PSO Algorithm Improvement Based on Particle Tracing Analysis

Zhongguo Li[*1], Jie Hou[2], Kai Wang[3], Qinghua Liu[4]

[1, 2, 3] School of Mechanical Engineering, Jiangsu University of Science and Technology, Zhenjiang, China

[4] School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang, China

[*1]lzg1975@163.com; [2]houjie031237@126.com; [3]cumtchild@126.com; [4]giant_liu@163.com

***Abstract*- Improvement of Particle Swarm Optimization (PSO) algorithm is a significant work. In this paper, a practical method is proposed to instruct this task with recording all particle search positions and tracking the best particle shift process. Firstly, appropriate velocity bounds are obtained with tracking particle velocity components during iterations. Then particle initialization method is modified. Uniform Probability Random Value (UPRV) is substituted with Uniform Distributed Fixed Value (UDFV) to initiate particles. And it concludes a significant performance improvement. Stochasticity of results initialized with UDFV apparently decreases. It also makes PSO better cover with the search space which causes greater probability to obtain the global best. At least 3 particles can be competent for task with UDFV initialization method after analyzing the best particle shift process among all particles. That greatly enhances the algorithm speed. This paper can be a reference for application and improvement of PSO algorithm used in Support Vector Machine (SVM) parameter optimization.**

Keywords*- *Support Vector Machine; Vertical Load; Road Type Recognition; Initialization Method

I. INTRODUCTION

For its competence for small samples set recognition and global search, Support Vector Machine (SVM) has been widely and successfully applied in realms of mechanical fault diagnosis [1], medical diagnosis [2], image recognition [3] and so on. The width of the radial basis function and the penalty coefficient which both have the great influence on the recognition result need to be optimized. The optimization problem is a classical problem. There are lots of methods, such as genetic algorithms, ant colony algorithm gradient-based method [4] and so on. Recently Particle Swarm Optimization (PSO) algorithm which is proposed by Eberhart and Kennedy in 1995 becomes research focus of the optimization problem [1, 5]. Compared with other algorithms, PSO is simple and easy to realize [6]. It can also get higher quality solutions in shorter time. But the basic PSO algorithm is often trapped to a local minimum. The convergence speed of PSO algorithm declines rapidly at end of searching [7] and the optimization results have a poor stability. For the shortcomings of the basic PSO algorithm, many people proposed improved methods, such as the PSO combined with chaos algorithm to add its ergodicity [8, 9], the PSO combined with immune algorithm to add its escape strategy [10], hybridized with the AIS [11], and hybridized with gradient method [12], etc. These improvements make PSO algorithm have a wide range of applications in engineering, such as the SVM parameter optimization [13], monostable stochastic resonance detection [14], Cloud computing [15], power system control [16], and mechanical and electrical system fault detection [17]. These improvements refine the searching performance of the algorithm, but some methods increase the computation complexity and some methods make some assumptions and simplification before theoretical analysis [18]. This paper attempts to dissect the PSO algorithm and modify it. A marked improvement for PSO has been obtained.

II. SVM AND PSO ALGORITHM THEORY

Two-class linear SVM is the simplest among all SVM. It can be described by maximal margin problem. It is assumed that each sample x_i in the training sample set $\{x_i, i=1,\cdots,n\}$ belongs to one of two classes and its corresponding label y_i is $+1$ or -1. The SVM classification problem is to determine the optimal hyperplane which provides the maximal margin. The SVM classification problem can be depicted as the following optimization problem

$$\left.\begin{array}{l} \min \phi(\omega) = 1/2\|\omega\|^2 = 1/2(\omega\cdot\omega) \\ s.t \quad y_i[(\omega\cdot X_i)+b]-1\geq 0 \\ \qquad (i=1,2,\cdots,n) \end{array}\right\} \tag{1}$$

where ω is the coefficient of the optimal hyperplane, and b is the separate margin.

