Neurocomputing ■ (■■■■) ■■■–■■■

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Melt index prediction using optimized least squares support vector machines based on hybrid particle swarm optimization algorithm

Huaqin Jiang, Zhengbing Yan, Xinggao Liu *

Zhejiang University, Control Department, State Key Laboratory of Industrial Control Technology, Zheda Road 38, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 1 May 2012
Received in revised form
3 March 2013
Accepted 4 March 2013
Communicated by J. Zhang

Keywords:
Industrial polypropylene manufacture
Melt index prediction
Immune clone particle swarm optimization
Ant colony optimization
Least squares support vector machines

ABSTRACT

Melt index (MI) is considered as one of the most important variables of the quality, which determines the product specifications. Thus, a reliable estimation of MI is crucial in the quality control of the practical processes in the propylene polymerization (PP). An optimal soft sensor, named the least squares support vector machines with Ant Colony-Immune Clone Particle Swarm Optimization (AC-ICPSO-LSSVM), is therefore proposed. It combines the advantages of the high accuracy of LSSVM and the fast convergence of PSO. Furthermore, the immune clone (IC) method is introduced into the PSO algorithm to make the particles of ICPSO diverse and enhance global search capability for avoiding the premature convergence and local optimization of the conventional PSO algorithm. Besides, to widen data range, improve search precision and convergence efficiency, and avoid premature convergence, Ant Colony Optimization (ACO) is introduced to find the initial particles for PSO algorithm. The resultant hybrid AC-ICPSO algorithm is then applied to optimize the parameters of LSSVM, so the optimal prediction model of melt index, AC-ICPSO-LSSVM, is obtained. As the comparative basis, the models of ICPSO-LSSVM, PSO-LSSVM, and LSSVM are also developed respectively. Based on the data from a real PP production plant, a detailed comparison of the models is carried out. These models are also compared with RBF method reported in the open literature. The research results reveal the prediction accuracy and validity of the proposed approach.

1. Introduction

Polymer production has a critical influence in some aspects of industry, military, economy and so on. Melt index (MI) of polypropylene determines the grade of the product as well as the other product properties, so it is considered as one of the most important quality variables in the practical industrial polymerization process. However, the MI of polypropylene is usually sampled online and then measured offline with an analytical procedure in the laboratory, which is not only costly but also time consuming (2–4 h) [1]. Therefore, the development of MI on-line estimation model is significant, not only as an on-line sensor but also as a forecasting system.

A number of researchers have attempted to find many modeling approaches for the prediction of the melt index or the application in the industrial process [2–7]. Embiruçu et al. [8] and Tian et al. [9] developed models based on process mass and energy balance. Feil et al. [10] presented a semi-mechanistic modeling to monitor the industrial processes. Because polymerization processes are usually highly nonlinear processes, and neural networks can approximate any continuous nonlinear functions, neural network (NN) and the improved NN have been applied to system modeling and control [11–14]. Shi et al. [15] and Shi and Liu [16] developed several soft-sensor models for MI prediction based on ICA-MSA-RBF (independent component analysis, multi-scale analysis and RBF neural network) and PCA-MS-RBF. Zhang et al. [17] sequentially presented a bootstrap aggregated neural network model, which could overcome the problems of the single NN, such as over-fitting and the lack of generalization capability. To avoid the problem of NN, Han et al. [18] employed three approaches, which were SVM (supported vector machines), PLS (partial least square) and artificial neural networks, for MI estimation of San and PP process. Detailed comparison researches among the standard SVM, LSSVM, and weighted LSSVM (WLSSVM) models were carried out by Shi and Liu [19]. Although these works have been done in MI prediction, greater performance in prediction and better universality of the estimation model are still the first-line goal in academia and industrial community.

In this paper, a novel ant colony-immune clone particle swarm optimization (AC-ICPSO) algorithm is proposed to optimize the parameters of LSSVM model, which uses ACO algorithm to find the initial particles and combines mutation and clone factor to efficiently control the global search of LSSVM model. The basic PSO [20–24] algorithm often suffers the problem of being trapped

* Corresponding author. Tel.: +86 571 87951970.
E-mail addresses: liuxg@iipc.zju.edu.cn, lxg@zju.edu.cn (X. Liu).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.03.006

Nomenclatures

PSO	particle swarm optimization
ACO	ant colony optimization
IC	immune clone
ICPSO	immune clone particle swarm optimization
AC-ICPSO	ant colony- immune clone particle swarm optimization
LSSVM	least squares support vector machine
MI	melt index
NN	neural network
ICA	independent component analysis
KKT	Karush-Khun-Tucker
IQR	interquartile range
CSTR	continuous stirred-tank reactor
FBR	fluidized-bed reactor
MAE	mean absolute error
MRE	mean relative error
RMSE	root mean square error
STD	standard deviation of absolute error
TIC	Theil's Inequality Coefficient
PP	polypropylene
T	temperature
p	pressure
l	level of liquid
a	percentage of hydrogen in vapor phase
$f1$, $f2$, $f3$	flow rate of 3 streams of propylene
$f4$	flow rate of catalyst
$f5$	flow rate of aid-catalyst

into local optimum which leads to premature convergence. Thus immune clone(IC) algorithm is further proposed to enhance the variety of colony particles and the reliability of global convergence while optimizing the parameters of LSSVM model.

The rest of the paper is organized as follows: Section 2 provides the theoretic descriptions of least squares support vector machine (LSSVM), ant colony optimization (ACO), particle swarm optimization (PSO), and immune clone(IC). The AC-ICPSO-LSSVM is presented. Case study is presented and discussed in Section 3. Finally, Section 4 concludes this paper.

2. Least squares support vector machines

The support vector machine method introduced by Vapnik is a valuable tool for solving pattern recognition and classification problems [25,26], and is employed to solve many problems successfully in lots of fields [27–29]. According to the practice, to overcome the disadvantage of slow training velocity in large-scale problem, least squares support vector machines (LSSVM) was presented by Suykens, which transforms the quadratic optimization problem into that of solving a linear equation set [30].

Suppose that there is a set of data $\{x_i, y_i\}_{i=1}^n$ with the input data x_i and the corresponding goal y_i. In the modeling of least square support vector machine, the error ξ_i quadratic norm is taken as the LSSVM's loss function. The following optimization problem is considered:

$$\min_{\omega,b,\xi} J(\omega,\xi)=\frac{1}{2}\omega^T\omega+\gamma\frac{1}{2}\sum_{i=1}^{n}\xi_i^2 \quad \gamma>0 \tag{1}$$

Subject to the equality constraints

$$y_i=\omega^T\varphi(x_i)+b+\xi_i, \quad i=1,2,\cdots,n \tag{2}$$

And, the model of LSSVM is as follows:

$$f(x)=\omega^T\varphi(x)+b \tag{3}$$

In Eq. (1), $J(\omega, \xi)$ is objective function, $1/2\omega^T\omega$ is used as a flatness measurement function. γ is a punishment factor, which determines the tradeoff between the training error and the model flatness. $\xi_i=\alpha_i/\gamma$ is the error variable. In Eq. (2), the nonlinear mapping φ maps the input data into a so-called high dimensional feature space, where a linear regression problem is obtained and solved. In Eq. (3), b is the bias and ω is a weight vector of the same dimension as the feature space.

