) indicates the j-th component of the vector, m is the population size, and n is the dimension number of the space.

2.6 Mutation procedure

While chromosomes are mutated by applying a quantum non-gate and randomly choosing a chromosome based on their mutation probability, quantum non-gates are changed at a number of quantum bits, which are selected randomly; then, these two probability amplitudes of the quantum bit are exchanged, causing the two chains of chromosomes to mutate.

3. DCQGSAA

3.1 Algorithm theory

SA can take on the ability of a sudden jump with a certain probability to effectively avoid the local minimum solutions in the search process. This algorithm receives a good solution as well as a worse solution when using a particular probability in the annealing process. This probability is affected by the temperature parameter, and it will decrease along with decreasing temperature. Therefore, SA is combined with DCQGA to complement their advantages. In DCQGSAA, during each loop, SA is implemented after DCQGA has been implemented.

3.2 Algorithm process

Step 1: Initialize the population. The original population comprises m chromosomes, which are generated by Eq. (3). Determine the original step size of the rotation angle with θ_0 and the mutation probability p_m.
Step 2: Determine the initial temperature t.
Step 3: Using the following formula, determine the fitness value of each chromosome p_i at the current temperature.

$$TF\left(p_i\right)=\frac{e^{\left(f\left(p_0\right)-f\left(p_i\right)\right)/t}}{\sum_{i=1}^{N} e^{-\left(f\left(p_0\right)-f\left(p_i\right)\right)/t}} \tag{10}$$

Here, p_0 is the chromosome corresponding to the optimal solution X_0 up to now, $f(\cdot)$ is the objective function being optimized, and N is the number of chromosomes.

Step 4: Use the roulette strategy to ascertain some chromosome substitute p_0' relating to the so far global optimal solution from all p_i.

Step 5: Transform the solution space. The approximate solution expressed by each chromosome is transformed from the unit space to the solution space described by Eq. (1), and the fitness of each chromosome is estimated using Eq. (2). Set the current optimal solution as $\overline{X}_0$ and its related chromosome as $\overline{p}_0$, and set the optimal solution so far as X_0 and its corresponding chromosome as p_0'.
When $fit(\overline{X}_0) > fit(X_0)$, $p_0' = \overline{p}_0$.

Step 6: Determine the rotation angle size of each quantum bit for each chromosome in the population space using Eq. (8). Their quantum bits are changed by the quantum rotation gate.

Step 7: Mutate each chromosome using the quantum non-gate in accordance with the mutation probability.

Step 8: Implement the annealing temperature operation.

Step 9: Return to Step 2. The program is not terminated until the convergence condition or the maximum number of iterations is satisfied.

The initial temperature and annealing method affect the accuracy of the algorithm to a certain extent, and they are given as $t_{k+1} = \lambda t_k$ and $t_0 = f(p_g)/\ln 5$, where p_g is the chromosome corresponding to the optimal solution in the initial population and λ is the annealing constant.

4. Mixture Model

The performance of LSSVR is associated with the type of kernel function, the corresponding kernel parameters, and the penalty coefficient C, among which the type of kernel function has little effect on LSSVR, but the latter two items have a considerable effect on the performance of LSSVR. The commonly used kernel functions are, for example, the polynomial function and radial basis function (RBF). Among them, the polynomial function has two adjustable parameters, but RBF has only one. Since choosing the proper kernel function helps to reduce the computational duty, RBF is commonly used. In addition, the polynomial kernel function only maps from the original space to a finite high-dimensional feature space, but RBF can map from the original space to an infinite high-dimensional feature space. Thus, we selected RBF to establish the prediction model.

RBF is expressed as

$$K(x, x_i) = \exp\left[-\frac{\|x - x_i\|^2}{2\sigma^2} \right], \tag{11}$$

where σ is the width of RBF. There is no uniform method for properly selecting C and σ, but the commonly used method is cross-validation. It can be seen from Refs. 33 and 34 that the cross-validation method cannot result in a more accurate performance of LSSVR than those using intelligent algorithms; thus, in this study, we used the intelligent algorithm DCQGA. The flowsheet of the proposed model is shown in Fig. 1.

