Melt Index Prediction by Adaptively Aggregated RBF Neural Networks Trained with Novel ACO Algorithm

Jiubao Li, Xinggao Liu, Huaqin Jiang, Yundu Xiao

State Key Laboratory of Industrial Control Technology, Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, People's Republic of China

Received 27 February 2011; accepted 20 September 2011
DOI 10.1002/app.35688
Published online 29 December 2011 in Wiley Online Library (wileyonlinelibrary.com).

ABSTRACT: Three estimation models of polypropylene (PP) process to infer the melt index, an important quality indicator determining product specification, are presented. Radial basis function (RBF) neural network (NN) is used to develop the models because of its capacity of fitting the complex relationship in PP process. A novel ant colony optimization (ACO) algorithm is also proposed and used to solve the optimization problem of the continuous linking weights when training the RBF NN. Based on the RBF NN and the novel ACO algorithm, a single NN model is developed. However, a single network cannot always work well due to some defects (such as overfitting) of a NN. Thus, as an improvement of the single NN model, several RBF NN trained with a certain objective are combined, and the aggregated NN model is obtained. To make the aggregated NN more robust and effective, an adaptive method of assigning the combinational weight to every individual network is applied to the former aggregated NN model and finally an adaptive aggregated NN model is achieved. Further researches of the three models are carried out on the data from a real industrial plant, and the prediction result shows that the performance of the obtained prediction models is better and better with every improvement step taken as above. The adaptive aggregated NN model works best, and the satisfying prediction error it provides depicts its prediction accuracy and universality, as well as an application prospect in PP process. © 2011 Wiley Periodicals, Inc. J Appl Polym Sci 125: 943–951, 2012

Key words: poly(propylene); melt; computer modeling; aggregated RBF networks; ACO

INTRODUCTION

The industry of polypropylene (PP) production has a critical influence in the world, especially in aspects of related industries, military, economy, and so on. The increasing global competition pushes the polymer industry to improve the product quality and reduce the cost. Consequently, the advanced monitoring and controlling of the properties of the products in PP process becomes a very important strategy in this field. PP MI, which is the key parameter in determining the product's property and quality controlling of practical industrial process, is defined as the mass rate of extrusion flow through a specified capillary under certain condition of temperature and pressure.[1] There are certain instruments developed to measure the MI directly, but these instruments are very expensive and difficult to maintain. Therefore, the PP is usually sampled on line and then measured off-line with an analytical procedure in the laboratory to obtain the MI of the product. However, the procedure takes a long time of 2–4 h, so the measured MI cannot be used to instruct the production. Off-specification products and enormous losses in profit are always resulted in.[2] The MI needs to be given timely and accurately, thereby it makes the development of MI on-line estimation model, not only as an on-line sensor but also as a forecasting system, very necessary.

As the MI is difficult to be measured directly, it is common to figure it out in an indirect way. Between the MI and some other easy-measured variables, there are some certain relationships that can be used to develop the MI prediction model. To infer the difficult-to-measure variable from easy-to-measure variables, the chemical and physical relationships can be used; thus, an on-line analyzer can be constructed with referring to the exact mechanism of polymerization process. However, the approach to model from the industrial process mechanism[2–6] is faced with a big challenge, due to the sophisticated engineering activity and the relatively high complexity of kinetic behavior and operation of polymer plants. The chemical and physical reactions in the reactors are so complicated that modeling the reactors or the reaction processes that happen in the

Correspondence to: X. Liu (liuxg@iipc.zju.edu.cn).

Contract grant sponsor: Zhejiang Provincial Natural Science Foundation for Distinguished Young Scientists (X. Liu); contract grant number: R4100133.

reactors[7–10] becomes a task with huge difficulty. Despite the simplification of the mechanical models, it still needs great efforts to fit the inner relationship between MI and some of the factors in the reaction.[10–12]

Another approach, the empirical model based on data and statistics, can estimate the hard-to-measure variable from the easy-to-measure variables without considering the complex chemical or physical reactions. Some industrial plants have used statistical methodologies to provide information for product and process design, monitoring and control,[13–18] and some researchers have also obtained easy empirical models for PP MI prediction through various methods. Han[19] used three different approaches, supported vector machines (SVM), partial least squares, and artificial neural networks (NN) for MI estimation of PP process. Shi[20,21] developed soft-sensor models for MI prediction based on weighted least squares support vector machines and independent component analysis, multiscale analysis, and radial basis function (RBF). NN have been widely applied to develop data-based model and control dynamic processes because of their extremely powerful adaptive capabilities in response to nonlinear behaviors.[22,23] Thus, Zhang[24] sequentially trained a set of NN, based on which the novel bootstrap aggregated NN are formed. Moreover, with the model developed by the aggregated NN, quite a good performance in the inferential estimation of the polymer MI in an industrial plant is obtained. These works have provided very good predictions, but greater performance and better universality of the estimation model are still necessary in academic and industrial community.