The LSSVM's loss function is different from the standard SVM [31]. For the optimization problem, the Lagrange function is introduced as follows:

$$L(\omega,b,\xi,\alpha)=J(\omega,\xi)-\sum_{i=1}^{n}\alpha_i[\omega^T\varphi(x_i)+b+\xi_i-y_i] \tag{4}$$

With $\alpha_i(i=1,2,\cdots,n)$ Lagrange multipliers, the conditions for optimality

$$\begin{cases}\frac{\partial L}{\partial\omega}=0\rightarrow\omega=\sum_{i=1}^{n}\alpha_i\varphi(x_i)\\ \frac{\partial L}{\partial b}=0\rightarrow\sum_{i=1}^{n}\alpha_i=0\\ \frac{\partial L}{\partial\xi_i}=0\rightarrow\alpha_i=\gamma\xi_i \quad i=1,\cdots,n\\ \frac{\partial L}{\partial\alpha_i}=0\rightarrow\omega^T\varphi(x_i)+b+\xi_i-y_i=0 \quad i=1,\cdots,n\end{cases} \tag{5}$$

can be written as the following set of linear equations, eliminating ω, ξ:

$$\begin{bmatrix}0 & 1 & \cdots & 1\\ 1 & K(x_1,x_1)+\frac{1}{\gamma} & \cdots & K(x_1,x_n)\\ \vdots & \vdots & \vdots & \vdots\\ 1 & K(x_1,x_n) & \cdots & K(x_n,x_n)+\frac{1}{\gamma}\end{bmatrix}\begin{bmatrix}b\\ \alpha\end{bmatrix}=\begin{bmatrix}0\\ y_1\\ \vdots\\ y_n\end{bmatrix} \tag{6}$$

where $\alpha=[\alpha_1,\cdots,\alpha_n]^T$, and the Mercer's condition $K(x_i,x_j)=\varphi(x_i)^T\varphi(x_j)$ is so-called kernel function. The resulting LSSVM model for function estimation becomes

$$f(x)=\sum_{i=1}^{n}\alpha_iK(x,x_i)+b \tag{7}$$

where α_i, b are the solution to Eq. (6), $K(\bullet,\bullet)$ represents the high dimensional feature spaces that is non-linearly mapped from the input data x.

Common kernel functions of SVM can be divided into the local kernel function and the whole kernel function. The whole kernel function has the overall situation characteristic, and the data points which are far between them have influence to the value of kernel function. On the other hand, the local kernel function has the topicality, which allows the dataset to affect the value of kernel function only in small domain.

There are some different forms of kernel functions $K(x, x_i)$

(1) The linear kernel: $K(x,y)=x^Ty$
(2) The poly kernel: $K(x,y)=(1+x^Ty/\sigma^2)^d$
(3) The RBF kernel: $K(x,y)=\exp\{-||x-y||^2/2\sigma^2\}$

In this paper, the RBF kernel is applied because it is frequently used for various regression problems for its high resolution power [18,19].

Please cite this article as: H. Jiang, et al., Melt index prediction using optimized least squares support vector machines based on hybrid particle swarm optimization algorithm, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.006

So the kernel parameter σ needs to be optimized, which will be introduced in Section 3.

3. AC-ICPSO algorithm

AC-ICPSO is a cooperation of two heuristic algorithms, i.e. ant colony optimization (ACO) and immune clone particle swarm optimization (ICPSO). The proposed hybrid optimization is split into two stages, where ACO is used in the first stage and ICPSO in the second stage. An ant in ACO can be regarded as a particle, and the dimension of every ant equates to the number of the destination notes which are the two parameters of LSSVM (the regularization factor γ and the kernel parameter σ). To find the best parameters in ICPSO, the number of ants is N in this paper. The basic process of AC-ICPSO is: after the initialization of ACO, find the best ants one by one until a complete solution is constructed, which is regarded as initial position of particles in ICPSO. Then update the position and the velocity of every particle followed by cloning a part on particles. To end with ICPSO algorithm, we find the best parameters of LSSVM, so the optimal AC-ICPSO-LSSVM model for predicting MI is proposed. D represents the number of parameters of LSSVM which need to be optimized, and also represents the dimension of every ant and every particle. N is the number of ants or particles.

3.1. Ant colony optimization algorithm

ACO is an improved version of the Ant System which was proposed by Colomi, Dorigo and Maniezzo [32–34]. This algorithm simulates the natural behavior of ants, including mechanisms of cooperation and adaptation. Each ant deposits some pheromone along the path it has passed through. The involved ants are steered toward the local and global optimization through a mechanism of feedback of the pheromone and pheromone intensity. Eventually the better path which has more pheromone deposited can be found and all the ants will favor the particular path, which is expected to be the optimum or a near-optimum solution for the target problem.

There are three core steps of ACO: selection probability, local pheromone update, and global pheromone update.

3.1.1. Selection probability

After initializing, each ant seeks next pixel according to the following regulations. If ant's current node is node i, and will transform to next node j, the probability is p_{ij}. The probability depends on two parameters, the trail level τ^{α}, and the heuristic information η^{β}.

$$p_{ij}=\begin{cases}\arg\max_{k\notin \text{tabu}}[\tau_{ik}^{\alpha}\eta_{ik}^{\beta}] & q\le q_0\\ \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{k\notin \text{tabu}}\tau_{ik}^{\alpha}\eta_{ik}^{\beta}} & j\notin \text{tabu},\quad q>q_0\\ 0 & j\in \text{tabu},\quad q>q_0\end{cases} \tag{8}$$

$$\eta_{ij}=\frac{E}{d(P_j,G)}=\frac{E}{\sqrt{(x_{P_j}-x_G)^2+(y_{P_j}-y_G)^2+(z_{P_j}-z_G)^2}} \tag{9}$$

where α indicates important degree of rudimental pheromone, τ_{ij} presents total amount of pheromone deposited on the edge between node i and node j, η_{ij} presents visibility degree between the node P_j and the goal point G, β controls the relative importance of visibility. tabu indicates the set of unallowed passed nodes of ant. $q\in(0,1)$ is random number, $q_0\in(0,1)$is constant. E is positive constant. $d(P_j,G)$ indicates the length between the two points P_j and G.