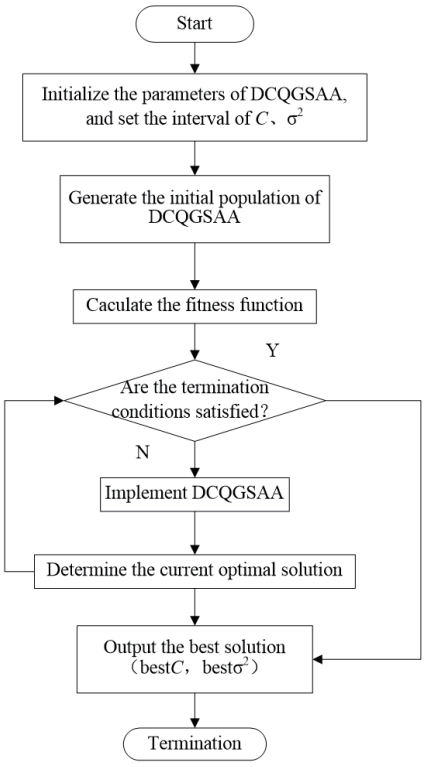

Fig. 1. Flowsheet of proposed model.

5. Application

5.1 Setup of experiment plan

The thrust of a liquid-fuel rocket engine is an important indicator of its health, and by predicting its changes and comparing the predicted value with its threshold, we can predict whether the engine will fail at some time. Under the given working conditions of the engine system, the thrust F is closely related to parameters such as the oxidant flow $\dot{m}_o$, the combustion flow $\dot{m}_f$, the pressure of the combustion chamber p_c, and the sampling time t. There exist high complexity and nonlinearity between each parameter and thrust, and this relationship can be summarized as the function $F = f(\dot{m}_o, \dot{m}_f, p_c, t,...)$. Therefore, F can be predicted using these parameters. Thus, it can be seen that the characteristics of the liquid-fuel rocket engine fault are the high nonlinearity between the causes and the result, the high dimension of experiment samples, and the small size of samples. However, the use of LSSVR alleviates such problems. Therefore, this mixture model can be used to establish the failure prediction model of the liquid-fuel rocket engine.

The training and testing samples from the trial running data shown in Table 1 are scaled.[31] From Table 1, the total number of samples is 25 and the sampling time is 0.1 s. We used the first

Table 1
Trial running data.

Sampling time $t(s)$	Combustion flow ($\dot{m}_f$)	Oxidant flow ($\dot{m}_o$)	Pressure of combustion chamber (p_c)	Thrust (F)
0.0	0.3179	0.4011	0.0098	0.0000
0.1	0.3931	0.4661	0.0559	0.0496
0.2	0.5780	0.5339	0.2238	0.1418
0.3	0.6532	0.5935	0.4643	0.3262
0.4	0.6012	0.5658	0.6170	0.4113
0.5	0.6012	0.5572	0.7049	0.4681
0.6	0.6763	0.6374	0.7832	0.5674
0.7	0.7746	0.6721	0.8447	0.6596
0.8	0.8555	0.8238	0.8951	0.7730
0.9	0.9017	0.8753	0.9301	0.8440
1.0	0.9306	0.9149	0.9580	0.8865
1.1	0.9526	0.9420	0.9790	0.9163
1.2	0.9665	0.9593	0.9874	0.9362
1.3	0.9769	0.9702	0.9930	0.9504
1.4	0.9827	0.9810	0.9944	0.9574
1.5	0.9884	0.9864	0.9951	0.9716
1.6	0.9442	0.9919	0.9972	0.9787
1.7	0.9965	0.9957	0.9986	0.9858
1.8	0.9977	0.9973	1.0000	0.9929
1.9	0.9988	0.9995	1.0000	0.9986
2.0	1.0000	1.0000	1.0000	1.0000
2.1	1.0000	1.0000	1.0000	1.0000
2.2	1.0000	1.0000	1.0000	1.0000
2.3	1.0000	1.0000	1.0000	1.0000
2.4	1.0000	1.0000	1.0000	1.0000

13 rows of the table as the training samples and the remaining 12 rows as the testing samples to determine the effect of prediction and validate the generalization capability of the model. The independent variables of the function $F = f(\dot{m}_o, \dot{m}_f, p_c, t,...)$ were used as inputs in the model and F as the output of the model. Thus, this model can use multiple parameters to predict the thrust.

5.2 Simulation experiment

In this experiment, the parameters of DCQGSAA are given as follows: the population size of 20, the number of iterations of 100, the mutation probability of 0.05, the step size of rotation angle of 0.001*pi, the initial temperature determined by $t_0 = f(p_g)$ /ln5, and the annealing constant of 0.7. Since the performance of LSSVR is extremely sensitive to the parameters of σ^2 and C, their ranges are related to the data distribution of the research object. According to Refs. 33–39, the range of σ^2 values is determined to be between 0.01 and 1000, and the range of C values is determined to be between 0.01 and 10000.