Recently, the artificial NN, especially the RBF NN, are widely used to develop empirical models for kinds of industry processes because of its powerful capacity in fitting the complex nonlinear relationships in the industry processes.[18,20,21] It is also quite a good choice to develop the empirical model for the PP process with RBF NN; therefore, the networks used to develop the MI prediction models below are all RBF NN. However, there are several points in implementing the idea, and the first one is the training of the RBF NN. The parameters of the network, such as linking weights between layers, biases and centers of hidden nodes, are the determining factors of the fitting capacity of the network. Because of the application of artificial intelligent searching algorithms, many optimization problems are solved[18,25–27] and the optimization of the RBF network structure can also be fulfilled with these algorithms. Here, a novel ant colony optimization (ACO) algorithm is proposed and used to optimize the RBF NN parameters; hence, the single NN model is obtained.

Another issue is that developing the model for PP process with a single RBF NN is not quite flexible, as the training of the network can easily be overfitted and make the model lack of generalization capacity. As an alternative and a more reliable approach, aggregated networks decrease the nonstability caused by the individual network and many engineering problems take this approach as a powerful solution.[28,29] Thus, developing the aggregated NN model with aggregated networks constructed by a group of different networks can further stabilize and improve the performance of the model.

To enhance the performance of aggregated NN model, a strategy for training the individual networks effectively and combining these networks reasonably is the most important. A sequential training method of developing aggregated NN model is presented here, and in this method, individual networks are sequentially trained to be as different from each other as possible. The first network is trained to minimize its prediction error, whereas the rest networks are trained to minimize the prediction error and the correlation with each other at the same time. To combine the networks more reasonable, an adaptive method to differ the importance of every network is adopted. Every time when a network is added, the combinational weight of every network in the aggregated networks is adjusted adaptively, letting the network with better prediction performance make more contribution to the aggregated NN model. According to this idea, the adaptive aggregated NN model is finally developed.

Based on the approaches above, three different models are developed. The first one is the single NN model formed by a RBF NN trained with the novel ACO algorithm. Then the aggregated NN model, where every network is trained with the novel ACO algorithm and the combinational weight of each network is the same, is obtained. Finally, the adaptive aggregated NN model is proposed, with individual networks trained by the novel ACO algorithm but the combinational weight of each network adaptively adjusted. The performance of the proposed models is illustrated and evaluated with an actual PP process, where the evaluation and research of the model presented in this article is carried out. The results obtained are then discussed and concluding remarks about the design are presented.

ADAPTIVELY AGGREGATED NETWORKS

RBF NN and a novel ACO algorithm

RBF NN

The RBF NN has satisfying global approximation and convergence property, which makes it the primary choice for the product quality prediction of the complex and correlated PP process.[30–33] It is a typical feed-forward network with a structure of three layers: the input layer, the hidden layer, and the output layer. The input layer collects the input information and formulates the input vector x. The hidden layer is composed by L hidden nodes, which apply nonlinear transformations to the input vector. The output layer gives the final responses. The RBF NN can be considered as a mapping in Euclidean space: $T : R^r \to R^s$. Let $x^p \in R^r$ be the input vector, and $c^i \in R^r (i = 1, 2, \cdots, k)$ be the center. The output is formed by a linear combination of the hidden layer responses, given by

$$y_j(x^p) = \sum_{i=1}^{k} w_{ji}\Phi_i(\|x^p - c^i\|), j = 1, 2, \cdots, s \quad (1)$$

where $\|\cdot\|$ is the Euclidean distance, k is the number of the hidden layer nodes, $\Phi_i(\cdot)$ is the hidden layer node response, w_{ji} is the output weight, x^p is the input vector, y_j is the output of jth output node, s is the number of the output nodes. In the current model, the hidden layer node uses the Gaussian activation function to make a response, that is

$$\Phi_i(\|x^p - c^i\|) = \exp\left(-\frac{(\|x^p - c^i\|)^2}{2\sigma_i}\right), i = 1, 2, \cdots, k \quad (2)$$

where c^2 and σ_i are the center and the width of the ith node in the hidden layer, respectively. They determine the receptive field around the node.

Novel ACO algorithm

ACO algorithm is a heuristic searching technique, which is inspired by the foraging mechanism of a real ant system and especially by the ability of the ants to figure out the shortest path between their nests and the food source.[34] Since its first mathematically submission by Dorigo in early 1990s, ACO algorithm has been widely used in various optimization problems but most discrete. ACO requires the problem be presented in discrete form, while the approach will largely influence the performance of the final prediction model because the linking weights in the networks are continuous real numbers. Thus, a novel ACO aimed at solving the continuous optimization problem here is proposed.

In the novel ACO algorithm, the solution of optimization problem is expressed by one node, which is a vector. The nodes represent food sources with certain pheromone concentrations which are related to the quantity of food in these sources. The objective of the ants is to find out the food source with most quantity of food. An ant starts searching with choosing a food source by a pheromone concentration-related probability, and tries to search around the food source it has chosen to find a more attractive food source. If a better food source, which contains more food than the original source does, is obtained, the ant gets excited and releases more pheromone according to the quantity of the food found. From the mathematic point of view, food source is a solution to the optimizing problem, and the quantity of food in the source is the fitness of the solution. Ants tend to search around good food sources to find out better sources, and it means that the algorithm is inclined to look for better solutions of the problem around the existing solutions with high fitness. It is the mechanism of the "local search" of the ants, which will be introduces later.