3.1.2. Local pheromone update

After each ant had passed an edge, updating pheromone on each edge according to the following regulations:

$$\tau_{ij}\leftarrow(1-\rho_1)\tau_{ij}+\rho_1\tau_{ij}\frac{z^*}{z} \tag{10}$$

$$\text{fitness}=(y_i^*-\hat{y}_i)^2/2 \tag{11}$$

where z^* is denoted as the fitness of the best result found so far, z is denoted as the fitness of current solution generated. The ratio is then multiplied with the pheromone level so as to renew the present pheromone level. $y_i^*, \hat{y}_i$ denote the measured value and the predicted result, respectively. The fitness indicates the error of prediction. So the lower the fitness is, the better the solution is. ρ_1 represents the evaporation of trail. The local pheromone update is performed by all the ants after each construction step.

3.1.3. Global pheromone update

$$\tau_{ij}\leftarrow\tau_{ij}+\rho_2\Delta\tau_{ij}^{BS} \tag{12}$$

$$\Delta\tau_{ij}^{BS}=\left(\frac{\bar{z}}{z^*}-1\right)\tau_{ij} \tag{13}$$

where ρ_2 is global pheromone volatilization coefficient. $\bar{z}$ is denoted as the average fitness of all the particles in the swarm, and z^* is the best fitness found so far. According to the Eq. (13), it encourages the best ant to release more pheromone on the trail if the average fitness is far away from the best. The advantage of global updating is that it reduces the computation effort of pheromone update.

3.2. Immune clone PSO algorithm

PSO algorithm is a method for optimizing hard numerical functions, originally developed by Kennedy and Eberhart [35], based on a social psychological metaphor of behaviors of flocks of birds and schools of fish. By randomly initializing a set of candidate solutions, the PSO successfully leads to a global optimum [36–39].

In PSO, each potential solution is represented as a particle, which is dynamically adjusted by updating the velocity of each particle, according to the experience of neighbor particles and its own experience. In detail, two properties associated with each particle are position x and velocity v. In each iteration, a fitness function is evaluated for all the particles and is used to determine the number of iterations. They are given as

$$x^i=(x^{i1},x^{i2},\cdots,x^{iD}) \tag{14}$$

$$v^i=(v^{i1},v^{i2},\cdots v^{iD}) \tag{15}$$

where D is size of the particles for the D-dimension problem. $v^i\in[-v_{\max},v_{\max}]$ and $v_{\max}$ is the designed maximum velocity. The velocity of each particle is updated by keeping track of two best positions. One is its current best position of a particle, named p_{best}, and the other is the best position of the whole population, called g_{best}. Hence, the velocity and position of a particle are updated as follows:

$$v_{k+1}^i=w_kv_k^i+c_1r_1(p_{\text{best}}-x_k^i)+c_2r_2(g_{\text{best}}-x_k^i) \tag{16}$$

$$x_{k+1}^i=x_k^i+v_{k+1}^i \tag{17}$$

$$w_k=w_{\max}-(w_{\max}-w_{\min})\times(k-1)/\text{iter}_{\max} \tag{18}$$

where w_k is inertia weight, controlling the impact of the previous velocity on its current one. c_1 and c_2 are positive constants, called acceleration coefficients. r_1 and r_2 are stochastic coefficients that are

Please cite this article as: H. Jiang, et al., Melt index prediction using optimized least squares support vector machines based on hybrid particle swarm optimization algorithm, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.006

uniformly distributed in the interval [0, 1]. k is the current number of iteration, and $\text{iter}_{\max}$ is the maximal number of iteration.

According to Eq. (16,17), the population of particles tends to cluster together from different directions. The PSO algorithm runs until the termination criterion is satisfied.

The immune clone algorithm (IC) has the desirable characteristics in optimization and offers significant advantages over the conventional PSO method. IC can enhance the immune ability through the mutation of antibody colony. An antibody has a great fitness and thus it must be preserved and developed as a new particle. IC algorithm can enhance the diversity in a colony and the reliability of global convergence. Fig. 1 shows a sketch of ICPSO algorithm, where a number of particles undertake mutation operation to further change their positions to increase the chance for the swarm to escape from local optima.

(1) Cloning operation. In the immune system, operation of clone generates offspring by asexual reproduction. There is an antibody $Q=\{x^1, x^2,\cdots, x^D\}$, which is the local best position p_{best} at one iteration. The cloning operation is defined as follows:

$$\begin{aligned}\text{clone}(Q) &= \text{clone}(\{x^1, x^2, \cdots, x^D\}) \\ &= \{\{x_1^1, x_1^2, \cdots, x_1^D\}, \{x_2^1, x_2^2, \cdots, x_2^D\}, \cdots, \{x_m^1, x_m^2, \cdots, x_m^D\}\} \\ &= \{Q_1, Q_2, \cdots, Q_m\} \end{aligned} \tag{19}$$

In order to let all antibodies have the same clone scale, here we define the following number:

$$m = [s \times \text{Popsize}/i] \quad i = 1, 2, \cdots, \text{Popsize} \tag{20}$$

where s is a constant number, Popsize is the size of the particles, $\lceil x \rceil$ rounds the elements of x to the nearest integers toward infinity.

(2) Mutating operation. The main strategy of IC algorithm is the operation of mutate. It can generate a new antibody by mutating one or multiple dots of the selected antibody. The mutating operation is defined as follows:

$$\begin{aligned}\text{mutate}(Q) &= \text{mutate}\left(\left\{x^1, \cdots, x^i, \cdots x^D\right\}\right) \\ &= \left\{x^1, \cdots, x^{i'}, \cdots x^D\right\} \end{aligned} \tag{21}$$

$$x^{i'} = p_{\text{best}} + \text{randn} \tag{22}$$

the value of antibody's i dimension changes in the operation of mutate. randn returns a pseudorandom, scalar value drawn from a normal distribution with mean 0 and standard deviation 1.

(3) Fitness evaluation. The fitness indicates whether a particle is excellent or not. A particle with a lower fitness can be selected and come into next generation. The function of fitness is Eq. (11).

(4) Selection. The best particles are selected to the next generation according their fitness to enhance the quality of the colony.