For an approximate linear question, slack variable $\xi_i > 0$ is introduced. The constraint is changed to $y_i[(\omega\cdot X_i)+b]-1+\xi_i \geq 0$. The sum of slack variables $\sum_{i=1}^{l}\xi_i$ is taken as the penalty term, which is multiplied with the penalty coefficient C and added to the target function. Then the target function changes to $\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}\xi_i$. The penalty coefficient C determines the proportion of slack variables in the target function. It decides the number of error samples. Larger C requires fewer wrong separated samples, which cause SVM over fitting and complex structure. On the contrary, less C make SVM under fitting. Both situations lead to lower recognition rate.

Some classification problems are linearly inseparable problems. Linear inseparable problem of low-dimensional space can become linearly separable problem after mapping into high-dimensional space. But the computational complexity will increase. This problem can be solved with the kernel function. In this paper, radial basis function $K(x, x_i) = \exp(-\frac{\|x - x_i\|^2}{\sigma^2})$ is chosen as kernel function. The width σ of the radial basis function determines the disperse degree of sample feature parameters after space transformation. The value of σ should be consistent with original features distribution width of samples. It causes low recognition rate whether σ is too big or too small. For above reasons, penalty coefficient C and kernel function parameter σ are needed to be optimized in the application of SVM.

PSO algorithm is derived from the simulation on the bird flock's migration and aggregation during searching food [19]. In the basic PSO algorithm, the particles velocity and position need to be randomly initialized at the beginning. The initial position and velocity of the d th dimensional parameter of the i th particle are determined by

$$x_{id}(0) = r_{0x}(x_{\max d} - x_{\min d}) + x_{\min d} \tag{2}$$

$$v_{id}(0) = r_{0v} v_{\max d} \tag{3}$$

where $i = 1,2,...,m$, m is the population of swarm, $d = 1,2,...,D$, D is the search space dimension, r_{0x} and r_{0v} are uniform probability random numbers in the interval [0, 1], $x_{\max d}$ and $x_{\min d}$ are the maximum value and the minimum values of the d th dimension parameter position respectively which determine the optimization range, and $v_{\max d}$ is the maximum velocity value of the d th dimensional parameter.

Then iterations are performed in the following. Two optimal solutions of two parameters are updated after iteration. One optimal solution is previously found by a particle itself. The other one is searched by whole swarm so far. Then the next step velocity of the i particle is determined as

$$v_{id}(t+1) = v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t)) \tag{4}$$

where t is the step number of the current iteration, c_1 and c_2 are non-negative acceleration constant named as "self-confidence" and "swarm confidence" respectively [6], r_1 and r_2 are uniform distribution random numbers in [0, 1].

The particle velocity should be restricted. When $v_{id} > v_{maxd}$, take $v_{id} = v_{maxd}$. And if $v_{id} < -v_{maxd}$, take $v_{id} = -v_{maxd}$. Then the position update formula of the i th particle is

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)\,. \tag{5}$$

Corresponding particle positions are similarly limited with $x_{\max d}$ and $x_{\min d}$.

The convergence speed of basic PSO algorithm declines rapidly at end of searching. So it is difficult to find the optimal solution [20]. The inertia weight ω in the velocity item is introduced to refine algorithm. Then the update formula of the i th particle velocity becomes

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t)) \ . \tag{6}$$

Inertia weight ω dynamically diminish as following linear formula

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T_{\max}} t \tag{7}$$

where $\omega_{\max}$, $\omega_{\min}$ are the maximum value and the minimum value of ω respectively, and $T_{\max}$ is the maximum iteration number. Acceleration constant, c_1 and c_2, are dynamically adjusted to improve the accuracy of optimization as

$$c_1 = R_1 + \frac{R_2 \times t}{T_{\max}} \tag{8}$$

$$c_2 = R_3 - \frac{R_4 \times t}{T_{\max}} \tag{9}$$

where R_1, R_2, R_3 and R_4 are initially set values.

