Adaptive Configuring of Radial Basis Function Network by Hybrid Particle Swarm Algorithm for QSAR Studies of Organic Compounds

Yan-Ping Zhou, Jian-Hui Jiang,* Wei-Qi Lin, Hong-Yan Zou, Hai-Long Wu, Guo-Li Shen, and Ru-Qin Yu*

State Key Laboratory of Chemo/Biosensing and Chemometrics, College of Chemistry and Chemical Engineering, Hunan University, Changsha 410082, P. R. China

Received May 29, 2006

The configuring of a radial basis function network (RBFN) consists of selecting the network parameters (centers and widths in RBF units and weights between the hidden and output layers) and network architecture. The issues of suboptimum and overfitting, however, often occur in RBFN configuring. This paper presented a hybrid particle swarm optimization (HPSO) algorithm to simultaneously search the optimal network structure and parameters involved in the RBFN (HPSORBFN) with an ellipsoidal Gaussian function as a basis function. The continuous version of PSO was used for parameter training, while the modified discrete PSO was employed to determine the appropriate network topology. The proposed HPSORBFN algorithm was applied to modeling the inhibitory activities of substituted bis[(acridine-4-carboxamide)propyl]methylamines to murine P388 leukemia cells and the bioactivities of COX-2 inhibitors. The results were compared with those obtained from RBFNs with the parameters optimized by continuous PSO and by conventionally RBFN training the algorithm for a fixed network topology, indicating that the HPSO was competent for RBFN configuring in that it converged quickly toward the optimal solution and avoided overfitting.

1. INTRODUCTION

The radial basis function network (RBFN) as an important modeling approach has been extensively applied in quantitative structure−activity relationship (QSAR) studies. The superiority of RBFN to other methods, such as partial-least-squares (PLS) regression, is due to its inherent potential to approximate various nonlinear relationships between the descriptors and bioactivities within sufficient accuracy.[1]

The configuring of a RBFN is commonly performed by first initializing the centers and widths using a K-means algorithm for a fixed network topology[2,3] and then training the weights by a regularization method or descent algorithms.[4] Though the K-means method can effectively allocate training samples into clusters, it only takes the inputs into account and pays no attention to the outputs. Consequently, subsequent optimization of the network weights by a regularization method may well yield underfitting and suboptimum values. The descent algorithms offer the possibility of improving the centers and widths. They, however, suffer from some well-known drawbacks and limitations, for example, premature convergence to suboptima and inclination to overfitting. Therefore, the RBFN configured by conventional approaches is exposed to a high risk of overfitting and local optima. More importantly, the RBFN topology, that is, the number of hidden nodes, is generally predefined. The network topology is one of the most crucial factors for RBFNs to effectively solve problems. Oversimplified network architecture might hamper the convergence of the network, while excessive hidden nodes would induce overfitting and, hence, poor generalization. Several algorithms are used for network structure determination by stepwise growing or pruning of the number of hidden nodes.[5−7] A main drawback of such methods is that the performance is strongly dependent on concrete operation of the network because the topology is not preliminarily fixed but evolved during tuning of the whole system. A genetic algorithm was also introduced for optimizing the hidden node number and network parameters of RBFN in terms of Akaike's information criterion.[8] This strategy, however, confines the centers as a subset of the training examples. Such a limited feasible region for the centers could not provide a sufficient flexible model to guarantee acceptable performance. Therefore, it is highly demanding to develop an approach that is capable of identifying adaptively an appropriate RBFN architecture and the network parameters.

The particle swarm optimization (PSO) method,[9−12] a new optimization technique, can also be employed for RBFN configuration, which is originated as a simulation of a simplified social system. Most PSO algorithms operate in a continuous or real-number space.[13,14] Recently, our group developed a modified discrete PSO algorithm that demonstrates fast convergence to the optimum for discrete optimization problems.[14−17] In this paper, a continuous version of PSO was implemented for the parameter training of RBFN, while the proper network architecture was handled by the modified discrete PSO. The two versions of PSO were jointly used for adaptive configuring of the RBFN via the simultaneous determination of an appropriate architecture and the network parameters. Furthermore, to increase the network flexibility and alleviate excessive variability in the input variables, ellipsoidal Gaussian functions were taken as the basis transform in RBFN instead of the commonly used standard Gaussian functions. The RBFN with the ellipsoidal Gaussian basis function could approximate a complex relationship without the need of using a large number of basis

* Corresponding author tel.: +86-731-8822577; fax: +86-731-8822782; e-mail: rqyu@hnu.cn (R.-Q.Y.), jianhuijiang@hnu.cn (J.-H.J.).

10.1021/ci600218d CCC: $33.50 © 2006 American Chemical Society
Published on Web 10/21/2006

functions and balance the effect of the input variables on different hidden nodes. This offered the possibility of reducing the risk of overfitting. On the basis of joint evaluation of the network complexity and the model error, an objective function was also formulated as a performance measure for the RBFN configuring.

The proposed algorithm was used for modeling the inhibitory activities of substituted bis[(acridine-4-carboxamide)propyl]methylamines to murine P388 leukemia cells and bioactivities of COX-2 inhibitors. The results revealed that the proposed procedure compared favorably with the conventional algorithm, enabled a rapid convergence to the global optimum, and had the ability to avoid overfitting, indicating that the proposed approach holds great promise in the adaptive configuring of RBFN.

2. THEORY

2.1. Radial Basis Function Network. The RBFN consists of three layers: the input layer, hidden layer, and output layer. The input layer has no calculation power and merely serves as an input distributor to the hidden layer. Each hidden node takes a basis function as a nonlinear transfer function to operate on the input data. The operation of the output layer is a linear combination of the RBF units according to the following expression:

$$y_k(\mathbf{x}) = \sum_j^{n_h} w_{kj}\, \phi_j(\mathbf{x}) + w_k \tag{1}$$

where y_k is the kth output unit for the input vector $\mathbf{x}$, n_h is the number of RBF units, w_{kj} is the weight between the kth output and the jth hidden nodes, ϕ_j is the notation for the output of the jth RBF unit, and w_k is the bias.

