A Genetic Algorithm Based on Prepotency Evolution Using Chaotic Initiation Used for Network Training

Qing-zhang Lü,[†,‡] Jian-hui Jiang,[†] Ru-qin Yu,*[,†] and Guo-li Shen[†]

State Key Laboratory of Chemo/Biosensing and Chemometrics, College of Chemistry and Chemical Engineering, Hunan University, Changsha 410082, P. R. China, and Chemistry Department, Henan Normal University, Xinxiang, 453002, P. R. China

Received October 7, 2002

The concept of "prepotency" is introduced in evolution algorithm. The logistic mapping as a simple and powerful device in the chaos theory is combined with the newly proposed prepotency evolution (PE) algorithm to formulate a new genetic algorithm. PE with a population initialized by chaotic numbers is applied to the multiple-layer feed-forward ANN training (PECNN). The logistic mapping ensures the PE starts each time from different initial population never used before. The newly designed PE operator partially includes the crossover and mutation operations implicitly. The proposed algorithm has a higher convergence speed comparing to the conventional GA. During the PE operation the distances between members would become smaller and smaller until all members turning to be almost identical with the potential best minimum being found. It does not waste searching time surrounding the testing minima like the conventional GA and not show symptoms of overfitting to the training set samples. The combination of logistic mapping and PE used in ANN training makes PECNN be able to test lots of minima rapidly and effectively. This greatly enlarges the opportunity to find the global minimum. The proposed algorithm has been testified by prediction of the frequency data of tetrahedral vibration modes (ν_1 and ν_2) of tetrahalide MX_4^n ions. The results obtained by the proposed PECNN compared favorably with those of the conventional chemometric method PLS regression.

1. INTRODUCTION

The multiple-layer feed-forward artificial neural networks (ANN) and genetic algorithms are two widely used schemes in chemometrics. It has been proved that a well-performed three-layer ANN could actually fit any nonlinear function, which is a very valuable property for chemometric studies. The popular error back-propagation (BP) algorithm[1] as the classical approach used in network training has some limitations such as overfitting and sinking into local optima. The genetic algorithm (GA) as a highly parallel randomly searching algorithm imitates life evolution in the nature according to Darwinian survival of the fittest principle.[2,3] The applications of GAs in chemometrics were reviewed by Leardi,[4] and the combinations of GA and ANN were attempted in some chemistry studies.[5−8] GA optimization procedures were always used for the purpose to avoid sinking into local optima. Many variants of GAs have been suggested according to the concrete situations of the problems involved, the evolution algorithm based on real number coded "chromosome" is an example among others. Although theoretically the GA can search globally in the response space, considering the fact that the distribution of the initial population is rather confined in a limited space, the searching of GA can scarcely reach the global optimum. For this reason, when genetic algorithm is adopted to adjust the linking weights of an ANN, the chance of ANN's premature convergence to local minima still exists. Since much calculation time is wasted during searching surrounding the local minima with GA, the convergence speed of the ANN used to be slow. In this work, the concept of "prepotency" is introduced in evolution algorithm to improve the efficiency of searching and testing each local minimum involved. To realize an exhaustive global search, the chaos concept is also introduced in our scheme. The chaotic dynamic model was first introduced in GA by the present authors to realize the mutation algorithm to maintain the diversity of the population involved.[9] As the chaos is radically different from statistical randomness, its adoption in the present work is to aid the prepotency evolution algorithm to perform a global search. The prepotency evolution (PE) algorithm with a population initialized by chaotic numbers is applied to ANN training (PECNN). The outstanding features of a chaotic system, especially the inherent ability to search the space of interest exhaustively, have been employed to strengthen the performance of the chemometric algorithm studied. The logistic mapping as a simple and powerful device in the chaos theory is combined with the newly proposed prepotency evolution algorithm. This scheme is applied in ANN training.

To testify the PECNN algorithm, the data of tetrahedral vibration modes (ν_1 and ν_2) of tetrahalide MX_4^n ions[10] were examined. The two vibration modes are Raman active. The PECNN results are compared favorably with those obtained by PLS algorithm.

* Corresponding author phone: 86-731-8822782; fax: 86-731-8822782; e-mail: rqyu@hnu.net.cn.
[†] Hunan University.
[‡] Henan Normal University.