To enable LSSVR to have good generalization performance, we used DCQGSAA to globally optimize the parameters of σ^2 and C. First, the training samples were used to train LSSVR; then, the test samples were input to the trained LSSVR to obtain predicted values; the negative value of the root mean squared error (RMSE) between the predicted values of the test samples and their real values, −RMSE, was used as the fitness function. Clearly, the larger the fitness function, the higher the generalization ability of LSSVR. Then, LSSVR uses the best σ^2 and C to train the training samples and obtain the fitted values on these samples. At the same time, the amount of time required for this training process is recorded; then, the test samples are input into the trained LSSVR with the best σ^2 and C to obtain predicted values of the test samples.

For the failure prediction of liquid-fuel rocket engine thrust, we used the proposed model; then, we obtained the fitted curve of the training samples, the forecast values and forecast curve of the test samples, and the changing curve of the fitness function. These simulation results are shown in Figs. 2–4.

Figure 2 shows that after using DCQGSAA to optimize the key parameters of LSSVR, the optimized LSSVR better fit the training samples. Figure 3 shows that the optimized LSSVR can also better predict the values of the testing samples, which indicates that the optimized LSSVR has good generalization performance. Figure 4 shows that when using DCQGSAA to optimize the key parameters of LSSVR, after 26 iterations, the fitness function becomes stable.

As seen from Table 2, when using DCQGSAA, the simulation results show that the average relative error (ARE) on the testing samples is 0.2977%; when not using any intelligent algorithm, this error is 0.37%.

As seen from the above, when adopting DCQGSAA to optimize the parameters of LSSVR, the number of iterations is very small and reduced; moreover, from the simulation results, we find that ARE on the testing samples decreases. Therefore, the proposed mixture model is effective.

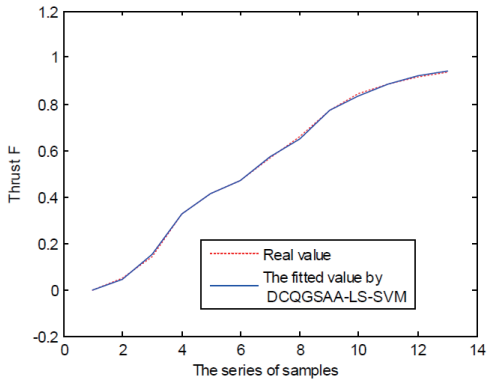

Fig. 2. (Color online) Fitted curve of the training samples using DCQGSAA.

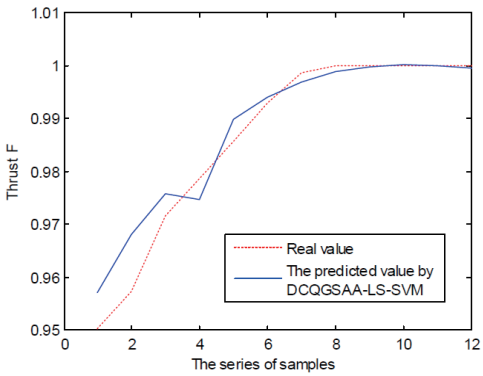

Fig. 3. (Color online) Predicted curve of the test samples using DCQGSAA.

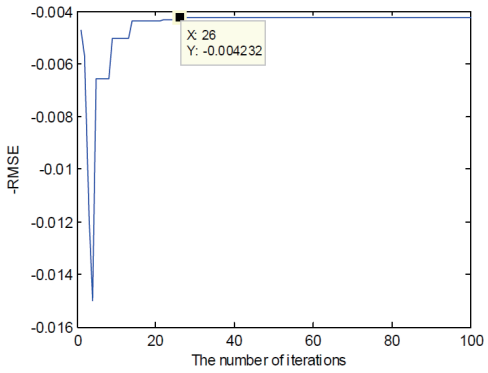

Fig. 4. (Color online) Changing curve of fitness function using DCQGSAA.

Table 2
Estimation results of test samples.