The most commonly used RBF is the Gaussian function characterized by a center and a width. In the present study, the following ellipsoidal Gaussian function was taken as the basis function:

$$\phi_j(\mathbf{x}) = \exp[-(\mathbf{x} - \mathbf{c}_j)^{\mathrm{T}}\mathbf{Q}_j^{-1}(\mathbf{x} - \mathbf{c}_j)] \tag{2}$$

where $\mathbf{Q}_j$ is a diagonal width matrix with the nth element referring to the width for the nth input variable ($n = 1, 2, ..., N$) at the jth RBF unit and $\mathbf{c}_j$ represents the center of the jth RBF unit, which is a vector with the same dimension as the input vector $\mathbf{x}$. The major difference between the ellipsoidal Gaussian function and the standard one lies in the width matrix in which the ellipsoidal version allows varying widths for different input variables, while the standard version has an identical width for all input variables. Such an ellipsoidal Gaussian basis function offers the flexibility to account for the variability in the spread of different input variables, thus enabling the ellipsoidal data clusters to be modeled with very few RBF units. In contrast, with the standard Gaussian basis function, these ellipsoidal data clusters might be split into multiple spheres, leading to an expanded complexity in the RBFN model and an increased risk of overfitting.

The configuring of a RBFN comprises the optimization of the network parameters (centers, widths, and weights) and the architecture. In this paper, a hybrid PSO was implemented for adaptively configuring the RBFN. The continuous PSO and the modified discrete PSO were jointly utilized to seek for the parameters and network architecture, respectively.

2.2. Particle Swarm Optimization. PSO, originally developed by Eberhart and Kennedy, is a stochastic global optimization method simulating the social behavior of bird flocking. It explores the problem space by a population of particles, each standing for a single solution. In PSO, each particle flies over the problem space with a velocity guiding the flying of the particle, keeping track of the best solutions encountered so far.

The first step of PSO is to randomly initialize the position and velocity of each particle in the swarm by dispersing them uniformly across the search space. The ith particle and its corresponding velocity, that is, the rate of the position change for the ith particle, are represented as $\mathbf{x}_i = (\mathbf{x}_{i1}, \mathbf{x}_{i2}, ..., \mathbf{x}_{iD})$ and $\mathbf{v}_i = (\mathbf{v}_{i1},\mathbf{v}_{i2}, ..., \mathbf{v}_{iD})$, respectively. In every cycle, updating each particle is realized by following the personal (individual) best position and the global best position. The former refers to the best previous position of the ith particle yielding the best fitness value, represented as $\mathbf{p}_i = (\mathbf{p}_{i1}, \mathbf{p}_{i2}, ..., \mathbf{p}_{iD})$, while the latter is the best particle among all of the particles in the population, represented as $\mathbf{p}_g = (\mathbf{p}_{g1}, \mathbf{p}_{g2}, ..., \mathbf{p}_{gD})$. Once the above-mentioned two best values have been found, the particle updates its velocity and position in terms of the following two equations:

$$v_{id}(\text{new}) = v_{id}(\text{old}) + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \tag{3}$$

$$x_{id}(\text{new}) = x_{id}(\text{old}) + \mu v_{id}(\text{new}) \tag{4}$$

where c_1 and c_2 are two positive constants named learning factors, and from experience, both constants take the integer value 2; r_1 and r_2 are random numbers in the interval (0,1). In eq 4, μ, a random number uniformly distributed in (0,1), is the restriction factor to determine velocity weight. The particle's velocity is renovated by employing eq 3 according to its previous velocity and the distances of its current position from its personal best position and the global best position. Then, the particle flies to a new position according to eq 4. Such an adjustment of the particle's movement through the space leads it to search around the two best positions. The algorithm is ceased with the minimum error criterion or the user-defined limit of the iteration number reached.

2.3. Modified Discrete Particle Swarm Optimization. The modified discrete PSO was proposed[15] on the basis of the same information-sharing mechanism as the continuous PSO, that is, updating the particle by following the personal and global best positions. As to a binary discrete problem, a particle moves in a search space confined to 0 or 1 on each bit. The particle renovation refers to variation of a bit that should be in either state 0 or 1. The velocity v_{id}, a random number in the range of (0, 1), represents the probability of a bit taking value 1 or 0. The resultant variation in position is specified by the following rule:

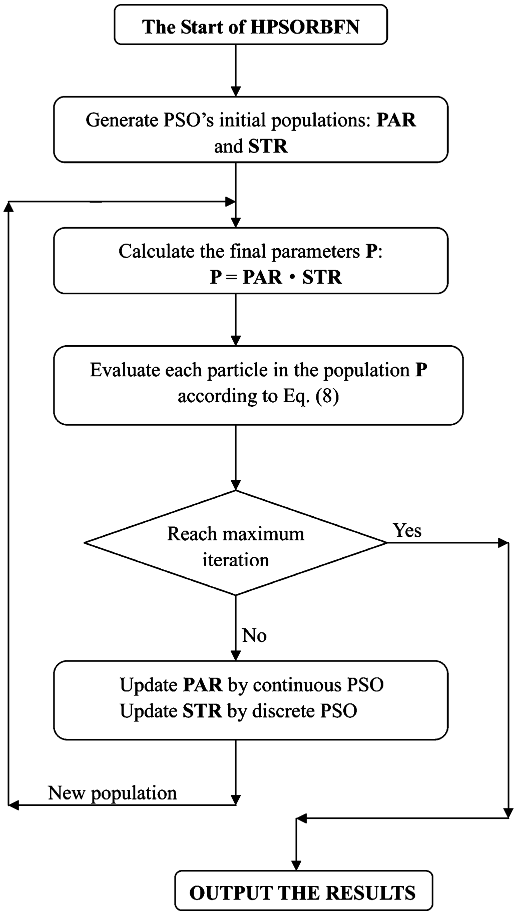
$$\text{if } 0 < v_{id} \leq a, \text{ then } x_{id}(\text{new}) = x_{id}(\text{old}) \tag{5}$$