10.1021/ci020068t CCC: $25.00
Published on Web 06/06/2003

2. THEORY AND ALGORITHMS

2.1. The Prepotency Evolution Algorithm. Genetic algorithm is well-known and widely used by scientists in many fields. It was modified as evolution algorithm to treat real number chromosome. To reduce searching time wasted in GA surrounding local minima and avoid over-fitting for the training set examples, a prepotency evolution algorithm is introduced in the present paper in ANN training. The prepotency evolution (PE) operation is applied to replace the mutation and partly crossover in GA. The new members produced by conventional mutation operation concern only one of the parent individuals. The PE operation, on the contrary, is based on a pair of parent members. When two parent members are selected, their fitnesses are compared, and the distance between them is calculated. The new member is produced around the better parent member in the super space. Each gene in the new member's chromosome is a random number taken in the range of the aforementioned distance with the corresponding gene of the best parent member located in the middle position (Figure 1a). On other words, the new member can be realized as a random point in a super sphere centered at the better parent point and half super distance between the two parents as radius in the multiple dimension super space. The coordinates of a point in this superspace are the real numbers (genes) of the corresponding chromosome. During the prepotency evolution the distances between members would become smaller and smaller until all members turning to be almost identical with the potential best minimum being found.

2.2. The Chaotic System of Logistic Mapping. Chaos can be considered as the movement within a limited range occurred in a deterministic nonlinear dynamic system. Logistic mapping[11,12] as one of the most important chaotic dynamic mappings defines a noninvertible mapping of the (0,1) interval onto itself. It is perhaps the simplest mapping for studying the period-doubling bifurcation. For a variable x, its $(i+1)$th value is obtained from the ith value by iteration according to the logistic mapping given by

$$x_{i+1} = A \times x_i \times (1 - x_i) \quad \{x:\ 0.0\sim1.0;\ i \text{ is time sequence ordering}\} \quad (1)$$

For different values of the parameter A one can iterate according to eq 1 to obtain x_{i+1}. When A reaches a critical point 3.5699456..., the system enters the chaos domain. With $A = 4.0$ the values of x_{i+1} would take any real numbers between 0.0 and 1.0 and never repeat a value having turned up already. That is to say, the system is able to search all the range of 0.0−1.0 exhaustively. This excellent exhaustively searching ability aids the algorithm to produce quite different initial population. In this work the A is selected as 4.0.

2.3. Three-Layer Feed-Forward ANN. The multiple-layer feed-forward ANN has been widely used in many scientific fields. ANN is constructed with nodes (named as neurons) and linking weights which sharpen the signal transferred between the nodes. The information is transferred through the node by a so-called transfer function based on a sum of all weighted inputs to it. The training of ANN is to find out optimal values of all the linking weights. In this work the supervised three-layer feed-forward ANN is taken as an example and trained with newly designed schemes involving the use of prepotency evolution algorithm and logistic mapping chaotic system. For the ANN its elementary feed forward algorithm is straightforward and quite simple. If the transfer function is f(x), the input I_j to node j is the weighted sum of the outputs of all nodes ($i = 1, 2, ..., n$) connected to it

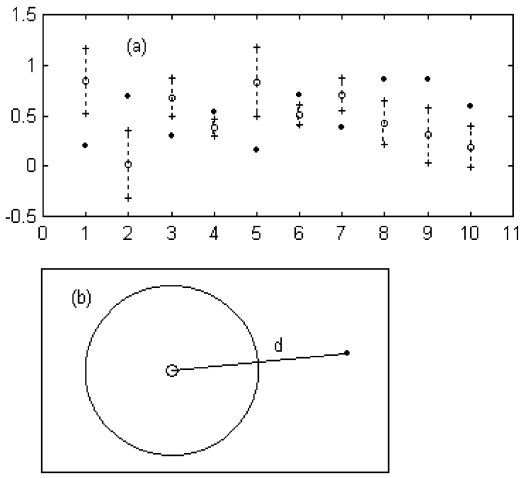

Figure 1. The prepotency evolution operation (Two chromosomes with 10 genes (a) from a given percentage of the best-fitted individuals of current generation are randomly taken as the starting points of PE operation. The distance **d** between the two individuals is calculated. The child individual is produced randomly in a supersphere (b) with the best individual ("o") as the center and 1/2 **d** as the radius.).