No.	Real value	Prediction value (using DCQGSAA)	Prediction value of LSSVR (Ref. 31)
1	0.9504	0.9570	0.9595
2	0.9574	0.9681	0.9694
3	0.9716	0.9759	0.9745
4	0.9787	0.9748	0.9835
5	0.9858	0.9899	0.9885
6	0.9929	0.9941	0.9931
7	0.9986	0.9970	0.9976
8	1.0000	0.9988	1.0013
9	1.0000	0.9997	1.0024
10	1.0000	1.0002	1.0024
11	1.0000	1.0001	1.0024
12	1.0000	0.9995	1.0024
ARE (%)		0.2977	0.37
Training time (s)		0 (Simulation result using MATLAB)	0.0006

6. Conclusion

In this study, after analyzing the characteristics of DCQGA, we combined DCQGA with SA and then proposed DCQGSAA. As the performance of LSSVR is extremely sensitive to its key parameters, we used DCQGSAA to optimize these parameters and established the hybrid nonparametric regression prediction model. This is of considerable significance to the health monitoring of a liquid-fuel rocket engine, and all types of sensors are used to monitor the factors that can affect the work condition of the engine from different aspects. The relationships between the thrust and these factors are nonlinear and difficult to express by some formulas. The fault characteristics of a liquid-fuel rocket engine show that its thrust is closely associated with various factors, and the fault samples are small and multidimensional. Thus, we used the proposed model to predict the thrust of a liquid-fuel rocket engine. The simulation results denote that the proposed model is effective for small and multidimensional samples, and thus, it is worth promoting.

References

1 Z. Xiao, S. J. Ye, B. Zhong, and C. X. Sun: Expert. Syst. Appl. **36** (2009) 273. https://doi.org/10.1016/j.eswa.2007.09.031
2 W. C. Chen, H. I. Lee, W. J. Deng, and K. Y. Liu: Expert. Syst. Appl. **32** (2007) 1148. https://doi.org/10.1016/j.eswa.2006.02.013
3 B. H. M. Sadeghi: J. Mater. Process. Tech. **103** (2000) 411. https://doi.org/10.1016/S0924-0136(00)00498-2
4 H. Jo, I. Han, and H. Lee: Expert. Syst. Appl. **13** (1997) 97. https://doi.org/10.1016/S0957-4174(97)00011-0
5 D. Z. Peng, Z. Yi, J. C. Lv, and Y. Xiang: Neurocomputing **71** (2008) 1748. https://doi.org/10.1016/j.neucom.2007.11.012
6 B. S. Ahn, S. S. Cho, and C. Y. Kim: Expert. Syst. Appl. **18** (2000) 65. https://doi.org/10.1016/S0957-4174(99)00053-6
7 J. Boritz and D. Kennedy: Expert. Syst. Appl. **9** (1995) 503. https://doi.org/10.1016/0957-4174(95)00020-8
8 X. Y. Fu and G. Chen: J. Ocean Univ. China **35** (2009) 85. https://doi.org/10.16411/j.cnki.issn1006-7736.2009.01.028
9 Y. Cissé, Y. Kinouchi, H. Nagashino, and M. Akutagawa: ITBM-RBM **21** (2000) 24. https://doi.org/10.1016/S1297-9562(00)90021-4
10 Z. J. Liu, C. Y. Wang, A. X. Liu, and Z. Niu: Future Gener. Comp. Syst. **20** (2004) 1119. https://doi.org/10.1016/j.future.2003.11.024
11 V. Vapnik: The Nature of Statistical Learning Theory (Springer, New York, 1995).
12 V. Vapnik: Statistical Learning Theory (Wiley, New York, 1998).
13 A. J. Smola and B. Schölkopf: Stat. Comput. **14** (2004) 199. https://doi.org/10.1023/B:STCO.0000035301.49549.88
14 S. W. Fei and Y. Sun: High Voltage Eng. **33** (2007) 81 (in Chinese). https://doi.org/10.13336/j.1003-6520.hve.2007.08.017
15 K. Z. Mao: IEEE Trans. Syst. Man Cybern. Syst. **34** (2004) 60. https://doi.org/10.1109/tsmcb.2002.805808
16 E. H. T. Francis and L. J. Cao: Omega **29** (2001) 309. https://doi.org/10.1016/S0305-0483(01)00026-3
17 S. Mukherjee, E. Osuna, and F. Girosi: Proc. 1997 IEEE Workshop on Neural Networks for Signal Processing (IEEE, 1977) 511–520. https://doi.org/10.1109/NNSP.1997.622433
18 U. Thissen, R. V. Brakel, A. P. D. Weijer, W. J. Melssen, and L. M. C. Buydens: Chemometr. Intell. Lab. **1** (2003) 35. https://doi.org/10.1016/S0169-7439(03)00111-4
19 J. A. K. Suykens and J. Vandewalle: Neural Process Lett. **9** (1999) 293. https://doi.org/10.1023/A:1018628609742
20 P. H. Chou, M. J. Wua, and K. K. Chen: Expert. Syst. Appl. **37** (2010) 4413. https://doi.org/10.1016/j.eswa.2009.11.087
21 C. L. Zhao, X. B. Sun, S. L. Sun, and T. Jiang: Expert. Syst. Appl. **38** (2011) 9908. https://doi.org/10.1016/j.eswa.2011.02.043
22 S. H. Min, J. Lee, and I. Han: Expert. Syst. Appl. **31** (2006) 652. https://doi.org/10.1016/j.eswa.2005.09.070