Sometimes, the food quantity in some sources is small and, at the same time, starting from these points, ants cannot find out better food sources than the original ones. Then these food sources should be replaced with the others, which are potential to have wonderful food sources around. To form the new food sources, the replaced ones are used to make combinations, where the ideas of mutation and cross basically used in GA algorithm are used.[35] The idea above is the "global search" of the ants, and the details will be explained subsequently.

The procedure of the novel ACO algorithm is as following:

Step 1: Prepare for the algorithm:

1.1. Initialize the searching space by giving the total number of ants m and a series of initial food sources $S = (s_1, s_2, \cdots, s_n)$. A food source $s_i = (x_1, x_2, \cdots, x_D)(i = 1, 2, \cdots, n)$ represents a solution to the D-dimension continuous optimizing problem, and n is the total number of starting points that ants will search around.

1.2. Calculate and record the fitness of $s_i(i = 1, 2, \cdots, n)$ as $F_i(i = 1, 2, \cdots, n)$.

1.3. Initialize the sequence number of iteration $k = 1$.

1.4. Specify the parameters of global search: R_1 and R_2. Here, R_1 is the number of sources that should be replaced by the new ones created with mutation operation, and R_2 is the number of sources that should be replaced by the new ones created with cross operation.

Step 2: Initialize the sequence number of ant $j = 1$ and do the local search:

2.1. Calculate the probability for choosing the source $s_i(i = 1, 2, \cdots, n)$ by:

$$P_i(k) = \frac{F_i}{\sum_{i=1}^{n} F_i}(i = 1, 2, \cdots, n) \quad (3)$$

2.2. Choose a s_{chosen} by following the roulette rules for ant j. Meanwhile, make sure that every source could not be chosen more than once during the kth iteration.

2.3. Create a distance with a direction $\text{del} = (d_1, d_2, \cdots, d_D)$ that ant j will walk and a new source s_{new} is obtained by:

$$S_{\text{new}} = S_{\text{chosen}} + \text{del} \tag{4}$$

Calculate the fitness of s_{new} and save as F_{new}, if $F_{\text{new}} > F_{\text{chosen}}$, s_{new} will be accepted and replace s_{chosen} in the set S. F_{chosen} in the set $F_i (i = 1, 2, \cdots, n)$ is also updated. Else, if $F_{\text{new}} \leq F_{\text{chosen}}$, do nothing.

2.4. $j = j + 1$; if $j > m$, go to Step 3; else go back to 2.1.

Step 3: In Step 2, local search has updated the set S, as well as $F_i (i = 1, 2, \cdots, n)$, and here the global search is processed:

3.1. Choose the sources to be replaced by following the roulette rules. The probability for source $s_i (i = 1, 2, \cdots, n)$ to be chosen is given by:

$$Q_i(k) = \frac{1/F_i}{\sum_{i=1}^{n} 1/F_i} (i = 1, 2, \cdots, n) \tag{5}$$

The total number of sources to be chosen is $R_1 + R_2$.

3.2. For the first R_1 sources to be replaced with mutation operation, let each initial source s_{old} shift a random distance in a random direction $\text{del} = (d_1, d_2, \cdots, d_D)$ to form a new source s_{new}:

$$S_{\text{new}} = S_{\text{old}} + \text{del} \tag{6}$$

Replace the chosen R_1 sources with the new ones.

3.3. For the left R_2 sources to be replaced with cross operation, let the initial source s_{old} cross with a random source s_{random} in S to get a new one s_{new}:

$$S_{\text{new}} = p.S_{\text{old}} + (1 - p) \cdot S_{\text{random}} \tag{7}$$

where, p is a probability parameter that can be adjusted.

Replace the chosen R_2 sources with the new ones.

Step 4: $k = k + 1$; if $k > \text{iter}_{\max}$, go to Step 5; else go back to Step 2.

Step 5: Take the source s_{best} with best fitness F_{best} to be the final solution of the optimizing problem.

When training an individual RBF NN based on the training dataset, best linking weights in the network are the most important parameters needed to be optimized, where the proposed novel ACO algorithm is applied.

Sequential training of the aggregated networks

A single network model can be obtained from the above approach, but,to achieve better performance, aggregated NN model is developed, where the individual networks are trained sequentially[24] and of course by the novel ACO algorithm. The output of the aggregated NN model is given by:

$$F(x) = \sum_{i=1}^{M} w_i F_i(x) \tag{8}$$

where, F is the aggregated networks, F_i is the ith individual network, and w_i is the combinational weight of the ith individual network. Here, w_i is simply assigned to be $1/M$.

The first network added into the aggregated networks is trained by the novel ACO, with an objective function as shown in eq. (9), to minimize the prediction error.

$$J_1 = \frac{1}{N} \sum_{j=1}^{N} (F_1(x_j) - d(x_j))^2 \tag{9}$$

where, N is the number of the training data points, d is the desired model output, and x_j is the jth training data point.