III. EXPERIMENTAL RESULTS

Different type roads provide different excitation to vehicle, which results in the variation of vertical load. Load variation is the main cause of vehicle component fatigue damage. The variance of the vertical load is also an indicator related to the driving manipulation stability and riding comfort. In the vehicle simulation experiment on indoor road, it can be used to solve how to load drive in the experiment if simulated road types could be correctly recognized with vertical load. In this paper, wheel force transducer is employed to collect the vertical load on different roads on car test ground. There is washboard road, stone road, gravel road, fish-scale road and cement road. 17 samples of five kind roads are taken as the sample set. For relatively little samples, the SVM is employed as classifier whose two parameters should be optimized.

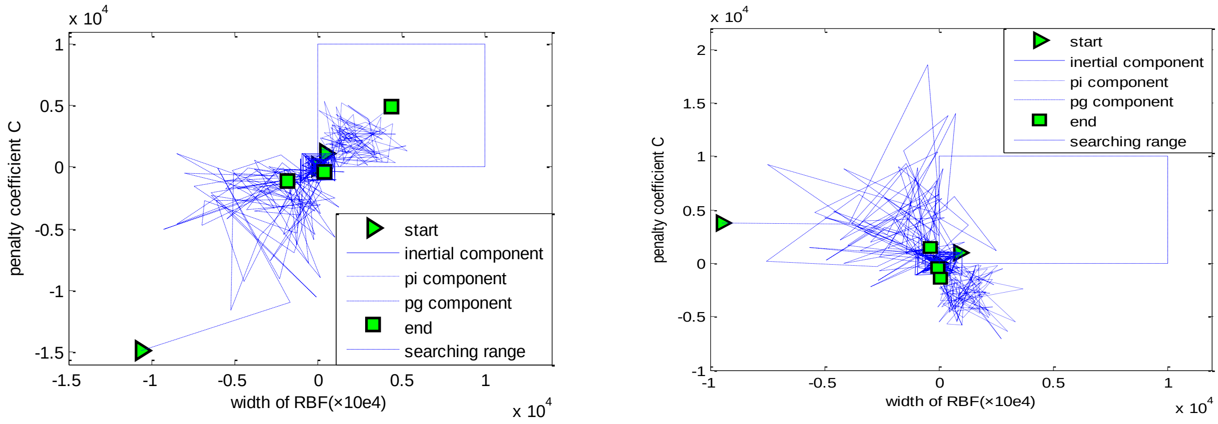
For all sample data, static wavelet transform is used to remove random noise signals. And spectral subtraction is adopted to remove periodic noise, such as the engine vibration, etc. Then bior1.5 wavelet is employed to take 3-layer wavelet decomposition. After normalized to [0, 1], means and variances of 3 layer detail coefficients and the 3rd layer approximation coefficients are selected as the feature parameters to recognize road roughness [21]. 12 of the 17 samples are applied for SVM training. Left 5 samples which belong to 5 kind roads are test samples. To eliminate the influence of sample choice, five times cross-validations are performed with changing the training samples and test samples. Finally, the average recognition rate of the five cross-validations is calculated and its reciprocal acts as the target function.

At beginning of PSO algorithm, initialization should be completed. The population of swarm m sets 20. The dimension D is 2. The maximum weighting factor $\omega_{\max}$ is 1.2. The minimum $\omega_{\min}$ is 0.4. The number of iterations sets 100. Set $R_1 = 1$, $R_2 = 0.5$, $R_3 = 6$, $R_4 = 2$. Because the feature parameters is normalized among [0, 1], search range of the kernel function parameter σ sets [10e-5, 1]. The corresponding maximum velocity v_{max1} is 0.09. Penalty coefficient C ranges among [1, 10000] and its maximum velocity v_{max2} is 1000.