23 A. L. Chen, Z. M. Wu, and G. K. Yang: LS-SVM Based on Chaotic Particle Swarm Optimization with Simulated Annealing, J. Y. Cai, S. B. Cooper, and A. Li, Eds. (Springer, Heidelberg, 2006). https://doi.org/10.1007/11750321-9
24 S. S. Sancho, P. C. Mario, P. C. Fernando, and B. C. Carlos: Proc. 2002 Int. Conf. Artificial Neural Networks (ICANN). (Lecture Notes in Computer Science, 2002) 547–552. https://doi.org/10.1007/3-540-46084-5_89
25 V. Coilliefmb, P. C. Verbekel, and D. Wulfrr: Remote Sens. Environ. **110** (2007) 476. https://doi.org/10.1016/j.rse.2007.03.020
26 W. X. Xing and J. X. Xie: Modern Optimization Calculation Methods (Tsinghua University Press, Beijing, 1999) pp. 180–182.
27 A. Narayanan and M. Moore: Proc. 1996 IEEE International Conf. on Evolutionary Computation (IEEE, 1996) 61–66. https://doi.org/10.1109/ICEC.1996.542334
28 K. H. Han and J. H. Kim: Proc. 2000 Int. Congress on Evolutionary Computation (IEEE, 2000) 1354–1360.
29 K. H. Han and J. H. Kim: IEEE Trans. Evol. Comput. **6** (2002) 580. https://doi.org/10.1109/TEVC.2002.804320
30 S. Y. Li and P. C. Li: Quantum Computation and Quantum Optimization Algorithms (Harbin Institute of Technology Press, Harbin, 2009) pp. 69–101.
31 G. Tian, W. Zhang, Z. W. Yang, and Y. J. Song: Mech. Sci. Technol. **29** (2010) 63 (in Chinese). https://doi.org/10.13433/j.cnki.1003-8728.2010.01.019
32 H. Xiang and D. S. Wang: Proc. 2011 Int. Conf. Mechatronic Science, Electric Engineering and Computer (IEEE, 2011) 2526–2529. https://doi.org/10.1109/MEC.2011.6026007
33 X. H. Xue: J. Comput. Civ. Eng. **31** (2016) 04016041. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000607
34 L. Cao, F. Wu, J. Huang, and X. J. Chen: Chinese J. Sci. Instrum. **30** (2009) 1313. https://doi.org/10.19650/j.cnki.cjsi.2009.06.037
35 X. Xue and M. Xiao: Tunn. Undergr. Space Technol. **69** (2017) 171. https://doi.org/10.1016/j.tust.2017.06.019
36 Z. Wang, X. H. Wang, L. Z. Wang, X. F. Hu, and W. H. Fan: Proc. 2017 IEEE Int. Conf. Prognostics and Health Management (IEEE, 2017) 260–265. https://doi.org/10.1109/ICPHM.2017.7998338
37 N. Zhu, Z. G. Feng, and Q. Wang: J. Nanjing University Sci. Technol. (Nature Science) **33** (2009) 16. https://doi.org/10.14177/j.cnki.32-1397n.2009.01.027
38 J. Chai, J. Z. Du, K. K. Lai, and Y. P. Lee: Math. Probl. Eng. **2015** (2015) 1. https://doi.org/10.1155/2015/231394
39 S. W. Fei, Y. B. Miao, C. L. Liu, and X. B. Zhang: High Voltage Eng. **35** (2009) 509 (in Chinese). https://doi.org/10.13336/j.1003-6520.hve.2009.03.036

About the Authors

Hao Xiang received his B.S. and M.S. degrees from Henan Polytechnic University, China, in 2000 and 2004, respectively, and his Ph.D. degree from Nanjing University of Science and Technology, China, in 2013. He currently works in Yulin Normal University, China. His research interests are in AI, PHM, and industrial big data processing.