After training the first network, individual networks are trained and added to the aggregated networks, sequentially. Every subsequent network is trained to minimize the prediction error and maximize the difference from the formal trained networks. For example, the training objective of the ith individual network is as follows:

$$J_i = \frac{1}{N} \sum_{j=1}^{N} (F_i(x_j) - d(x_j))^2 - \frac{\lambda}{N} \sum_{j=1}^{N} (F_i(x_j) - F(x_j))^2 \tag{10}$$

and

$$F(x) = \sum_{j=1}^{i-1} w_j F_j(x) \tag{11}$$

$$w_j = \frac{1}{i-1}, j = 1, 2, \cdots, i - 1. \tag{12}$$

The first term of the training objective function minimizes the prediction error of the individual network, whereas the second term minimizes the correlation between the ith network and the previously trained networks, and λ is the weight of the second term.

The linking weights in the subsequent network are the objects needed to be optimized with novel ACO algorithm. The training process of the aggregated NN model can be terminated when its prediction error no longer decreases after adding an optimized subnet, or the number of the networks in the aggregated networks is larger than a certain value.

Adaptively combination of the aggregated networks

Though the aggregated networks can avoid the poor performance caused by some overfitting individual networks, when a subnet with a wonderful prediction performance is added, its contribution, on the good aspect, to the aggregated NN model is limited due to the average value of its combinational weight, which is a great pity. Thus, an adaptive approach of combining the individual networks is presented here, to make full use of the individual networks with great performance and to decrease the influence resulted from the overfitting subnets at the same time.

The training of the first network is the same as mentioned in last chapter, taking an objective of minimizing the prediction error [as shown in eq. (13)] and using the novel ACO algorithm to optimize the linking weights in the network. After the training process, the combinational weight of the network is assigned to be $w_{c1} = 1$, as it is the only network.

$$J_1 = \frac{1}{N}\sum_{j=1}^{N}(F_1(x_j) - d(x_j))^2 \qquad (13)$$

Then train the subsequent networks sequentially and add them into the aggregated networks adaptively, step by step. The key of the adaptive combination lies in the distribution of the combinational weights of the individual networks. The weight of every individual network is determined by the prediction errors of itself and the other subnets previously trained. If it is added into the aggregated networks, the combinational weights of every single subnet will be adjusted according to the following equation:

$$w_{ck} = (1/e_k)/\left(\sum_{j=1}^{i} 1/e_j\right), k = 1, 2, \cdots, i. \qquad (14)$$

where, w_{ck} is the combinational weight of the kth subnet, e_k is the prediction error of the kth individual network as a single network model.

When training the ith subnet, the objective function is as follows:

$$J_i = \frac{1}{N}\sum_{j=1}^{N}(F_i(x_j) - d(x_j))^2 - \frac{\lambda}{N}\sum_{j=1}^{N}(F_i(x_j) - F(x_j))^2 \qquad (15)$$

$$F(x) = \sum_{j=1}^{i-1} w_{cj}F_j(x) \qquad (16)$$

However here, $w_{cj}(j = 1, 2, \cdots, i-1)$ are no longer average values, but have been assigned certain values adaptively when the $i - 1$th subnet was added into the aggregated networks.

After the training of the ith subnet, a test is carried out, to determine whether the subnet should be added into the aggregated networks. The test is to calculate the prediction error of the aggregated NN model on the training dataset and figure out if the prediction error is decreased due to the adding of the ith subnet. If so, it should be a new subnet of the aggregated networks and the combinational weight of every subnet is adaptively adjusted according to eq. (14). Otherwise, it is not taken as a subnet of the aggregated networks, and the training and adding new networks into the aggregated networks are stopped. The networks already trained and added into the aggregated networks will be kept and the output of the adaptive aggregated NN model can be obtained according to eq. (16), but the combinational weights have been adaptively adjusted every time when a new network is added into the aggregated networks.

The termination criteria is not changed, that is terminating the algorithm and returning the aggregated network model when the adding of a new subnet cannot decrease the prediction error of the adaptive aggregated NN model, or when the number of the subnets in the aggregated networks is larger than a specified value.

CASE STUDY

A propylene polymerization process, which is currently operated for commercial purposes in a real plant in China, is considered in this article, and Figure 1 depicts the schematic diagram of the process. The process consists of a chain of reactors in series, two continuous stirred tank reactors, and two fluidized-bed reactors. The fed to the reactor is comprised of propylene, hydrogen, and catalyst. These liquids and gases are reactants for the growing polymer particles and also the provider of the heat transfer media. In the first two reactors, the polymerization reaction takes place in a liquid phase, and in the third and fourth reactors, the reaction is completed in vapor phase to produce the powdered polymer products. The melt index (MI) of the PP, which determines the properties and quality of the product, depends on the catalyst properties, reactant composition, and reactor temperature, and so on. Hydrogen can also regulate the molecular weight of PP.