$$I_j = \theta_j + \sum_i w_{ji} O_i \quad (2)$$

$$O_j = \mathrm{f}(I_j) \quad (3)$$

where O_i and O_j are the outputs of the ith and jth nodes, respectively, w_{ji} is the linking weight between node j and i, and θ_j is a threshold value. The sigmoid function is selected as transfer function. During the training process, different algorithms can be invited to adjust the w_{ji}-s. GA algorithm is selected to optimize the ANN structure. Each chromosome of the population in GA is a set of all linking weights of an ANN, and each linking weight of the ANN is a gene of a chromosome. Taking advantage of the ability of the proposed GA with chaotic initiation and prepotency evolution to find the global optimum in a complicated searching space with possible local minima, it is used to train the ANN. In all cases the fitness is judged by root-mean-squares (RMS) error of all the scaled values (0−1.0) of samples in the training set which is taken as the objective function in the algorithms discussed below

$$\mathrm{RMS} = \sqrt{\sum_i (O_i^{\mathrm{out}} - d_i)^2 / n} \quad (4)$$

where o_i^{out} and d_i are respectively the actual and desired output values of output node. The values of d_i are scaled into (0, 1.0).

2.4. Prepotency Evolution based on Chaotic Numbers in Network Training (PECNN). After preliminary tests of a few schemes, a more efficient scheme has been designed

to train the ANN by prepotency evolution with the aid of chaotic mapping to produce initial population of GA. In the present work real numbers are adopted to code all the chromosomes of GAs. The ANN trained with prepotency evolution initialized by logistic chaotic system is described as follows.

The first step is randomly producing all the seed initial strings with certain size as the population of GA. The second step is to take advantage of logistic mapping based on the seed to produce new seed (eq 5) and construct the initial population for PE (eq 6)

$$|\text{seed}| = 4 \times |\text{seed}| \times (1 - |\text{seed}|) \quad (5)$$

$$|W| = 2 \times F \times (0.5 - |\text{seed}|) \quad (6)$$

where $|W|$ is the matrix of linking weights of ANN and F is a factor to control the initial weight's span. In this work the value of F is set as 4.0. Each real number coded string (chromosome) in the population stands for a weight set which makes up of one ANN together with all its nodes. In step three, for all samples of the training set, the RMS according to eq 4 of the ANN corresponding to each chromosome of the population will be calculated. Then select a given percentage of the best-fitted members from all the chromosomes of the current generation based on their RMS. The selected members as part of the next generation are also used as parent members to produce new members in the next step. Then single point crossover operation is used to produce a given percentage of new members of the next generation by mating operation between the selected parent members. In this work, the set of weights is divided into two parts, the first consisting all weights between the input and hidden nodes, and the second consisting all weights connecting with output nodes. Exchanging one part of the weights between two ANNs or chromosomes performs the crossover. The PE algorithm described as section **2.1** will produce other new members. This step will be cycled certain times in a subloop. At the end of a subloop, if the RMS of the best chromosome fulfills the desired condition, the training stops with the results output. Otherwise, the procedure goes to the second step to run the chaotic dynamics repeating the subloop with a new population. The PECNN scheme is charted in Figure 2. The program was compiled in MATLAB and run on a Pentium V PC (CPU: 1.5 GHz; Memory: 256 M).

3. THE VIBRATION FREQUENCY DATA OF TETRAHEDRAL MODES OF TETRAHALIDES (MX_4^N)

The data of tetrahedral vibration modes (ν_1 and ν_2) of tetrahalide MX_4^n ions[10] were examined. Among 72 tetrahedral tetrahalide (MX_4^n), 42 samples are used as the training set and 30 samples as the prediction set, though the experimental ν_2 values of 14 ions in the prediction set are unknown. The three-layer ANN composed of 1, 4, and 5 nodes in output, hidden, and input layers, respectively, were used for prediction of the frequency of interest. The input nodes receive the five parameters selected as descriptors, including ionic radius of M in tetrahedral center with high spin, valence and atomic weight of M ion, atomic weight, and radius of halogen atoms. The output node provides the predicted frequency value.