$$\text{if } a < v_{id} \leq (a + 1)/2, \text{ then } x_{id}(\text{new}) = p_{id} \tag{6}$$

$$\text{if } (a + 1)/2 < v_{id} \leq 1, \text{ then } x_{id}(\text{new}) = p_{gd} \tag{7}$$

where a is a random number ranging from 0 to 1, named

Scheme 1. Flowchart of the HPSORBFN Scheme


The Start of HPSORBFN
Generate PSO's initial populations: PAR and STR
Calculate the final parameters P:
P = PAR • STR
Evaluate each particle in the population P according to Eq. (8)
Reach maximum iteration
Yes
No
Update PAR by continuous PSO
Update STR by discrete PSO
New population
OUTPUT THE RESULTS

the static probability, which is started with a value of 0.5 and decreases to 0.33 when the iteration terminates. A total of 10% of the particles are forced to fly randomly, not following the two best positions to enhance the potential of the discrete PSO technique to circumvent local optima.

2.4. Adaptive Configuring of RBFN by Hybrid PSO (HPSORBFN). In the present study, a hybrid PSO was introduced to adaptively configure RBFN, in which the continuous PSO and modified discrete PSO were implemented for optimizing network parameters and architecture, respectively. The two tasks were performed simultaneously. In continuous PSO, each particle was encoded as a real string, the bits of which were real numbers denoting centers, widths, and weights for a RBFN with sufficient hidden units. In the modified discrete PSO, each particle was encoded as a string of binary bits associated with the centers, widths, and weights. Actually, the training of the network architecture refers to the pruning of the hidden nodes. Because a hidden unit can be easily pruned by setting the weight between this hidden and output node as zero, the bits of each particle in the modified discrete PSO associated with the weights were set to either zero or one. A bit of zero in a particle implies that the corresponding hidden unit was excluded in the final RBFN model; that is, the corresponding weight is useless. A bit of one in a particle means that the corresponding hidden node was included in the RBFN computation. The bits of each particle corresponding to the centers and widths in the modified discrete PSO are all set to one for reducing the computation burden, because these bits have no direct effect on the optimization of the topology. The adaptive configuring of RBFN by hybrid PSO is described as follows:

Step 1. Generate randomly all of the initial strings **PAR** and **STR** in two versions of PSO with a population of proper size. **PAR**s are encoded as real number-coded strings, representing the parameter values involved in the RBFN. **STR**s are binary strings corresponding to **PAR**s encoded as described above.

Step 2. Calculate the final parameters **P** involved in the RBFN computation using the Hadamard product of **PAR** and **STR**; that is, if $\mathbf{PAR} = (\mathrm{par}_{ij})_{mn}$ and $\mathbf{STR} = (\mathrm{str}_{ij})_{mn}$, then Hadamard multiplication gives a product $\mathbf{PAR{\bullet}STR} = (\mathrm{par}_{ij} \times \mathrm{str}_{ij})_{mn}$.

Step 3. Calculate the fitness function (vide infra) of the RBFN associated with each particle of the population **P**. If the current iteration number reaches the predefined maximum iteration number, the training is ceased with the results output; otherwise, go to the next step.

Step 4. Update the **PAR** population in terms of the fitness function by employing the continuous PSO. Update the bits in the **STR** population associated with the weights according to the modified discrete PSO, and keep the bits corresponding to the centers and widths fixed at one. Return to step 2 to run the next iteration. The flowchart of HPSORBFN is given in Scheme 1.

2.5. Fitness Function. In this paper, a predefined fitness function is used to evaluate the performance of each particle in HPSORBFN, whose minimization would yield an optimum RBFN. Configuring RBFN by hybrid PSO should necessarily search the optimum network architecture and optimum set of the parameters and bias. On the basis of these requirements, an objective function is introduced as follows:

$$f = \mathrm{RMSE}[1 + \lambda(m/n_c)] \tag{8}$$

where RMSE (root-mean-squared error) represents the network accuracy, the term m/n_c stands for the network complexity, in which m is the number of weights that are not equal to 0, that is, the number of weights involved in the RBFN calculation, and n_c is the total number of weights, and λ is the weighting coefficient controlling the tradeoff between the accuracy and the network complexity. The larger the value of the parameter λ, the simpler would be the network structure. This may prevent convergence of the network, and the algorithm may converge to a poor solution. On the contrary, a small value of λ is beneficial to the algorithm to converge more easily but may possibly lead to overfitting. From experience, λ is set to 0.1 to keep a balance between the accuracy and complexity of the network.

3. DATA SETS

3.1. Data Set 1. To evaluate the performance of the proposed algorithm, 40 substituted bis[(acridine-4-carboxamide)propyl]methylamines with the corresponding inhibitory activities to murine P388 leukemia cells were used as a data set.[18] The molecular basic structure of bis(acridine-4-carboxamides) is shown in Figure 1. The bioactivities of the compounds are measured as IC_{50}, the concentration of the drug required to reduce cell numbers to 50% of the control cultures in leukemia cells. The detailed structures and the corresponding bioactivities of the compounds are listed in Table 1 of the Supporting Information. The data set was

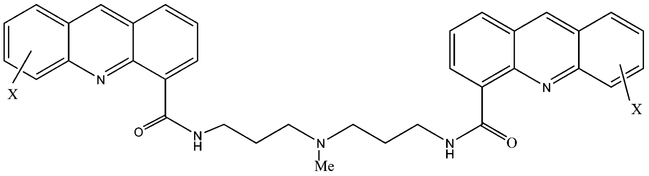


Figure 1. Molecular structure of bis(acridine-4-carboxamides).