To develop a prediction model to estimate the MI from a group of easy-measured variables, a pool of process information formed by nine process variables (T, p, l, a, f_1, f_2, f_3, f_4, and f_5) in PP process introduced above, which influence the process most greatly according to experience and mechanism, have been chosen. T, p, l, and a are process temperature, pressure, level of liquid, and percentage of hydrogen in vapor phase respectively. f_1, f_2, f_3 represent flow rates of three streams of propylene into the reactors, and f_4 and f_5 are flow rates of catalyst

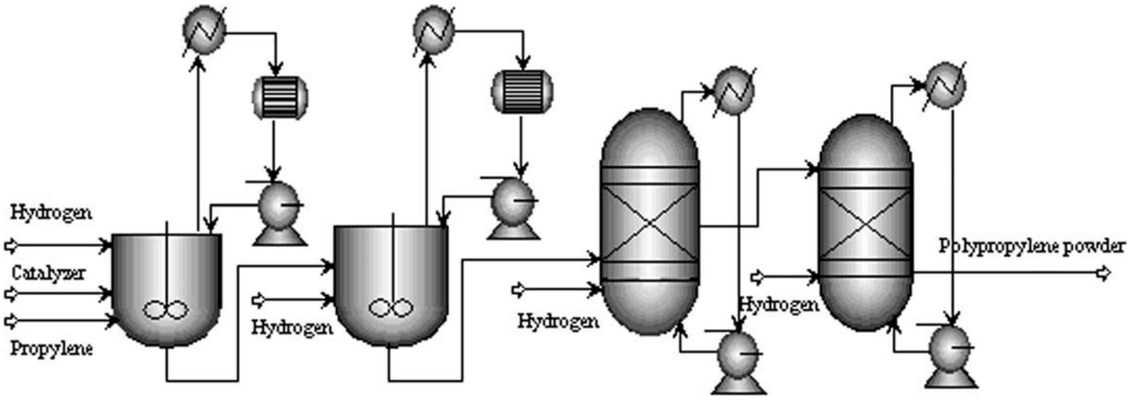

Figure 1 General scheme of propylene polymerization.

and aid-catalyst. A group of operational data has been taken from the discrete control system (DCS) historical log recorded in the real propylene polymerization plant, and the data are filtered at first, to discard abnormal situations and to improve the quality of prediction model. Normalization operation is also implemented to the input and output variables with respect to their maximum and minimum values. How to choose the most suitable training dataset from all the available process information is one of the most important issues before model learning. The approach to construct the training dataset here is dividing the data into training dataset, testing dataset, and generalization dataset according to the time series of the recorded data, then single NN model, aggregated NN model, and adaptive aggregated NN model are developed based on the training dataset. After the training of the models is finished, the testing dataset and generalization dataset are used to test the accuracy and the universality of the predictions of the models. There are 120 points in the training dataset and 30 points in the testing dataset, respectively, leaving the rest of the chosen data regarded as the generalization dataset. That the testing dataset and training dataset are from the same batch, whereas the generalization dataset is derived from another batch, is an important issue should be noted, because on this condition an accurate prediction on the testing dataset and the generalization dataset indicates the overall accuracy prediction and the pretty universality of the model, respectively. The RBF NN used here has five neurons and every neuron gets an activation function of "Gaussian function" as described in eq. (2). The single RBF NN used the novel ACO to training its weights, bias, and so on. To study the MI prediction accuracy of models statistically, the difference between the output of the models and the desired output (the analytic MI values from laboratory) is considered as the error and represented in several ways. In this article, the following measures are used for model evaluations: mean absolute error (MAE), mean relative error (MRE), root of mean square error (RMSE), and Theil's inequality coefficient (TIC). The calculation equations of these error indicators are shown as following:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{17}$$

$$\mathrm{MRE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{18}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{19}$$

$$\mathrm{TIC} = \frac{\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}}{\sqrt{\sum_{i=1}^{N}y_i^2} + \sqrt{\sum_{i=1}^{N}\hat{y}_i^2}} \tag{20}$$

where y_i and $\hat{y}_i$ denote the measured value and predicted result of MI, respectively.

The MAE, MRE, and RMSE confirm the prediction accuracy of the proposed methods. TIC indicates a good level of agreement between the proposed model and the studied process.[36]

The data listed in Table I show that the adaptive aggregated NN model has the best performance overall on the testing dataset. In a detailed study through error indicators mentioned above, single NN model gives an MAE of 0.0286, an MRE of 1.09%, a RMSE of 0.0341, and a TIC of 0.0064. For the aggregated NN model, composed by a group of RBF NNs optimized with the novel ACO algorithm, obtains an MAE of 0.0205, an MRE of 0.78%, a RMSE of 0.0252, and a TIC of 0.0047. The aggregated NN model has already obtained improved prediction accuracy than the single NN model. However, the adaptive aggregated NN model achieves even

TABLE I
Performance of the Models on the Testing Dataset

Models	MAE	MRE (%)	RMSE	TIC
Conventional single NN	0.0374	1.43	0.0458	0.0087
Single NN with novel ACO	0.0286	1.09	0.0341	0.0064
Aggregated NN with novel ACO	0.0205	0.78	0.0252	0.0047
Adaptive aggregated NN with novel ACO	0.0103	0.39	0.0121	0.0023

better performance. The MAE, MRE, RMSE, and TIC are 0.0103, 0.39%, 0.0121, and 0.0023, with percentage decreases of 49.76, 50.00, 51.98, and 51.06% compared to those of the aggregated NN model, respectively. These error indicators prove the adaptive aggregated NN model provides wonderful MI prediction accuracy for the propylene polymerization process. The performance of the single NN trained by conventional BP algorithm is also list in Table I, and it is clear that the performance of single NN model is better than Conventional single NN model, which demonstrates the advantage of the new ACO algorithm. The same thing happened in Table II.