When the current target function value is less than the global or particle optimal during the iteration, the optimal is updated to the current position. The program is repeated for five times and results are 72.5%, 67.5%, 72.5%, 72.5% and 72.5%. Then the update condition of optimal position is slightly changed to when the current target function value is less than or equal to the global or particle optimal. Because there are many positions equal to the current optimal position during the iteration, the current optimal will be more frequently updated. This maybe enhance the diversity of particles position and maybe helpful to find the global optimal solution. After this adjustment, results are 72.5%, 72.5%, 72.5%, 72.5% and 67.5%. It can be found that just sequences of 2 group results are difference and values are same. But the optimal average recognition rates are different among 5 results which are affected by the randomly generated initial position and initial velocity. Therefore, the results of above two cases cannot exactly witness which one is better.

In order to remove the influence of the initial value, Uniform Distributed Fixed Value (UDFV) is used to initial particle position. Because there is no or little correlation between the two-dimensional parameters in SVM, initialization values of 2 parameters are independently dispersed in uniformly distribution of 1-dimensional. It can be found in Figures 3(a). The searching range is divided into m segments in accordance with the population of swarm and the middle point of segment is taken as each particle initial position. The other work is same as before. When the current target function value is less than the global or particle optimal in update condition, five results are 70%, 72.5%, 72.5%, 72.5%, 72.5%. When it is less than or equal to, result is the same. There is also no improvement between thus two update conditions. In the following study, the former update condition is still adopted. But the minimum value is 70% when initialization with UDFV, which is larger than 67.5% when Uniform Probability Random Value (UPRV) is adopted.

There is no certain theory to determine the velocity bound choice of two parameters. When no velocity bound are set, velocity value produced in (6) is observed. According to (6), the velocity has three parts: the first one with term ω called the inertial component; the second part named p_i component caused by individual optimal; and the third part is p_g component caused by global optimal. The 10th particle velocity components as iteration with 2 initialization methods are shown in Figure 1 (it is similar to other particles).



(a) With UDFV initialization (b) With UPRV initialization

Fig.1 Comparison of three velocity component between 2 initialization methods

(The RBF width showed in the picture has been multiplied by 10e4 for better display). With UDFV initialization, the starting point of inertial component, p_i component and p_g component are [1 047.1 914.94], [0 0] and [-12 610 -10 423] respectively. As is shown in the Fig. 1 (a), some component values are far larger than the particle maximum allowed position or velocity. After partly amplified, it can be found that the absolute value of p_g component is relatively larger than the rest 2 components. It is similar to UPRV initialization. Therefore bound of velocity should be set. Too large velocity bound may cause poor accuracy and fail to find global optimal. At the same time, too small bound will invalidate the velocity calculation in (6) and PSO cannot search the whole range. Then bound of the velocity radial basis width is set to 0.09, and that of penalty coefficient C is 1000. After limiting velocities, velocities of the two parameters with UDFV initialization are shown in Figure 2.

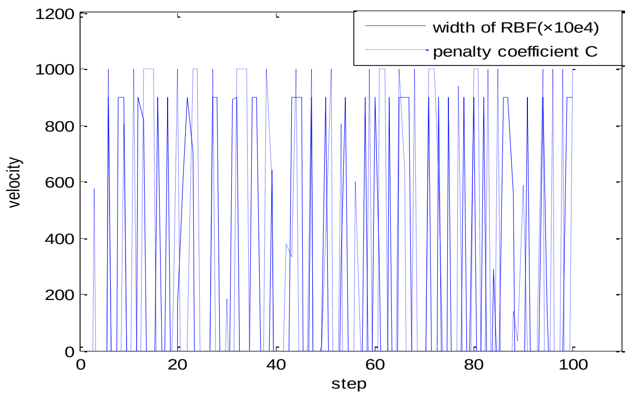
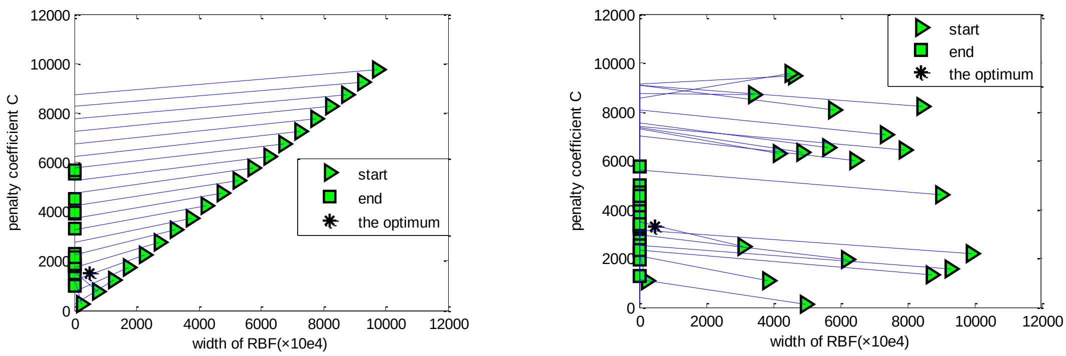