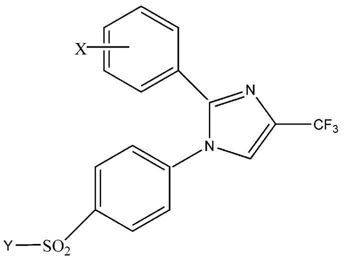


Figure 2. Parent structure of 1,2-diarylimidazoles as COX-2 inhibitors.

randomly divided into two subsets, the training set of 30 compounds and the test set of 10 compounds. In addition to the molecular descriptors listed in ref 18, over 70 descriptors calculated by the Cerius^2 3.5 software system on a Silicon Graphics R3000 workstation were also utilized to describe these compounds. The calculated descriptors include topological, electronic, spatial, structural, and thermodynamic descriptors. As to each compound, 14 parameters were used for multivariate modeling, including density, two dipole moments (Dipole-Y and Dipole-Z),[19] four Jurs descriptors (Jurs-DPSA-2, Jurs-WNSA-2, Jurs-RPCG, and Jurs-RPSA),[20] three shadow indices (shadow-Xlength, shadow-Ylength, and shadow-Zlength), the desolvation free energy of water (Fh2o),[21] and three verloop's sterimol parameters (B5$_5$, B5$_7$, and L$_6$). B5$_5$ and B5$_7$ define the maximum width of the substituents. L$_6$ represents the length substituent moiety.

3.2. Data Set 2. A total of 72 1,2-diarylimidazoles, as COX-2 inhibitors, with their corresponding bioactivities were used as another data set to further check the validity of the newly developed algorithm. These data were taken from a recent review contributed by Garg et al.[22] Khanna et al.[23] synthesized and evaluated their inhibitory activities to the human COX-2 enzyme. The molecular parent structure of COX-2 inhibitors is presented in Figure 2. The activity was expressed as IC$_{50}$, the molar concentration of the compound causing 50% inhibition of the enzyme. Table 2 of the Supporting Information lists the detailed structures of the compounds and the corresponding bioactivities. We stochastically divided the data set into a training set of 49 compounds and a test set of 23 compounds. When the Cerius^2 3.5 software system was used on a Silicon Graphics R3000 workstation, over 70 descriptors representing the chemical structure were calculated as the original variables. These descriptors cover different aspects of the molecular structure, including electronic, spatial, structural, and thermodynamic descriptors and E-state indices. On the other hand, three descriptors used by Garg et al.[22] were also introduced into the family of variables. Each sample is described by the following eight variables, including Jurs-FPSA-3,[20] the density, two shadow indices (shadow-Xlength and shadow-Ylength), the number of rotatable bonds (Rotlbonds), two indicator variables (I$_Y$ and L$_{X,2}$), and McGowan's volume (Mgvol). Mgvol is the molar volume calculated by using the methods of McGowan.

The algorithms used in this study were written in Matlab 5.3 and run on a personal computer (Intel Pentium processor 4/2.66 GHz with 256 MB of RAM).

4. RESULTS AND DISCUSSION

4.1. Data Set 1. Generally, the performance of RBFNs is sensitive to the number of hidden nodes. Because the useless weights in HPSORBFN are automatically set as zero during architecture optimization, the increase of hidden neurons yields little effect on the network performance. In this study, even the RBF units were set as 50; the symptom of overfitting is not likely to occur under this condition. For this data set, we set the RBF units as 15, taking into account the fact that small networks could reduce the enormous search space in learning and are usually faster and cheaper to establish.

As a comparison, the PLS method was first applied to this data set with five latent variables. The statistical results of PLS are shown in Table 1. The model by PLS yielded RMSEs of 0.3483 and 0.5846 for the training set and test set, respectively. The correlation between the calculated and observed values of bis(acridine-4-carboxamides) is shown in Figure 3. As shown in Figure 3 and Table 1, the correlation was rather poor and the modeling error is quite high. The presence of nonlinearity in this data set was proven by a run test[24] that yielded a statistical value of -3.36, whose absolute value is larger than the critical value 1.96. It seems that PLS is inadequate for modeling bis(acridine-4-carboxamides) because of the presence of nonlinearity among this data set. To compare with HPSORBFN, parameters of RBFN trained by PSO (PSORBFN) and obtained by a conventional K-means and regularization method (KRRBFN) were also carried out for this data set. The correlation coefficients (R) for the training set obtained by PSORBFN and KRRBFN were 0.8626 and 0.8834, respectively. The models by PSORBFN and KRRBFN gave the correlation coefficients of 0.8380 and 0.8524 for the test set, respectively. The statistical results for PSORBFN and KRRBFN, together with those for PLS, are summarized in Table 1. A comparison of PLS with PSORBFN and KRRBFN indicates that better results are obtained by PSORBFN and KRRBFN.

To further improve the QSAR model, the HPSORBFN method was employed to model the bioactivities of a series of bis(acridine-4-carboxamides). In HPSORBFN, the objective of the modified discrete PSO is to find the appropriate network architecture while the role of the continuous PSO is to optimize the parameters involved in RBFN. The two tasks are performed simultaneously. Instead of a standard Gaussian basis function, an ellipsoidal Gaussian function acts as the basis function in RBF units. For this data set, the RBFN is used with 15 hidden nodes for 14 variables. In HPSORBFN, the population size of PSO is selected as 100. The correlation between the calculated and observed values of bioactivities is depicted in Figure 4. The statistical results by HPSORBFN are also listed in Table 1. The correlation coefficients of 0.8835 and 0.8942 were obtained by HPSORBFN respectively for the training set and test set. The correlation coefficient for the test set is comparable to that for the training set by HPSORBFN, indicating no sign of

Table 1. Results of QSAR Analysis of Bis[(acridine-4-carboxamide)propyl]methylamines Using HPSORBFN Compared with Those Obtained by PLS, PSORBFN, and KRRBFN

	R (correlation coefficient)				RMSE (root-mean-squared error)			
data set	method 1[a]	method 2[b]	method 3[c]	method 4[d]	method 1[a]	method 2[b]	method 3[c]	method 4[d]
training set	0.8549	0.8626	0.8834	0.8835	0.3483	0.3482	0.3147	0.3164
test set	0.6637	0.8380	0.8524	0.8942	0.5846	0.4259	0.4207	0.3496

[a] QSAR study by PLS. [b] QSAR study by PSORBFN. [c] QSAR study by KRRBFN. [d] QSAR study by HPSORBFN.