A more distinct illustration in how better the adaptive aggregated NN model works than the single NN model and the aggregated NN model do on the testing dataset is shown in Figure 2. The curve marked with crosses is the real MI value obtained from analysis in laboratory, while the curve marked with circles is the MI value predicted by single NN model. The results predicted by aggregated NN model and adaptive aggregated NN model are depicted by the curves marked with squares and triangles respectively. Obviously, the adaptive aggregated NN model's result is best and nearly being the real MI value on every data point. The aggregated NN model's prediction is better than single NN model, but not as good as that of the adaptive aggregated NN model. The visual comparison proves the great prediction accuracy of the developed adaptive aggregated model.

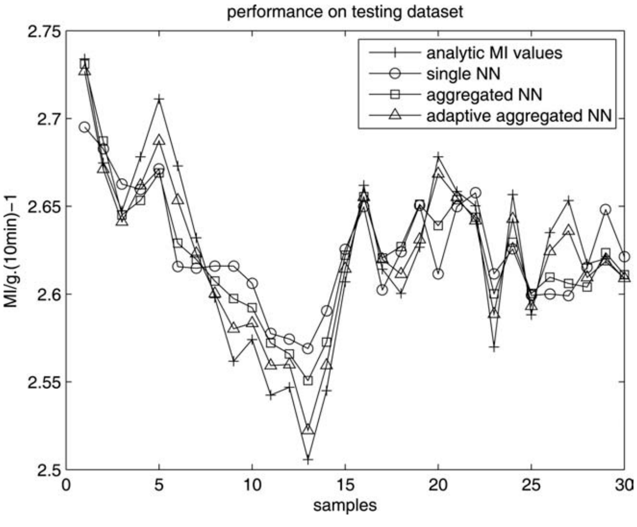

Figure 2 Performance of the models on the testing dataset.

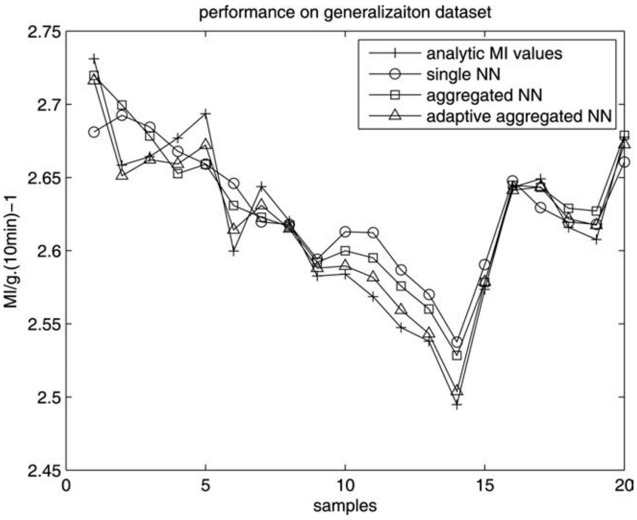

Figure 3 Performance of the models on the generalization dataset.

To see more about the universality of the proposed MI prediction models, models are further evaluated on the generalization dataset. An accurate prediction of MI on this dataset gives a strong support that the model owns good universality.

Table II lists the specific error indexes for single model, aggregated NN model, and adaptive aggregated NN model when they predict the MI on generalization dataset. The adaptive aggregated NN model defeats the other two models again, with a decrease of 51.43% in MRE from 0.70 to 0.34%, compared to the aggregated NN model. In addition, almost the same situation is occurred in terms of MAE, RMSE, and TIC. Aggregated NN model still wins over single NN model but looses to adaptively aggregated NN model very obviously.

Moreover, another visual comparison to study how the models work on generalization dataset is given in Figure 3, and it supports the conclusion even more strongly than the data in Table II does. The curves marked with circles, squares, and

TABLE II
Performance of the Models on the Generalization Dataset

Models	MAE	MRE (%)	RMSE	TIC
Conventional single NN	0.0381	1.46	0.0460	0.0088
Single NN with novel ACO	0.0245	0.94	0.0280	0.0055
Aggregated NN with novel ACO	0.0181	0.70	0.0205	0.0041
Adaptive aggregated NN with novel ACO	0.0089	0.34	0.0106	0.0020

TABLE III
The Prediction Results Comparison between Our Work and the Published Literature

Models	MAE	MRE (%)	RMSE	TIC
Ref. [19]	–	–	1.51	–
Ref. [20]	0.0635	2.49	0.0313	0.0138
This work	0.0089	0.34	0.0106	0.0020

triangles are still the MI values predicted by single NN model, aggregated NN model and adaptive aggregated NN model, respectively, whereas the analytic MI value curve is marked with crosses. Clearly, adaptive aggregated NN model gives a nearly real MI value prediction, more accurate than single NN model and aggregated NN model do. Thus, it is proved that the adaptive aggregated NN model holds excellent universality in MI prediction both statistically and graphically.