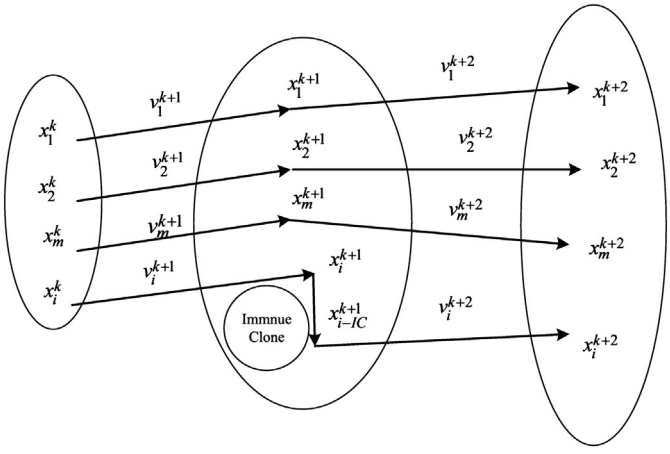


Fig. 1. Sketch of ICPSO.

3.3. Hybrid AC-ICPSO-LSSVM algorithm

Based on the above mentioned Section 3.1–3.2, the AC-ICPSO algorithm optimizes the parameters of LSSVM, which are the regularization factor γ and the kernel parameter σ, named AC-ICPSO-LSSVM algorithm. The σ value is related to the distance between the training points and the smoothness of the interpolation of the model. The γ value is in charge of the trade-off between the smoothness of the model and its accuracy. The flow chart of AC-ICPSO-LSSVM is shown in Fig. 2, where the whole optimization process is shown. Furthermore, the detailed procedure of AC-ICPSO-LSSVM algorithm can be described in 2 stages with 16 steps as follows:

Step 1. Initialization: α, β, ρ_1, ρ_2, q_0,E,$c_1, c_2, r_1, r_2, v, w_{\max}, w_{\min}$, N (*Popsize*), $iter_{\max}$and initial pheromone $\tau_{ij}(0)$.

Stage 1

Step 2: An ant is randomly placed on the set of machines.
Step 3: Ant moving: each ant in ant group moves a step based on the probability information shown in Eq. (8,9). Repeat this step until a complete solution is constructed.
Step 4: Local updating: update the pheromone trail using the local pheromone update shown in Eqs. (10,11).
Step 5: Global updating: when each ant arrives at G, the pheromone trail is updated by using global pheromone as shown in Eqs. (12,13) to generate the solution closer to the best solution.
Step 6: Repeat step 2 to 5 for N ants to generate a swarm of particles.

Stage 2

Step 7: According to the model LSSVM, we get the initial result $\hat{y}_i$. Evaluate the initial fitness of each particle according to Eq. (11). Then we let the particles with the best fitness to be the initial p_{best}. From the p_{best} particles, we can get the initial g_{best}.
Step 8: According to the Eq. (18), calculate the weights w_k.
Step 9: Update the position and the velocity of every particle x^i according to the Eqs. (16,17). In this paper, the parameters are taken as, $c_1=c_2=2.0$. If $v^i > v_{\max}$, refine $v^i = v_{\max}$; if $v^i < -v_{\max}$, refine $v^i = -v_{\max}$.
Step 10: Set $i=1$, we made m clones of the p_{best} of each particle according to Eqs. (19,20), then mutate the particles which are generated by cloning according to Eqs. (21,22).
Step 11: According to the prediction of the model LSSVM, calculate the fitness of particles which are generated by cloning and mutating.
Step 12: Update the p_{best} according to the fitness. If the fitness of current particle is lower than the fitness of the initial p_{best}, we should use the current p_{best} and update the g_{best}.
Step 13: $i=i+1$, if $i >$ Popsize, go to Step 14; else go back to Step 10.
Step 14: Taking the updated p_{best} as the parameters of the model LSSVM, we can obtain the new prediction. Then we update the p_{best} and the g_{best} according to the new fitness.
Step 15: Repeat Step 8 to 14 until $iter_{\max}$ is met, otherwise proceed to Step 16.
Step 16: Take the updated g_{best} as the optimized parameters of the model LSSVM. The final solution of the optimizing problem is found.

Please cite this article as: H. Jiang, et al., Melt index prediction using optimized least squares support vector machines based on hybrid particle swarm optimization algorithm, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.006