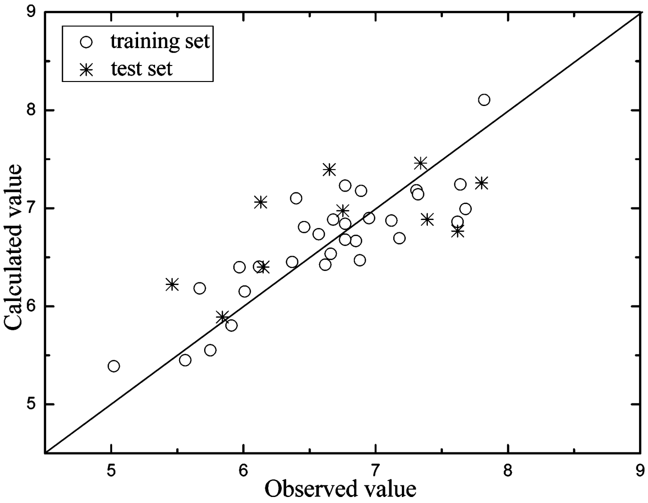


Figure 3. Calculated versus observed values of the bioactivities by PLS modeling with five latent variables for a series of bis(acridine-4-carboxamides).

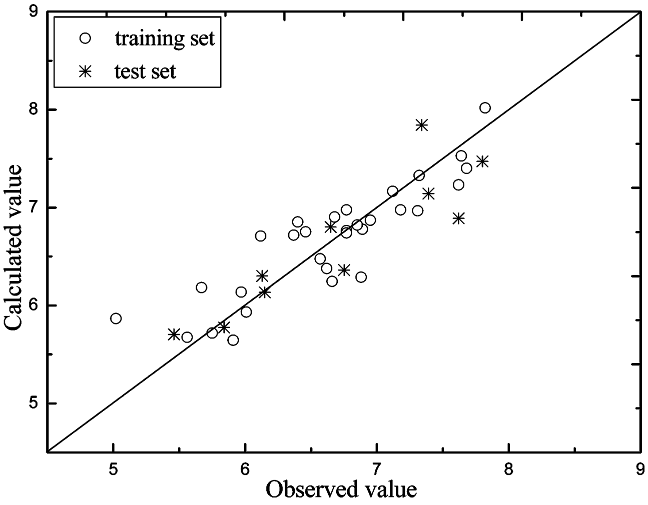


Figure 4. Calculated versus observed values of the bioactivities using the HPSORBFN algorithm for a series of bis(acridine-4-carboxamides).

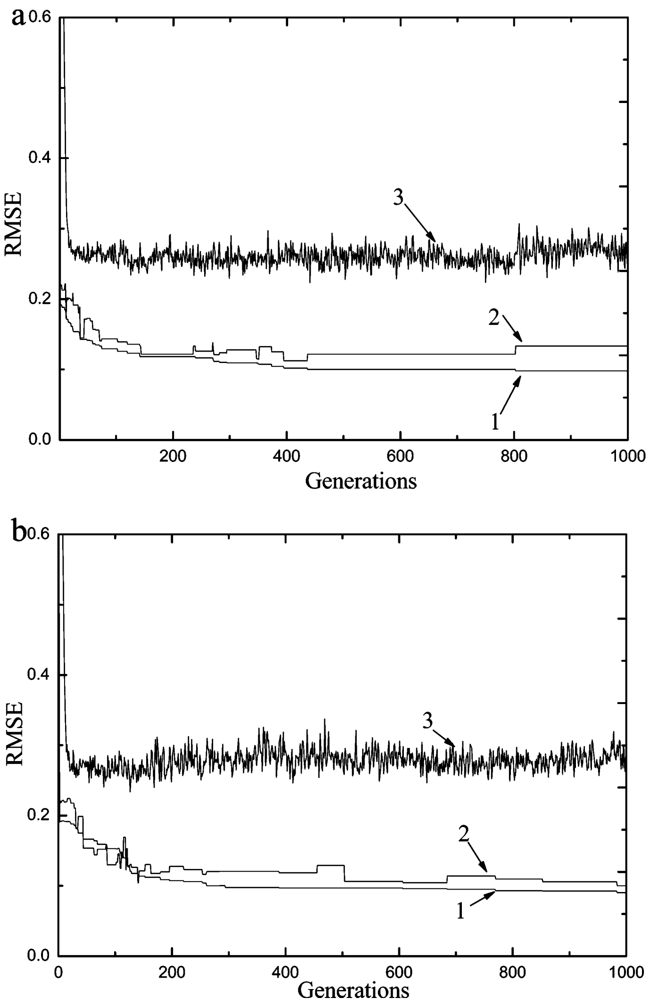


Figure 5. Convergence curves for PSORBFN (a) and HPSORBFN (b) for bis(acridine-4-carboxamides). Curve 1: RMSE for the training set gained with the most fitted member in each generation. Curve 2: RMSE for the test set calculated by this most fitted member for the training set. Curve 3: The average of the RMSEs of the training set obtained by all members of each generation of each algorithm.

overfitting in HPSORBFN. In Garg's work,[18] four out of a total of 40 compounds were identified as outliers and were not involved in the QSAR modeling. The model established by Garg offered a correlation coefficient of 0.8944, which was comparable to the ones obtained in the present study without outliers taken out. This indicates that better results are obtained in the present study. By using HPSORBFN, the RMSE for the test set was reduced from 0.4207 by KRRBFN and 0.4259 by PSORBFN to 0.3496, indicating the superior performance of HPSORBFN compared to KRRBFN and PSORBFN.