Table III gives the comparison of prediction results between our work and published literatures. The advantage of our work can be revealed clearly, as the prediction errors given in the work are much smaller than those of published works. It supports our research work again.

As complement of this work, Table IV which includes the list of parameters of the testing dataset to predict MIs of PPs synthesized by various conditions is also given.

CONCLUSIONS

Based on the RBF NN and a novel ACO algorithm aimed at continuous linking weights optimization of the network, several estimating models for the PP MI are developed. The first one is a single NN model, using only one RBF NN and trained by the novel ACO. It works pretty well and provides quite an accurate prediction which can be seen from the error indicators. Considering the over-fitting problem that single NN often faces with, several RBF NN trained sequentially are combined to construct an aggregated NN model. These individual networks are trained to minimize the prediction error and minimize the correlation with each other, and because of the training objective, the aggregated NN model achieves a much better prediction. In terms of MRE, the aggregated NN model's prediction error on testing dataset and generalization dataset decrease a percentage of 28.44 and 25.53%, respectively, compared to that of single NN model. As aggregated NN model combines every network with the same combinational weight, the adding of some

TABLE IV
Parameters of the Testing Dataset to Predict MIs

	a	f_4	f_5	f_1	f_2	f_3	l	p	T
1	0.351	48.33	24.01	11.44	3712.0	4005.0	41.14	29.672	69.95
2	0.277	51.61	24.176	10.11	3741.0	3996.0	40.92	29.637	70.0
3	0.248	51.69	24.18	9.97	3807.0	4000.0	40.64	29.613	69.96
4	0.277	55.02	24.259	10.25	3799.0	3999.0	40.48	29.592	69.96
5	0.317	55.86	24.226	10.15	3811.0	4007.0	39.05	29.66	70.03
6	0.277	56.44	24.385	10.82	3800.0	4000.0	36.31	29.527	69.87
7	0.233	56.76	24.446	10.63	3800.0	3994.0	36.32	29.614	69.96
8	0.2	57.77	24.842	9.16	3800.0	3995.0	35.98	29.6	70.0
9	0.173	52.14	24.874	10.97	3801.0	3992.0	36.33	29.556	70.01
10	0.18	55.95	24.813	10.71	3800.0	4007.0	36.49	29.521	69.99
11	0.147	61.26	24.761	10.1	3800.0	4000.0	36.6	29.57	69.98
12	0.143	64.92	27.047	10.22	3800.0	3993.0	38.96	29.531	70.0
13	0.112	64.33	19.889	10.14	3800.0	3996.0	39.18	29.504	69.98
14	0.143	64.09	27.252	10.62	3799.0	3992.0	39.28	29.562	70.1
15	0.191	60.06	27.982	9.76	3800.0	2933.0	39.56	29.468	70.03
16	0.244	59.72	27.91	9.8	3800.0	2876.0	39.55	29.55	70.07
17	0.196	59.82	27.845	10.59	3799.0	2891.0	39.6	29.494	69.92
18	0.185	59.58	27.853	10.17	3800.0	2903.0	39.48	29.565	70.06
19	0.209	59.16	27.999	9.28	3799.0	2870.0	39.73	29.652	70.13
20	0.269	58.84	27.917	9.85	3763.0	3808.0	40.04	29.439	69.76
21	0.25	58.94	28.205	10.3	3768.0	3790.0	39.9	29.595	70.06
22	0.242	59.36	28.235	9.94	3798.0	3814.0	39.65	29.654	70.16
23	0.165	59.84	28.191	10.09	3796.0	3786.0	39.77	29.574	70.03
24	0.244	60.83	26.204	10.17	3800.0	3759.0	40.19	29.547	69.94
25	0.178	60.87	26.266	11.37	3800.0	3718.0	40.0	29.587	69.99
26	0.223	60.59	26.188	11.0	3798.0	3788.0	39.91	29.494	69.87
27	0.253	62.5	26.213	4.31	3802.0	3744.0	32.84	29.43	69.79
28	0.221	62.5	26.256	3.32	3800.0	3794.0	31.37	29.541	69.98
29	0.214	62.58	26.298	3.47	3799.0	3768.0	36.31	29.564	70.01
30	0.203	62.28	26.404	6.25	3801.0	3848.0	36.16	29.588	69.99

relatively poor performance networks will weaken the role of some good performance networks. Thus, the combinational weights should be adjusted adaptively, to maximize the contribution of good-performance to the aggregated NN model, and because of this idea, the adaptive aggregated NN model is developed. It obtains even smaller prediction error than the aggregated NN model does, with a decrease percentage of 50.00 and 51.43% in MRE on testing dataset and generalization dataset respectively, compared to that of aggregated NN model.