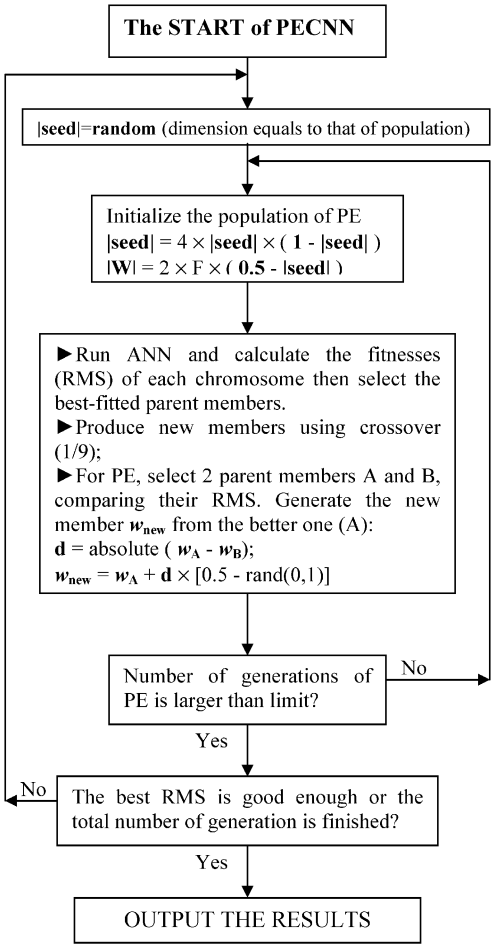

Figure 2. The chart of the PECNN scheme.

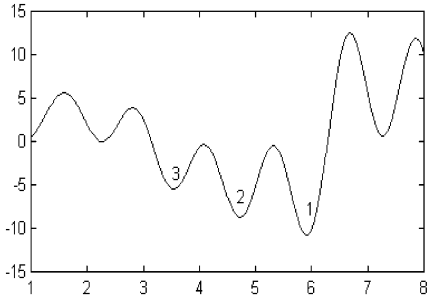

Figure 3. The curve according to eq 7.

4. RESULTS AND DISCUSSION

4.1. Simple Test of the Prepotency Evolution (PE) Algorithm. To test the prepotency evolution algorithm, PE is applied to find the global minimum of a nonlinear function (eq 7) in [1.0, 8.0]. The nonlinear curve is

$$y = x\sin(x) + \sin(x) + (x - 3)\sin(3x) + x\sin(5x) \quad (7)$$

as shown in Figure 2. As this is a one-dimensional minimum search, i.e. each chromosome has only one gene, so no crossover operation in this simple example. All the new members produced only by PE algorithm. When the size of the population is 55 and the parents' number is 10, the prepotency evolution algorithm can always quickly find the global minimum (no. 1 in Figure 3). When the population size is 36 and the parent's number is 8, after running the PE algorithm for 100 times, the PE let all the members converge

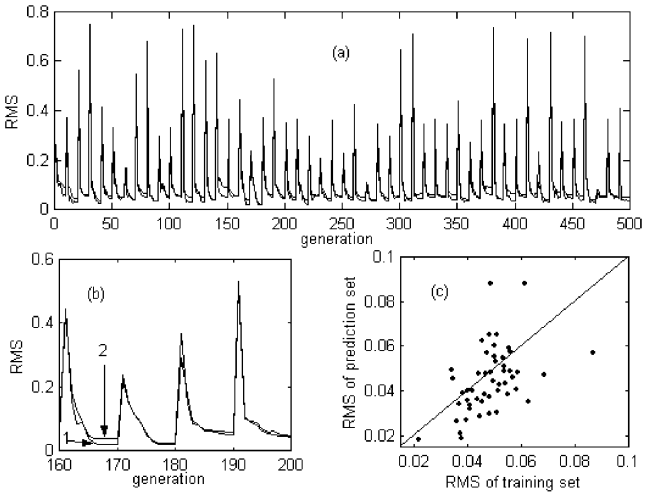

Figure 4. The searching condition of PECNN in 500 generations: (a) the whole convergence curve; (b) a part of convergence curve between 160 and 200 generations; and (c) RMS of prediction set vs RMS of training set for all the 50 minima found by PECNN.

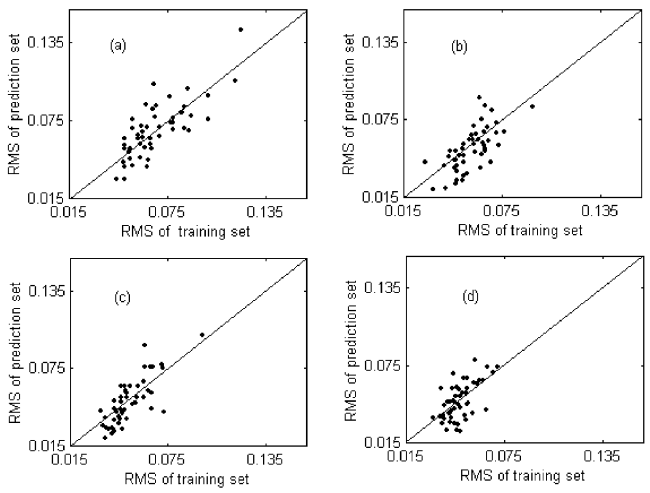

Figure 5. RMS of prediction set vs training set of PECNN for all the searched minima when (a) 20%, (b) 40%, (c) 60%, and (d) 90% new members produced by PE.