The convergence processes for PSORBFN and HPSORBFN can be examined in Figure 5a and b by plotting the RMSE versus the generation. The two algorithms were both stopped after 1000 cycles. Curve 1 is the convergence curve drawn with the RMSE for the training set gained with the most fitted member in each generation. Curve 2 represents the RMSE curve for the test set calculated by this most fitted member for the training set. Curve 3 refers to the average of the RMSEs of the training set obtained by all members of each generation for each algorithm. From Figure 5, one can discern that PSO converges to the best solution quickly. In the parameter training of RBFN by PSO, though the RMSE for the test set (curve 2, Figure 5a) obviously drops at the very beginning generations, the performance of the fittest member of each generation for the test set (curve

Table 2. Results of QSAR Analysis of COX-2 Inhibitors Using HPSORBFN Compared with Those Obtained by PSORBFN and KRRBFN

	R (correlation coefficient)			RMSE (root-mean-squared error)		
data set	method 1[a]	method 2[b]	method 3[c]	method 1[a]	method 2[b]	method 3[c]
training set	0.9287	0.9378	0.9382	0.3205	0.2962	0.3025
test set	0.8944	0.9032	0.9210	0.4015	0.3923	0.3228

[a] QSAR study by PSORBFN. [b] QSAR study by KRRBFN. [c] QSAR study by HPSORBFN.

2) improved very slowly compared to the convergence rate for the same member as evaluated by the training set (curve 1). With the reduction of RMSE for the training set (curve 1, Figure 5a), the RMSE for the test set tends to increase slightly, and no further improvement was observed. Such phenomena imply that overfitting occurs when only the parameters are optimized. Figure 5b depicts the convergence processes for HPSORBFN. By visual inspection of curve 1 in Figure 5b, one can discern that HPSORBFN is capable of converging to a satisfactory solution within 200 cycles. The variation trends of curve 2 and curve 1 appear to be parallel to each other for HPSORBFN and might finally turn out to be identical, indicating no sign of overfitting of the training set, as often occurred in the RBFN configuring. The time required to perform the proposed algorithm is only several minutes.

When comparing with PSORBFN and KRRBFN, HPSORBFN provides better performance. It shows that the two versions of PSO can jointly search the optimal RBFN architecture and parameters and the introduction of a hybrid PSO for the RBFN architecture optimization is very beneficial for overcoming the overfitting issue and improving the characteristic performance of the configured RBFN. This may benefit from the fact that HPSO cannot only hold population diversity but also maintain the best particle among all of the particles in the population. The way used to retain the diversity of populations is capable of preventing the issue leading to suboptima. According to eq 1, it can be concluded that the obtained estimate for the weights between the hidden layer and output layer for a complex network topology, that is, the regression coefficients for the linear model defined by eq 1, will have inflated variance, leading to overfitting of the RBFN model to the data errors. Therefore, the pruning of the hidden nodes, that is, the optimizing of the network topology, is equivalent to the feature selection in the general linear regression analysis of eq 1, offering a convenient strategy to combat overfitting. Moreover, the introduction of the ellipsoidal Gaussian function as the basis function is conducive to increasing the network flexibility and reducing the risk of overfitting so as to further improve the network performance.

To further assess the behavior of QSAR modeling, we divided randomly the whole data set into a training set and a test set five times. Each time, the model obtained on the basis of the training set was used to predict the bioactivities of compounds in the test set. The mean correlation coefficients for the training and test sets by HPSORBFN for data set 1 in five computations were 0.8901 and 0.8867, respectively. As compared to the correlation coefficients of 0.8835 and 0.8942 for the original training and test sets, it is clear that the model obtained by HPSORBFN is statistically stable. As to the PSORBFN training, the mean correlation coefficients for the training set and test one were 0.8684 and 0.8396, respectively. In KRRBFN learning, the mean correlation coefficients were 0.8716 and 0.8430 respectively for the training and test sets. These results revealed that HPSORBFN offered better generalization performance than KRRBFN and PSORBFN, and it was not a fortuitous choice of the training set that produced the good results.

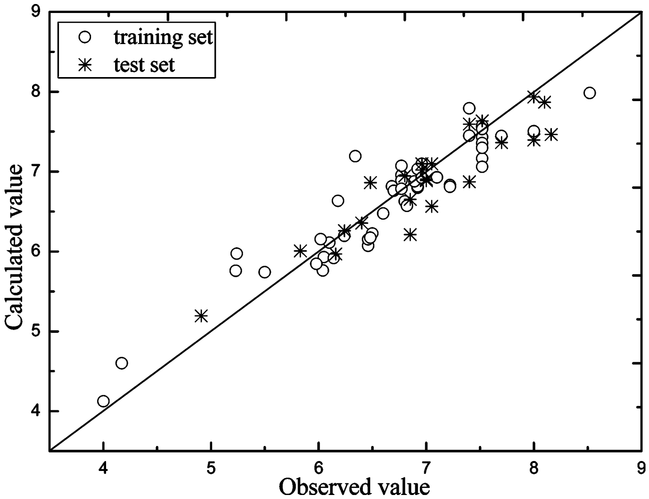


Figure 6. Calculated versus observed values of the bioactivities using the HPSORBFN algorithm for COX-2 inhibitors.

4.2. Data Set 2. For further checking of the performance of the proposed HPSORBFN algorithm, we applied this approach to modeling the bioactivities of COX-2 inhibitors. The PSORBFN and KRRBFN methods, as comparisons, were also investigated with the hidden neurons identified as 20. The statistical results obtained by the PSORBFN and KRRBFN methods are summarized in Table 2. The correlation coefficients for the training set by PSORBFN and KRRBFN were 0.9287 and 0.9378, respectively. The PSORBFN and KRRBFN methods gave correlation coefficients of 0.8944 and 0.9032 for the test set, respectively. The *R*s for the test set are much poorer than those for the training set by PSORBFN and KRRBFN. This seems a sign of overfitting to the training set.