Fig. 2 Velocity variety as iteration after limiting maximums

In the figure, 2 velocities reach to the limited values only at some times. Velocity calculations do not invalidate. At the same time the limited values and number of iterations can guarantee that any particle can search the whole optimization space, which proves that the choice of the maximum velocity is suitable. With the above velocity bounds, positions of all particles $x_i(t)$ changing as iteration are shown in Figure 3. Two initialization methods are compared at the same time.

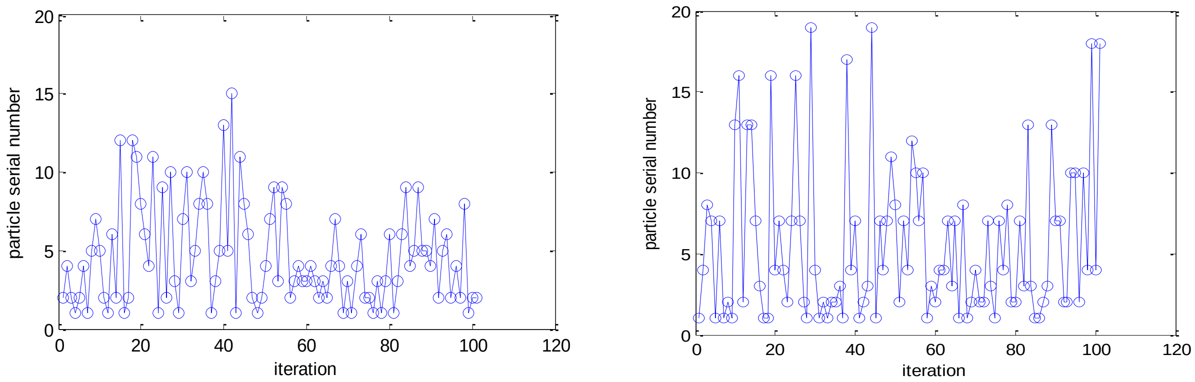


(a) With UDFV initialization (b) With UPRV initialization

Fig. 3 Comparison of all particle search path between 2 initialization methods

With UDFV initialization method, the searched range almost occupies the whole upper left corner of the entire plane. Most particle trajectories are some parallel to each other, so it performs a more uniform searching. But the particle coverage area is uncertain with UPRV initialization method. And its searching is some coarse. With attractions of the individual optimal and global optimal, UDFV initialization gets greater chance to find optimum solution than UPRV initialization. Finally all particles converge to several points on the vertical axis. That is because the optimal parameters are not unique when recognition rate reach to 72.5%.

During iterations, shift process of the optimal particle among all particles can contribute to analyze the optimization process. The optimal particle shift processes between two initialization methods are compared in Figure 4.