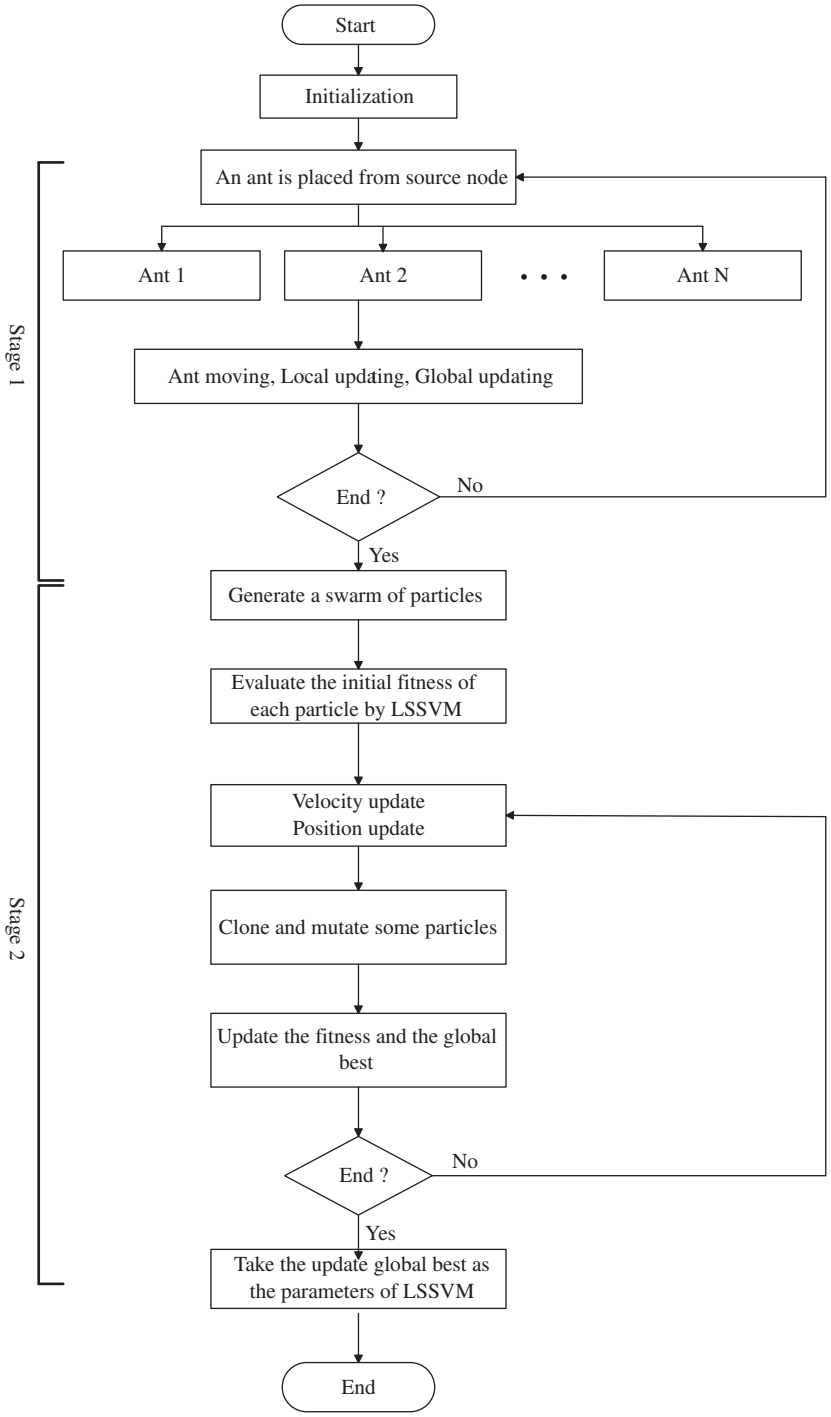


Fig. 2. The flow chart of hybrid AC-ICPSO-LSSVM algorithm.

The proposed AC-ICPSO methods, which combine ant colony optimization (ACO) and immune clone particle swarm optimization (ICPSO), can inherit the advantage of global convergence and avoid premature convergence of the conventional PSO algorithm. The immune clone algorithm (IC) has the desirable characteristics in optimization and offers significant advantages over the conventional PSO method.

4. Case study

A simple schematic diagram of a propylene polymerization process, which is recently operated for commerce in a real plant, is shown in Fig. 3. There are a chain of reactors in series, two continuous stirred-tank reactors (CSTR) and two fluidized-bed reactors (FBR).The polymerization reaction begins with a liquid phase in the first two CSTRs and is completed in a vapor phase in the third and fourth FBRs to produce the powdered polymer product. The melt index of the polypropylene, which depends on reactant composition, reactor temperature, etc., can determine the property and quality of the product.

To develop a prediction model to estimate the MI, a group of easy-measured variables, which are acquired from the historical logs recorded in a real propylene polymerization plant, have been chosen. Thirty process variables were measured in our study.

Please cite this article as: H. Jiang, et al., Melt index prediction using optimized least squares support vector machines based on hybrid particle swarm optimization algorithm, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.006

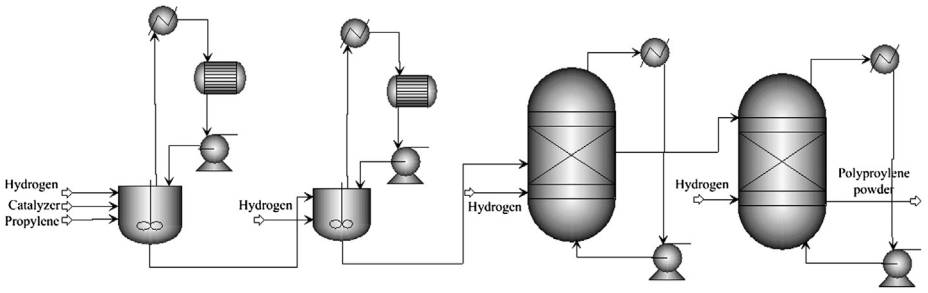


Fig. 3. General scheme of propylene polymerization.

A pool of process variables (T, p, l, a, $f1$, $f2$, $f3$, $f4$, $f5$) are chosen as the input data according to the workers' experience and our mechanism analysis [16,19], where T, p, l, a are process temperatures, pressure, level of liquid, and percentage of hydrogen in vapor phase respectively; $f1$, $f2$, $f3$ are flow rate of 3 streams of propylene respectively; $f4$, $f5$ are flow rate of catalyst and aid-catalyst respectively. Following, we take the percentage of hydrogen in vapor phase as illustration. As shown in the following figure, there is a strong correlation between the percentage of hydrogen in vapor phase and melt index, which was therefore chosen as the input variable.

As shown in Fig. 4, the melt index changes over time, these including the data of 9 operation variables constitute the data set of the current work, which are the time series data of the considered real polypropylene producing process and are therefore the dynamic data. Data are filtered to discard abnormal situations and to improve the quality of the prediction results. The variables are normalized with respect to their standard deviation value and mean value. Data from the time records are separated into training, test, generalization sets. Here the data sets are selected from the time series of recorded plant data as sample number. The sample number represents sample time, 50 of which are used as the training set, 20 are used as the test set, and the remaining 15 are used as generalization set. It should be noted that the current dynamic time series data are from the same brand of polypropylene production, not from the brand transition for the multistage productions in polypropylene manufacturing process. The former is a slowly varying dynamic process; the latter is dynamic process with intense disturbance. And it is also noted that the training set is taken from the same batch with the test set, while the generalization set is obtained from another one. So in this paper the performance of test can reflect the model's prediction accuracy and the performance of generalization can reflect the model's universality.

The initial parameters of the AC-ICPSO algorithm are given as follows: in the ACO algorithm, the ant number N is set to 30, $\alpha=\beta=1.0$, $\rho_1=0.5$, $\rho_2=0.7$, $q_0=3.0$, $E=1.0$. In the PSO algorithm, the dimension number $D=2$, the particle number Popsize$=N=30$, c_1 and $c_2$are initialized to 2.0. The initial value of velocity vector is constrained into $[-1, 1]$, and the modified $v_{\max}$ is set to 2. For the inertia weight w_k, we set $w_{\max}=0.8$,$w_{\min}=0.2$. The algorithm stops when the iter$_{\max}$ meets 100. In the IC algorithm, the parameter s is set to 1.

In this paper, the difference between the output of the models and the real output is considered as the error and represented in several ways, including mean absolute error (MAE), mean relative error (MRE), root mean square error (RMSE), standard deviation of absolute error (STD), and Theil's Inequality Coefficient (TIC) [40]. They are defined as the following, respectively:

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{23}$$

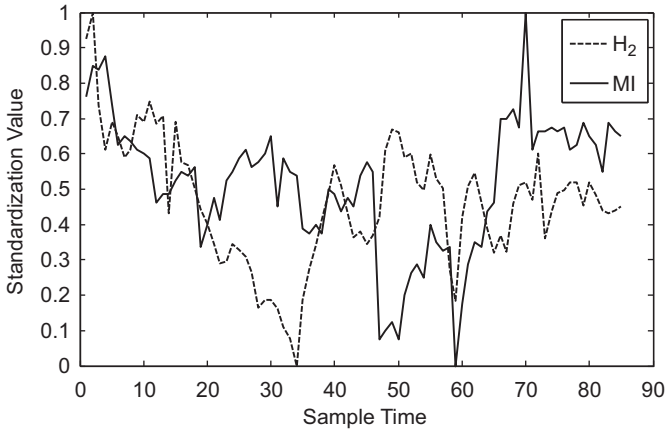


Fig. 4. The correlation of percentage of hydrogen in vapor phase and MI.

$$\mathrm{MRE}=\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i-\hat{y}_i|}{y_i}\times 100\% \tag{24}$$

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \tag{25}$$

$$\mathrm{STD}=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(e_i-\bar{e})^2} \tag{26}$$

$$\mathrm{TIC}=\frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{n}y_i^2}+\sqrt{\sum_{i=1}^{n}\hat{y}_i^2}} \tag{27}$$

where $e_i=y_i-\hat{y}_i$, $\bar{e}=1/n\sum_{i=1}^{n}e_i$, and y_i, $\hat{y}_i$ denote the measured value and predicted result, respectively.