94 times to no. 1 minimum, 4 times to no. 2 minimum, and 2 times near the no. 1 minimum. When the population size is 21 and the parent's number is 6, there is 87 times to no. 1 minimum, 4 times to no. 2 minimum, 2 times to no. 3 minimum, and 7 times near the no. 1 minimum out of 100 times running the PE. When the population size is 15 and the parents' number is 5, there is 69 times to no. 1 minimum, 12 times to no. 2 minimum, 2 times to no. 3 minimum, and 17 times near the no. 1 minimum. Based on the data above, one can find when the population size is not large enough one cannot ensure the prepotency evolution to converge to the global minimum. So a sufficient number of cycles running the PE is necessary, and the desired effect of the enlarged number of cycles could be reached only in the case when each cycle is started from a different initial population. This is the exact reason to bring forward the chaos concept to guarantee the diversity of the initial populations.

4.2. Using PECNN Scheme To Predict Frequencies of Tetrahalides. Based on the simple test of the prepotency evolution algorithm, one would realize that a huge enough size of the population is essential for searching the global minimum in multidimensional super space. Considering the limitation of the memory of a personal computer, the population size cannot be defined too largely. So repeatedly running the prepotency evolution algorithm becomes essentially necessary. In PECNN, the logistic mapping ensures the prepotency evolution starts each time from a different initial population never used before. In the present work, the population size is selected as 200 with a selection rate of 10%. In the new generation, besides 10% members selected from the previous generation as parent chromosomes, there are 10% members newly produced by crossover and the remaining 80% newly produced by prepotency evolution. Each subloop is repeated only 10 times, i.e. only 10 generations needed to evolve.

For a discussion of the PECNN, the calculation results concerning ν_1 mode of the tetrahalides are given in detail. The convergence process can be examined in Figure 4. Figure 4a is the whole searching course convergence curves. Curve 1 (Figure 4b) is the convergence curve drawn with the RMS for the training set obtained with the best member in each generation. Curve 2 is the RMS curve for predicted set calculated by this fittest member. Figure 4b is a part of Figure 4a including four times running of the subloop with prepotency evolution algorithm. In each subloop the population evolves only 10 generations based on an initial population produced by logistic chaotic mapping. In fact, when the prepotency evolution cycles about 8 times the algorithm will find a potential "local global" minimum, and all distances between any pair of members are very small, i.e. all the members are almost identical and the population will stop evolving. So 10 cycles in each subloop are just appropriate. More cycles in each subloop seem just to waste calculation time. In each subloop the algorithm can quickly find a best minimum in its searching location. But it used to not be the global minimum nor the satisfactory one for the research of interest. So sufficient times of repeated prepotency evolutions from different initial populations are essential. Fortunately the logistic mapping has an excellent property to fulfill the requirement. Figure 4c presents the RMS-s (RMS of training set vs RMS of prediction set) of all the 50 potential minima in 500 generations of searching. From Figure 4c one can clearly find there is not any pair of identical points. That is to say, the suggested scheme always tends to find out different local minima. This behavior has great potential to find the global minimum or an excellent local minimum. The general genetic algorithm convergence curve was given elsewhere.[9] The conventional GA tends to overfit for the training samples, while the corresponding curve for prediction samples remains unchanged. The prepotency evolution does not show symptoms of overfitting for the training set samples, and the PECNN has many chances to search out better results than the conventional GA.

To check the effect of the rate of new members produced by prepotency evolution (PE) algorithm in each generation, we run the program at 20%, 40%, 60%, and 90% rate of PE with the data of ν_1 mode. All the RMS relationships between prediction and training sets are shown in Figure 5. By comparison among parts a−d of Figure 5 one notices that the higher the rate of PE is, the higher the opportunity for finding lower RMS local minima. Clearly the crossover operation is not so important in PECNN scheme. Is the crossover operation impotent? A deeper examination of the