As to this data set, we used the network with 20 RBF units for the selected eight variables in the HPSORBFN. The results gained by HPSORBFN, together with those obtained by PSORBFN and KRRBFN, are listed in Table 2. The correlation coefficients for the training set and test one obtained by HPSORBFN were 0.9382 and 0.9210, respectively, as compared to the correlation coefficient value of 0.9126 in Garg's review, indicating that better results for COX inhibitors were obtained in the present study. From Table 2, one can see that, by using HPSORBFN, the RMSE for the test set was reduced from 0.4015 by PSORBFN and 0.3923 by KRRBFN to 0.3228. The plot of calculated versus observed values of inhibitory activities of COX-2 inhibitors by HPSORBFN is shown in Figure 6.

The algorithm was stopped after 1000 cycles. Parts a and b of Figure 7 reflect the convergence processes for PSORBFN

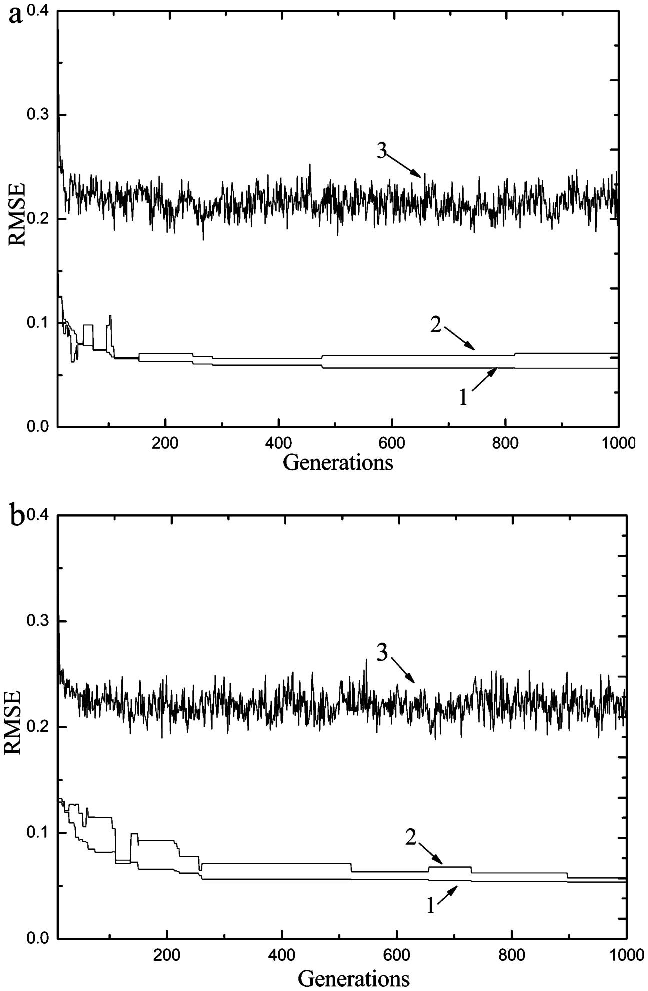


Figure 7. Convergence curves for PSORBFN (a) and HPSORBFN (b) for COX-2 inhibitors. Curve 1: RMSE for the training set gained with the most fitted member in each generation. Curve 2: RMSE for the test set calculated by this most fitted member for the training set. Curve 3: The average of the RMSEs of the training set obtained by all members of each generation of each algorithm.

and HPSORBFN, respectively. From Figure 7, one can see that the fitness value drops quickly in PSORBFN and HPSORBFN, confirming that PSO converges to the best solution quickly. In PSORBFN, with the decrease of RMSE for the training set (curve 1, Figure 7a), the RMSE for the test set turned to increase slightly, and no further improvements were observed. This means that overfitting for the training set occurs under this condition. The trends of variation of curves 2 and 1 in Figure 7b seem to be parallel to each other for HPSORBFN. It seems to be no sign of overfitting of the training set under this situation. The results with this data set revealed again that the introduction of network architecture optimization into the algorithm can offer substantially improved performance compared to PSORBFN and KRRBFN. Moreover, the network is more flexible and exhibits a stronger generalization ability by using the ellipsoidal Gaussian function as the basis function in RBF units.

For this data set, the assessment of the behavior of QSAR modeling was also investigated by dividing randomly the whole data set into training and test sets five times. The mean correlation coefficients obtained by HPSORBFN in five computations were 0.9354 and 0.9309, respectively, for the training and test sets. In KRRBFN training, the mean correlation coefficients for the training set and test one were 0.9367 and 0.9014, respectively. As to PSORBFN learning, the mean correlation coefficients for the training and test sets were 0.9182 and 0.8893, respectively. These results also showed the superior ability of HSORBFN to PSORBFN and KRRBFN in QSAR modeling.

The overfitting issue tended to occur in PSORBFN during the later period training for both data sets, while HPSORBFN continued to generate sound RBFN models throughout the configuring period. These results indicate that RBFN with the introduction of the HPSO pruning architecture is essential for overcoming overfitting and local optima. One can also observe that both PSORBFN and HPSORBFN provided a high convergence rate, indicating that the PSO algorithm ensures convergence to the global optima with relatively high efficiency.

5. CONCLUSION

The efficient procedure for the configuration of RBFN is to optimize the proper network topology and parameters simultaneously. In the present study, the continuous PSO was used for the parameter training of RBFN, and the modified discrete PSO was invoked to search the optimum RBFN architecture. The two versions of PSO were jointly used to simultaneously search the network topology and parameters. To increase the network flexibility and enhance the ability of RBFN to circumvent overfitting, the ellipsoidal Gaussian function was taken as the basis function in RBFN. A fitness function was formulated to identify the proper network topology and optimum parameter values. The performance of the algorithm was assessed using two QSAR data sets. Experimental results revealed that the hybrid PSO was effective for configuring RBFN, which converged quickly to the optimal solution and was capable of greatly improving the generalization ability of the model.

ACKNOWLEDGMENT

The work was financially supported by the National Natural Science Foundation of China (Grant Nos. 20375012, 20205005, 20105007, and 20435010).

Supporting Information Available: Detailed structural formulas of the compounds. This information is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Park, J.; Sandberg, I. W. Approximation and Radial-Basis-Function Networks. *Neural Comput.* **1993**, *5*, 305−316.