In order to investigate the performance of the proposed, are also AC-ICPSO-LSSVM model, the other two models, i.e., LSSVM, PSO-LSSVM developed to be as comparison basis, where LSSVM is based on a previous research of the group [19]. The detailed comparison of test performance among the four models and RBF method reported in the open literature [15] is listed in Table 1. The MAE, MRE, and RMSE confirm the prediction accuracy of the proposed methods, the STD indicates the predictive stability of the methods, and the TIC indicates the level of agreement between the proposed models and the real process.

It shows that the LSSVM model with AC-ICPSO has the best performance over all, with MAE of 0.0518, compared with 0.0538 and 0.0590 obtained from the corresponding ICPSO-LSSVM and PSO-LSSVM, 0.0842 of LSSVM which was reported by Shi and Liu [19], 0.0848 of ICA-MS-RBF in [15]. In terms of MRE, the AC-ICPSO-LSSVM's prediction accuracy is 1.97% and that of ICPSO-LSSVM is 2.05%, much better than LSSVM (3.66%), decreasing 46%, 44%

Please cite this article as: H. Jiang, et al., Melt index prediction using optimized least squares support vector machines based on hybrid particle swarm optimization algorithm, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.006

Table 1
Prediction of LSSVM models before and after optimization on the test dataset.

Model	MAE	MRE (%)	RMSE	STD	TIC	Correlation coefficient
ICA-MS-RBF [15]	0.9848	3.50	0.1034	–	0.0211	–
LSSVM [19]	0.0842	3.66	–	0.1116	0.0240	–
PSO-LSSVM	0.0590	2.24	0.1041	0.0943	0.0209	0.7369
ICPSO-LSSVM	0.0538	2.05	0.0965	0.0931	0.0193	0.7639
AC-ICPSO-LSSVM	0.0518	1.97	0.0938	0.0913	0.0188	0.7789

Table 2
Prediction of LSSVM models before and after optimization on the generalization dataset.

Model	MAE	MRE (%)	RMSE	STD	TIC	Correlation Coefficient
ICA-MS-RBF [15]	0.0757	2.98	0.0799	–	0.0156	–
LSSVM [19]	0.0662	2.60	0.0313	0.0751	0.0138	–
PSO-LSSVM	0.0574	2.40	0.0746	0.0773	0.0153	0.8949
ICPSO-LSSVM	0.0455	1.90	0.0597	0.0616	0.0122	0.9381
AC-ICPSO-LSSVM	0.0411	1.72	0.0545	0.0560	0.0111	0.9550

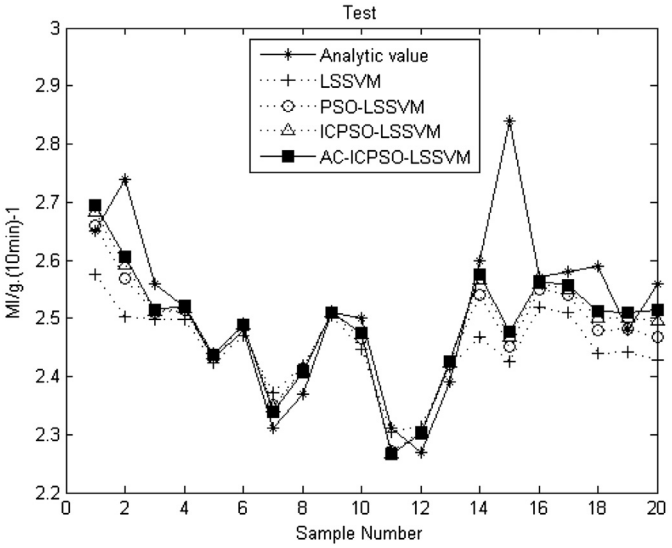


Fig. 5. Prediction of the optimized models on the test dataset.

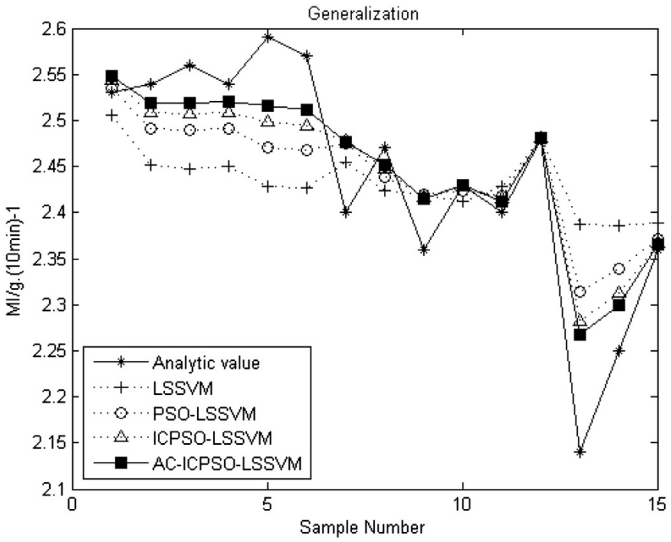


Fig. 6. Prediction of the optimized models on the generalization dataset.

respectively. Similar results are observed in terms of RMSE, with a decrease from 0.1041 to 0.0938. Moreover, the STD obtained by AC-ICPSO-LSSVM model is 0.0913, while that of ICPSO-LSSVM is 0.0931, that of PSO-LSSVM is 0.0943 and that of LSSVM is 0.1116. So the AC-ICPSO-LSSVM model and ICPSO-LSSVM model have better stability. It is noted that the TIC of AC-ICPSO-LSSVM (0.0188) is quite acceptable when compared with that of ICPSO-LSSVM (0.0193), PSO-LSSVM (0.0209), LSSVM (0.0240), ICA-MS-RBF (0.0211), which indicates a good level of agreement between the proposed model and the real process. All in all, the MAE, MRE, STD and TIC of the AC-ICPSO-LSSVM model are the most smallest, with percentage decreasing of 38%, 46%, 18%, 22% compared to that of the pure LSSVM obtained by Shi [19].