(a) With UDFV initialization (b) With UPRV initialization

Fig. 4 The best particle serial number variety as iteration

Shift process of the optimal particle is related to complexity of the target function along with variation of parameters. If the target function is flat, shift of global optimal particle is more frequent. Shift process of the optimal particle is also related to the search direction. If searching direction is right, shift times will decrease. Because of fewer shift, it can be concluded that the search direction of UDFV initialization is better than that of UPRV initialization from Fig. 4. The optimal particle of the UDFV initialization changes among fewer particles, then it means fewer particles can be competent for searching task. Therefore, population of particles is selected 2, 3, 4 and 5 one by one to search optimization. With UDFV initialization, the average recognition rates are 65%, 72.5%, 72.5% and 72.5% respectively. With UPRV initialization, they are 60%, 60%, 65%, 67.5% respectively. With UDFV initialization, the recognition rate becomes the highest 72.5% when population of swarm is only 3. That greatly increases the PSO algorithm speed.

IV. MATHEMATICAL ANALYSIS

Assumed the range of 1 dimension parameter is [0, 10000]. With UDFV initialization method, 20 particles initiation position are 500, 1000…, 9000 and 9500 respectively.

With UPRV initialization method, uniform probability density adopted by rand() function is

$$p(x)=\begin{cases}\dfrac{1}{b-a}, a<x<b \\ 0, \text{others}\end{cases} \tag{10}$$

Where the bottom bound a is 0, and the top bound b is 10000.

According (10), the probability of a particle fall in [0, 9500] is 0.95. That of all 20 particle fall such range is 0.95^{20}=0.3585. So the probability of at least 1 particle fall in [9500, 10000] is 1-0.3585=0.6415. That of both 2 parameter at least 1 particle fall in [9500, 10000] is 0.6415×0.6415=0.4115. Similarly that fall in [1, 500] is also 0.4115. Then the probability of both at least 1 particle in [1, 500] and at least 1 particle in [9500, 10000] is greatly less than 0.4115. So initialization particle range with UPRV method is difficult to get same range with UDFV method. And initialization particles are strictly uniform dispersed with UDFV initialization method. But those with UPRV initialization method are only obeying uniform probability. The strict uniform and large range make UDFV initialization show good performance than UPRV initialization.

V. CONCLUSION

In this paper, further analysis of PSO algorithm on SVM parameter optimization is performed with tracking all particle searching positions, velocities and the optimal particle shift process. Several conclusions are the following.

(1) Among components of particle velocity, p_g component is too large. So velocity should be limited during optimization process. Maximum velocities of the radial basis width σ and the penalty coefficient C are set to 0.09 and 1000 respectively.

(2) After the update condition is changed from less than to less than or equal to, results did not show improvement.

(3) The opportunity to obtain the optimal value with UDFV initialization is larger than that with UPRV initialization. At least 3 particles can be competent to optimize with UDFV initialization, which greatly increases the PSO algorithm speed. The strict uniform and large range make UDFV initialization show good performance than UPRV initialization.

(4) The recognition rate of SVM changes slowly with the radial basis width σ and the penalty coefficient C. The improvement of PSO algorithm shows outstanding performance in SVM parameters optimization.

ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation (51008143).

REFERENCES

[1] CAI Zhen-yu, SHI Qiao-bo. Fault diagnosis of rotating machinery based on PSO-SVM model. Coal Mine Machinery, 2011, 32(11):276-278 (in Chinese).

[2] Nahla H. B., Andrew P. B., Mohamed N. H. B.. Intelligible support vector machines for diagnosis of diabetes mellitus. IEEE Transactions on Information Technology in Biomedicine, 14(4), 2010:1114-1120.

[3] John M., Mike R., Paul D., et al. A morphological approach to pipe image interpretation based on segmentation by support vector machine. Automation in Construction, 2010, 19(7):875-883.

[4] LIU Chang-ping, FAN Ming-yu, WANG Guang-wei, et al. Optimizing parameters of support vector machine based on gradient algorithm. Control and Decision, 2008, 23(11): 1291-1295, 1300 (in Chinese).

[5] LIU Jin-ping, YU Jin-xiang. Parameter estimation of multiple linear regression models based on the improved particle swarm optimization algorithm. Computer Engineering ＆ Science, 2010, 32(4):101-105 (in Chinese).