(2) Hwang, Y. S.; Bang, S. Y. An Efficient Method to Construct a Radial Basis Function, Neural Network Classifier. *Neural Networks* **1997**, *10*, 1495−1503.

(3) Rollet, R.; Benie, G. B.; Li, W.; Wang, S.; Boucher, J. M. Image Classification Algorithm Based on RBF Neural Network and K-means. *Int. J. Remote Sens.* **1998**, *19*, 3003−3009.

(4) Moody, J.; Darken, C. J. Fast Learning in Networks of Locally-Tuned Processing Units. *Neural Comput.* **1989**, *1*, 281−294.

(5) Wan, C.; Harrington, P. de B. Self-Configuring Radial Basis Function Neural Networks for Chemical Pattern Recognition. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1049−1056.

(6) Musavi, M. T.; Ahmed, W.; Chan, K. H.; Faris, K. B.; Hummels, D. M. On The Training of Radial Basis Function Classifiers. *Neural Networks* **1992**, *5*, 595−603.

(7) Lee, S.; Rhee, M. K. A Gaussian Potential Function Network with Hierarchically Self-Organizing Learning. *Neural Networks* **1991**, *4*, 207−224.

(8) Billings, S. A.; Zheng, G. L. Radial Basis Function Network Configuration Using Genetic Algorithms. *Neural Networks* **1995**, *8*, 877−890.

(9) Kennedy, J.; Eberhart, R. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, 1995; Institute of Electrical and Electronics Engineers: Piscataway, NJ, 1995; Vol. 4, pp 1942−1948.

(10) Shi, Y.; Eberhart, R. A Modified Particle Swarm Optimizer. *Proceedings of IEEE World Congress on Computational Intelligence*, Piscataway, NJ, 1998; Institute of Electrical and Electronics Engineers: Piscataway, NJ, 1998; pp 69−73.

(11) Clerc, M.; Kennedy, J. The Particle Swarms Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Transactions on Evolutionary Computation*; Institute of Electrical and Electronics Engineers: Piscataway, NJ, 2002; Vol. 6, pp 58−73.

(12) Shi, Y.; Eberhart, R. Fuzzy Adaptive Particle Swarm Optimization. *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, 2001; Institute of Electrical and Electronics Engineers: Piscataway, NJ, 2001; Vol. 1, pp 101−106.

(13) Lin, W. Q.; Jiang, J. H.; Wu, H. L.; Shen, G. L.; Yu, R. Q. Piecewise Hypersphere Modeling by Particle Swarm Optimization in QSAR Studies of Bioactivities of Chemical Compounds. *J. Chem. Inf. Model* **2005**, *45*, 535−541.

(14) Shen, Q.; Jiang, J. H.; Jiao, C. X.; Lin, W. Q.; Shen, G. L.; Yu, R. Q. Hybridized Particle Swarm Algorithm for Adaptive Structure Training of Multilayer Feed-Forward Neural Network: QSAR Studies of Bioactivity of Organic Compounds. *J. Comput. Chem.* **2004**, *25*, 1726−1735.

(15) Shen, Q.; Jiang, J. H.; Jiao, C. X.; Shen, G. L.; Yu, R. Q. Modified Particle Swarm Optimization Algorithm for Variable Selection in MLR and PLS Modeling: QSAR Studies of Antagonism of Angiotensin II Antagonists. *Eur. Pharm. Sci*. **2004**, *22*, 145−152.

(16) Lin, W. Q.; Jiang, J. H.; Shen, G. L.; Yu, R. Q. Optimized Block-Wise Variable Combination by Particle Swarm Optimization for Partial Least Squares Modeling in Quantitative Structure−Activity Relationship Studies. *J. Chem. Inf. Model.* **2005**, *45*, 486-493.

(17) Shen, Q.; Jiang, J. H.; Jiao, C. X.; Huan, S. Y.; Shen, G. L.; Yu, R. Q. Optimized Partition of Minimum Spanning Tree for Piecewise Modeling by Particle Swarm Algorithm. QSAR Studies of Antagonism of Angiotensin II Antagonists. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 2027−2031.

(18) Garg, R.; Denny, W. A.; Hansch, C. Comparative QSAR Studies on Substituted Bis-(acridines) and Bis-(phenazines)-carboxamides: A New Class of Anticancer Agents. *Bioorg. Med. Chem.* **2000**, *8*, 1835−1839.

(19) Gasteiger, J.; Marsali, M. Iterative Partial Equalization of Orbital Electronegativity−A Rapid Access to Atomic Charges. *Tetrahedron* **1980**, *36*, 3219−3228.

(20) Stanton, D. T.; Jurs, P. C. Development and Use of Charged Partial Surface Area Structural Descriptors in Computer-Assisted Quantitative Structure−Property Relationship Studies. *Anal. Chem.* **1990**, *62*, 2323−2329.

(21) Hopfinger, A. J. *Safe Handling of Chemical Carcinogens, Mutagens, Teratogens and Highly Toxic Substances*; Ann Arbor Press: Ann Arbor, MI, 1980; p 385.

(22) Garg, R.; Kurup, A.; Mekapati, S. B.; Hansch, C. Cyclooxygenase (COX) Inhibitors: A Comparative QSAR Study. *Chem. Rev.* **2003**, *103*, 703−731.

(23) Khanna, I. K.; Weier, R. M.; Yu, Y.; Xu, X. D.; Koszyk, F. J.; Collins, P. W.; Koboldt, C. M.; Veenhuizen, A. W.; Perkins, W. E.; Casler, J. J.; Masferrer, J. L.; Zhang, Y. Y.; Gregory, S. A.; Seibert, K.; Isakson, P. C. 1,2-Diarylimidazoles as Potent, Cyclooxygeanse-2 Selective, and Orally Active Antiinflammatory Agents. *J. Med. Chem.* **1997**, *40*, 1634−1647.

(24) Centner, V.; de Noord, O. E.; Massart, D. L. Detection of Nonlinearity in Multivariate Calibration. *Anal. Chim. Acta* **1998**, *376*, 153−168.

CI600218D