The correlation coefficients are also listed in Table 1. The correlation coefficients of PSO-LSSVM, ICPS-LSSVM and AC-ICPSO-LSSVM are 0.7369 0.7639, 0.7789 respectively, which means the prediction value of AC-ICPSO-LSSVM has a better correlation with the analytic value than PSO-LSSVM and ICPS-LSSVM.

A more distinct illustration, in which shows how better the AC-ICPSO-LSSVM model works than the other three models on the testing dataset, is shown in Fig. 5. The curve marked with asterisks is the real MI value of the real plant, while the curves marked with crosses and circles are the prediction values by LSSVM model and PSO-LSSVM, respectively. The results predicted by ICPSO-LSSVM model and AC-ICPSO-LSSVM model are depicted by the curves marked with triangles and squares respectively. Obviously, the LSSVM models with optimization (PSO-LSSVM, ICPSO-LSSVM, AC-ICPSO-LSSVM) are better than the pure LSSVM model. As visual, the AC-ICPSO-LSSVM model yields consistently good prediction and performance a little better than the other models do.

Table 3
Computation time of models (Intel 3.2GHz/8G RAM).

Model	Computation time(s)
PSO-LSSVM	0.15
ICPSO-LSSVM	0.43
AC-ICPSO-LSSVM	1.62

To illustrate the universality of the proposed four models, a digital comparison on the generalization dataset is listed in Table 2. Giving an MRE of 1.72%, the AC-ICPSO-LSSVM model defeats the other three models again, with a decrease of 33.85% compared with 2.60% of LSSVM obtained by Shi and Liu [19]. And the same trend occurs in terms of MAE, RMSE, STD, TIC and correlation coefficient. Moreover, the visual comparison is shown in Fig. 6. The curves marked with crosses, circles, triangles, and squares are still the predicted values of LSSVM, PSO-LSSVM, ICPSO-LSSVM, and AC-ICPSO-LSSVM respectively, while the desired value curve is marked with asterisks. Clearly, the ICPSO-LSSVM model and PSO-LSSVM model show better prediction results than the pure LSSVM model. Moreover, the AC-ICPSO-LSSVM model predicts the closer to the real MI value than the ICPSO-LSSVM model. Obviously, this phenomenon can be clearly seen from points 12–15, where the result of AC-ICPSO-LSSVM model is the closest to the analytic value. Meanwhile, the same result can be seen in Fig. 5.

The computation time (Intel Xeon 3.20G/8G RAM) of PSO-LSSVM, ICPSO-LSSVM, AC-ICPSO-LSSVM is listed in Table 3.

Please cite this article as: H. Jiang, et al., Melt index prediction using optimized least squares support vector machines based on hybrid particle swarm optimization algorithm, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.006

The computation time increases with the complexity of models. However, even the computation time of the most complex model, AC-ICPOSO-LSSVM, is 1.62 s, it is suitable for the on-line predictor and further researches on control.

5. Conclusion

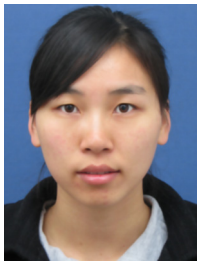
The prediction of the melt index of polypropylene in a real industrial plant using AC-ICPSO-LSSVM model, which aimed at continuous optimizing problems in MI prediction, is presented in this paper. To avoid the deficiency of the PSO algorithm, the ICPSO is proposed, which overcame the premature convergence of PSO. To further avoid premature convergence and widen data range, the ACO is used to find the initial particles for ICPSO algorithm. For comparison, LSSVM, PSO-LSSVM, ICPSO-LSSVM and AC-ICPSO-LSSVM are developed and evaluated. The estimation errors can be reduced by using ICPSO-LSSVM model, and can be further reduced by using AC-ICPSO-LSSVM model. The application of the proposed training method to the test and generalization data gotten from an industrial polymerization plant demonstrates its effectiveness and reliability. The AC-ICPSO-LSSVM model predicts MI with MRE of 1.97% on the test dataset, which is much more accurate than PSO-LSSVM model with the MRE of 2.24% and much better than the best literature results reported by Shi and Liu [19]. These models are also compared with RBF method reported in the open literature. The research results reveal the prediction accuracy and validity of the proposed approach.

Acknowledgments

This work is supported by Joint Funds of NSFC-CNPC of China (Grant U1162130), National High Technology Research and Development Program (863, Grant 2006AA05Z226) and Zhejiang Provincial Natural Science Foundation for Distinguished Young Scientists (Grant R4100133), and their supports are thereby acknowledged.