Making full use of several networks with minimized prediction error and least correlation with each other, the aggregated NN model makes an improvement in MI prediction over the single NN model. However, through enhancing the role of relatively good performance networks, the adaptive aggregated NN model steps further than the aggregated NN model. The research on the data from a real plant indicates that the proposed models provide prediction reliability and accuracy, especially the adaptive aggregated NN model, which is robust to some extent and supposed to have promising potential in industrial use.

References

1. Bafna, S. S.; Beall, A. M. J Appl Polym Sci 1997, 65, 277.
2. McAuley, K. B.; MacGregor, J. F. AIChE J 1991, 37, 825.
3. Ahmad, S.; Mohamed, A. H.; Farouq, S. M.; Navid, M. Chem Eng J 2010, 161, 240.
4. Lee, E. H.; Kim, T. Y.; Yeo, Y. K. Korean J Chem Eng 2008, 25, 613.
5. Palmer, G.; Demarquette, N. R. Polymer 2005, 46, 8169.
6. Achilias, D. S.; Verros, G. D. J Appl Polym Sci 2010, 116, 1142.
7. Kanellopoulos, V.; Tsiliopoulou, E.; Dompazis, G.; Touloupides, V.; Kiparissides, C. Ind Eng Chem Res 2007, 46, 1928.
8. Khare, N. P.; Lucas, B.; Seavey, K. C.; Liu, Y. A.; Sirohi, A.; Ramanathan, S.; Lingard, S.; Song, Y. H.; Chen, C. C. Ind Eng Chem Res 2004, 43, 884.
9. Lo, D. P.; Ray, W. H. Ind Eng Chem Res 2006, 45, 993.
10. Chen, Y.; Liu, X. G. Polymer 2005, 46, 9434.
11. Ghasemi, S. M.; Sadeghi, G. M. M. J Appl Polym Sci 2008, 108, 2988.
12. Nele, M.; Latado, A.; Pinto, J. C. Macromol Mater Eng 2006, 291, 272.
13. Ahmed, F.; Nazir, S.; Yeo, Y. K. Korean J Chem Eng 2009, 26, 14.
14. Sharmin, R.; Sundararaj, U.; Shah, S.; Griend, L. V.; Sun, Y. J. Chem Eng Sci 2006, 61, 6372.
15. Rallo, R.; Ferre-Gine, J.; Arenas, A.; Giralt, F. Comput Chem Eng 2002, 26, 1735.
16. Kaneko, H.; Arakawa, M.; Funatsu, K. AIChE J 2009, 55, 87.
17. Neumann, G. A.; Finkler, T. F.; Cardozo, N. S. M.; Secchi, A. R. Ind Eng Chem Res 2006, 45, 2651.
18. Sarimveis, H.; Alexandridis, A.; Mazarakis, S.; Bafas, G. Comput Chem Eng 2004, 28, 209.
19. Han, I. S.; Han, C. H.; Chung, C. B. J Appl Polym Sci 2005, 95, 967.
20. Shi, J.; Liu, X. G. J Appl Polym Sci 2006, 101, 285.
21. Shi, J.; Liu, X. G.; Sun, Y. X. Neurocomputing 2006, 70, 280.
22. Chaudhuri, B. B.; Bhattacharya, U. Neurocomputing 2000, 34, 11.
23. Hunt, K. J.; Sbarbaro, D.; Zbikowski, R.; Gawthrop, P. J. Automatica 1992, 28, 1083.
24. Zhang, J.; Jin, Q. B.; Xu, Y. M. Chem Eng Technol 2006, 29, 442.
25. Sanchez, V. D. Neurocomputing 2002, 42, 147.
26. Lee, C. M.; Ko, C. N. Neurocomputing 2009, 73, 449.
27. Saha, B.; Ghoshal, A. K. Ind Eng Chem Res 2007, 46, 5485.
28. Ahmada, Z.; Zhang, J. Comput Chem Eng 2005, 30, 295.
29. Ahmada, Z.; Zhang, J. Neurocomputing 2009, 72, 1198.
30. Scarselli, F.; Tsoi, A. C. Neural Networks 1998, 11, 15.
31. Elanayar, S.; Shin, Y. C. IEEE Trans Neural Networks 1994, 5, 594.
32. Rivas, V. M.; Merelo, J. J.; Castillo, P. A.; Arenas, M. G.; Castellano, J. G. Inf Sci 2004, 165, 207.
33. Yu, D. L.; Gomm, J. B.; Williams, D. Control Eng Pract 1999, 7, 49.
34. Dorigo, M.; Blum, C. Theor Comp Sci 2005, 344, 243.
35. Lee, Z. J.; Su, S. F.; Chuang, C. C.; Liu, K. H. Appl Soft Comput 2008, 8, 55.
36. Murray-Smith, D. J. Math Compt Model Dyn Syst 1998, 4, 5.