[6] Swagatam D., Ajith A., Amit K.. Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives, Studies in Computational Intelligence. Heidelberg: Springer Berlin, 2008: 1–38.

[7] ALFI A.. PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems. Acta Automatica Sinica, 2011, 37(5): 541-549.

[8] WANG Yan, SUN Xiang-feng, LI Ming. Training method for support vector machine based on chaos particle swarm optimization. Computer Engineering, 2010, 32(23): 189-191 (in Chinese).

[9] LIU Dao-hua, YUAN Si-cong, LAN Yang, et al. Method of particle swarm optimization based on the chaos map [J]. Journal of Xidian University (Natural Science), 2010, 37(4):764-769 (in Chinese).

[10] XIE Zheng-gui, ZHONG Shao-dan, WEI Yu-ke. Modified particle swarm optimization algorithm and its convergence analysis. Computer Engineering and Applications, 2011, 47(1):46-49 (in Chinese).

[11] K. Thanushkodi, K. Deeba. Hybrid intelligent algorithm (improved PSO with AIS) for multiprocessor job scheduling. European Journal of Scientific Research, 2012, 70(4): 539-553.

[12] K. Kawakami, Z. Meng. Improvement of particle swarm optimization. PIERS Online, 2009, 5(3): 261-264.

[13] WANG Jia, XU Wei-hong. Parameter optimization of mixed kernel SVM based on momentum particle swarm optimization. Journal of Computer Applications, 2011, 31(2): 501-503,516 (in Chinese).

[14] LI Ji-meng, CHEN Xue-feng, HE Zheng-jia. Adaptive monostable stochastic resonance based on PSO with application in impact signal detection. Journal of Mechanical Engineering, 2011, 47(21):58-63 (in Chinese).

[15] Suraj Pandey, Linlin Wu, Siddeswara Guru, et al. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. 24th IEEE International Conference on Advanced Information Networking and Applications, Perth, Australia, 2010: 400-407.

[16] S. K. Sinha, R. N. Patel and R. Prasad. Application of GA and PSO tuned fuzzy controller for AGC of three area Thermal-Thermal-Hydro power system. International Journal of Computer Theory and Engineering, 2010, 2(2):1793-8201.

[17] ZUO Lei, Hou Li-gang, ZHANG Wang, et al. Analog circuit fault diagnosis based on particle swarm optimization support vector machine. Systems Engineering and Electronics, 2010, 32(7):1553-1556 (in Chinese).

[18] KANG Qi, WANG Lei, AN Jing, et al. Approximate dynamic programming based parameter optimization of particle swarm systems. Acta Automatica Sinica, 2010, 36(8): 1171-1181 (in Chinese).

[19] T. Niknam, A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem, Applied Energy, 2010, 87:327-339.

[20] Chander, A., Chatterjee A., Siarry P. A new social and momentum component adaptive PSO algorithm for image segmentation. Expert Systems with Applications, (2010), doi: 10. 1016/ j. eswa. 2010. 09. 151.

[21] Li Zhongguo, Zhang Weigong, Kuang jun, et al. Wavelet feature extraction in road roughness recognition based on dynamic load [J]. Journal of Jiangsu University (Natural Science Edition), 2007, 28(4): 305-308 (in Chinese).

Zhongguo Li 1975.6 He is an associate professor in Jiangsu University of Science and Technology. He got Ph.D. in Southeast University of China in 2007. His research interests cover data analysis of automotive testing, machinery fault diagnosis.

Jie Hou 1987.1. He is a M.E candidate in Jiangsu University of Science and Technology. His research interests include fault diagnosis and pattern recognition.

Kai Wang 1985.5. He is a Master's degree candidate in Department of Mechanical Engineering, Jiangsu University of Science and Technology. His research interests include digital image processing.

Qinghua Liu 1977.3 He is an associate professor in Jiangsu University of Science and Technology. He got Ph.D. in Southeast University of China in 2008. His research interests include automobile electronics and intelligent transportation.