References

[1] S.S. Bafna, A.M. Beall, A design of experiments study on the factors affecting variability in the melt index measurement, J. Appl. Polym. Sci. 65 (2) (1997) 277–288.
[2] F. Ahmed, S. Nazir, Y.K. Yeo, A recursive PLS-based soft sensor for prediction of the melt index during grade change operations in HDPE plant, J. Chem. Eng. 26 (1) (2009) 14–20.
[3] L. Xie, Y.J. Liu, H.Z. Yang, F. Ding, Modeling and identification for non-uniformly periodically sampled-data systems, IET Control Theory Appl. 4 (5) (2010) 784–794.
[4] Y. Chen, X.G. Liu, Modeling mass transport of propylene polymerization on Ziegler–Natta catalyst, Polym. J. 46 (2005) 9434–9442.
[5] R.K. Al Seyab, Y. Cao, S. Yang, A. Case Study of Predictive Contorl for the Alstom Gasifier Problem, in: Proceedings of UK Control, 2004.
[6] E.H. Lee, T.Y. Kim, Y.K. Yeo, Prediction of the melt index in a high-density polyethylene process, J. Chem. Eng. Jpn. 40 (10) (2007) 840–846.
[7] Z.H. Luo, S.H. Wen, Z.W. Zheng, Modeling the effect of polymerization rate on the intraparticle mass and heat transfer during propylene polymerization in a loop reactor, J. Chem. Eng. Jpn. 42 (8) (2009) 576–580.
[8] M. Embiruçu, E.L. Lima, J.C. Pinto, Continuous soluble Ziegler–Natta ethylene polymerizations in reactor trains. I. Mathematical modeling, J. Appl. Polym. Sci. 77 (7) (2000) 1574–1590.
[9] C.Q. Tian, Y.F. Liao, X.T. Li, A mathematical model of variable displacement swash plate compressor for automotive air conditioning system, Int. J. Refrig. 29 (2) (2006) 270–280.
[10] B. Feil, J. Abonyi, P. Pach, S. Nemeth, P. Arva, M. Nemeth, G. Nagy, Semi-mechanistic models for state-estimation-Soft sensor for polymer melt index prediction, in: Proceedings of the 7th International Conference on Artificial Intelligence and Soft Computing, 2004, pp. 1111–1117.
[11] C. Peng, Y.C. Tian, M.O. Tade, State feedback controller design of networked control systems with interval time-varying delay and nonlinearity, Int. J. Robust Nonlinear Control 18 (12) (2008) 1285–1301.
[12] Z.H. Xiong, Y.X. Xu, J. Zhang, J. Dong, Batch-to-batch control of fed-batch processes using control-affine feedforward neural network, Neural. Comput. Appl. 17 (4) (2008) 425–432.
[13] T. Virkki-Hatakka, A. Bulsari, Scalability of neural network models makes industrial use easier, Pap. Puu 84 (7) (2002) 463–466.
[14] C. Lu, B.X. Shi, L. Chen, An expandable on-chip BP learning neural network chip, Int. J. Electron. 90 (5) (2003) 331–340.
[15] J. Shi, X.G. Liu, Y.X. Sun, Melt index prediction by neural networks based on independent component analysis and multi-scale analysis, Neurocomputing 70 (2006) 280–287.
[16] J. Shi, X.G. Liu, Melt index prediction by neural soft-sensor based on multi-scale analysis and principal component analysis, Chin. J. Chem. Eng 13 (6) (2005) 849–852.
[17] J. Zhang, Q.B. Jin, Y.M. Xu, Inferential estimation of polymer melt index using sequentially trained bootstrap aggregated neural networks, Chem. Eng. Technol. 29 (4) (2006) 442–448.
[18] I.S. Han, C. Han, C.B. Chung, Melt index modeling with support vector machines, partial least squares, artificial neural networks, J. Appl. Polym. Sci. 95 (4) (2005) 967–974.
[19] J. Shi, X.G. Liu, Melt index prediction by weighted least squares support vector machines, J. Appl. Polym. Sci. 101 (1) (2006) 285–289.
[20] Y. Marinakis, M. Marinaki, A hybrid genetic-Particle Swarm Optimization Algorithm for the vehicle routing problem, Expert. Syst. Appl. 37 (2) (2010) 1446–1455.
[21] S. Yagiz, H. Karahan, Prediction of hard rock TBM penetration rate using particle swarm optimization, Int. J. Rock. Mech. Min. 48 (3) (2011) 427–433.
[22] C.J. Lin, S.J. Hong, The design of neuro-fuzzy networks using particle swarm optimization and recursive singular value decomposition, Neurocomputing 71 (1–3) (2007) 297–310.
[23] S. Onüt, U.R. Tuzkaya, B. Dogaç, A particle swarm optimization algorithm for the multiple-level warehouse layout design problem, Comput. Ind. Eng 54 (4) (2008) 783–799.
[24] G.L. Chen, W.Z. Guo, Y.Z. Chen, A PSO-based intelligent decision algorithm for VLSI floorplanning, Soft Comput. 14 (12) (2010) 1329–1337.
[25] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[26] V.N. Vapnik, Statistical Learning Theory, John Wiley, New York, 1998.
[27] K.S. Shin, T.S. Lee, H.J. Kim, An application of support vector machines in bankruptcy prediction model, Expert Syst. Appl. 28 (1) (2005) 127–135.
[28] J.L. Yang, H.X. Li, Y. Hu, A probabilistic SVM based decision system for pain diagnosis, Expert Syst. Appl. 38 (8) (2011) 9346–9351.
[29] B. Wang, F. Wan, Classification of single-trial EEG based on support vector clustering during finger movement, Lect. Notes. Comput. Sci. 5552 (2009) 354–363.
[30] J.A.K. Suykens, J. Vandewalle, Least squares support vector machines classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.
[31] X.W. Yang, J. Lu, G.Q. Zhang, Adaptive pruning algorithm for least squares support vector machine classifier, Soft Comput. 14 (7) (2010) 667–680.
[32] A. Colorni, M. Dorigo, V. Maniezzo, Positive Feedback as a Search Strategy, Politecnico di Milano, Italy, Technical Report no. 91-016 1991.
[33] A. Colorni, M. Dorigo, V. Maniezzo The Ant System: an Autocatatlytic Process, Politecnico di Milano, Italy, Technical Report no. 91-016, 1991.
[34] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. 26 (1) (1996).
(35) J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks Conference, 4, 1995, pp. 1942–1948.
[36] X.H. Yuan, A.J. Su, H. Nie, Y.B. Yuan, L. Wang, Unit commitment problem using enhanced particle swarm optimization algorithm, Soft Comput. 15 (2011) 139–148.
[37] Z.H. Che, PSO-based back-propagation artificial neural network for product and mold cost estimation of plastic injection molding, Comput. Ind. Eng. 58 (4) (2010) 625–637.
[38] H. Modares, A. Alfi, S. Naghibi, B. Mohammad, Parameter estimation of bilinear systems based on an adaptive particle swarm optimization, Eng. Appl. Artif. Intell. 23 (7) (2010) 1105–1111.
[39] Z.H. Zhan, J. Zhang, Y. Li, H.S.H. Chung, Adaptive particle swarm optimization, IEEE Trans. Syst. Man. Cybern. Part. B. Cybern. 39 (6) (2009) 1362–1381.
[40] D.J. Murray-Smith, Methods for external validation of continuous system simulation models: a review, Math. Compt. Model. Dyn. Syst. 4 (1) (1998) 5–31.



Huaqin Jiang received her B.Sc. degree in IOT Engineering from the Jiangnan University, China, in 2008. Then, she received her M.Sc. degree in Control Science and Technology from the Zhejiang University, China, in 2012. Her research topics include propylene polymerization modeling, multivariate statistical process control, optimization and control of complex industrial process, support vector machines.

Zhengbing Yan received his Ph.D. degree in Control Science and Technology from the Zhejiang University, China, in 2010. His research topics include propylene polymerization modeling, multivariate statistical process control, optimization and control of complex industrial process.

Xinggao Liu is a Professor of Control Science and Engineering at the Zhejiang University. He received his Ph.D. degree in Control Science and Technology from the Zhejiang University in 2000. Then, he was a Post Doctoral Fellow in Automation Department of Tsinghua University from 2000–2002. He received his B.Sc. (1991) degree from the Tianjin University, M.Sc. (1996) degrees from the Zhejiang University. His research interests lie in process systems engineering (PSE) and include multivariate statistical modeling, model identification, optimization and control of complex industrial process, and dynamic optimization.

Please cite this article as: H. Jiang, et al., Melt index prediction using optimized least squares support vector machines based on hybrid particle swarm optimization algorithm, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.03.006