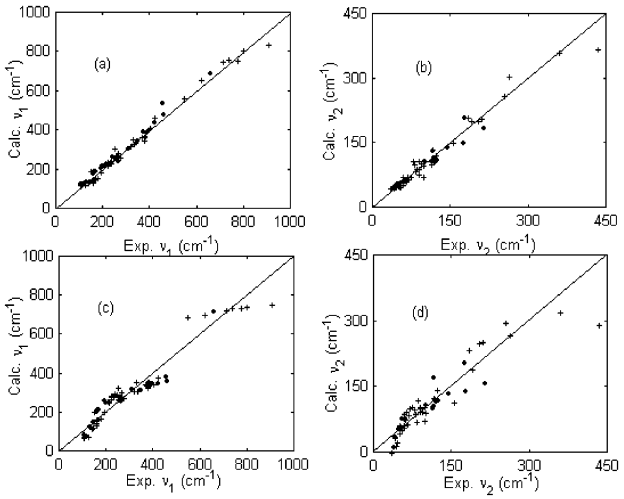

Figure 6. Calculated frequencies vs experimental frequencies: (a) the PECNN calculated ν_1 vs experimental ν_1; (b) the PECNN calculated ν_2 vs experimental ν_2; (c) the PLS calculated ν_1 vs experimental ν_1; and (d) the PLS calculated ν_2 vs experimental ν_2; ("+" stands for training samples; "●" stands for prediction samples).

PE operation suggested in this work would find that PE actually implies two of the actions including crossover and mutation. Just the implied crossover operation is different from the corresponding operation in conventional GA. This seems to be the reason PECNN could function even without the use of crossover operation (Figure 5d). To keep some characteristic of the traditional mating operation and to deal with some unexpectable situations, 10% new members produced by crossover operation are saved in the proposed PECNN algorithm.

So far as the range of new members produced by PE is concerned, we checked the radius at 0.1, 0.3, 0.5, 0.8, and 1.0 times the distance between the pair parent members. Only when the radius is very small the searching result is unsatisfactory. So 0.5 times of distance between the parent members is selected as the radius for producing new member in the suggested PECNN scheme. For the size of population used in the PECNN, a large size population is propitious to rapid convergence in finding the global minimum, but the extent of the increase of the chances for searching the lower RMS minima is not so significant comparing to that of the increase of the population size. Under the suggested population size, 500 generations' searching by PECNN needs only 68 s. A large population size would need more generations in each prepotency evolution subloop and spend much more time to evolve. The high-speed CPU of PC can circumvent this difficulty making the use of a large population size possible when necessary.

A comparison between PECNN and the conventional chemometric method the partial least-squares (PLS) regression has been undertaken. The relationships between experimental frequencies (ν_1 and ν_2 modes) and the calculated results by PECNN and PLS are plotted in Figure 6. The results of PECNN are much better than those of PLS. The calculated frequencies by PECNN are presented in Table 1, and the correlation coefficients between experimental and calculated frequencies by PECNN and PLS are presented in Table 2. The data of RMS of the two methods for the ν_1 and ν_2 modes are also given in the table. The superiority of

Table 1. Experimental and PECNN Values and of ν_1 and ν_2 (cm^{-1})

	exptl		PECNN	
samples	ν_1	ν_2	ν_1	ν_2
	Training Set			
$[BeF_4]^{2-}$	547	255	555.7	257.6
$[MgCl_4]^{2-}$	252	100	257.4	104.8
$[MgI_4]^{2-}$	107	42	119.1	46.9
$[BF_4]^-$	777	360	749.0	357.4
$[BCl_4]^-$	405	190	407.4	198.5
$[AlF_4]^-$	622	210	651.1	203.5
$[AlBr_4]^-$	212	98	224.6	69.8
$[AlI_4]^-$	146	51	133.9	48.9
$[GaBr_4]^-$	210	71	218.4	62.4
$[GaI_4]^-$	145	52	132.8	45.9
$[InCl_4]^-$	321	89	311.4	87.9
$[InI_4]^-$	139	42	129.1	43.3
$[CF_4]$	908.4	434.5	833.8	365.9
$[CBr_4]$	267	123	280.3	117.9
$[CI_4]$	178	90	152.3	74.0
$[SiF_4]$	800.8	264.2	799.5	300.9
$[SiBr_4]$	246.7	84.8	258.2	87.7
$[GeF_4]$	738	205	757.3	198.9
$[GeBr_4]$	235.7	74.7	240.1	68.5
$[GeI_4]$	158.7	60	140.1	49.0
$[SnCl_4]$	369.1	95.2	357.8	100.1
$[SnBr_4]$	222.1	59.4	222.1	60.7
$[PbCl_4]$	331	90	346.5	105.8
$[PBr_4]^+$	254	116	302.0	133.0
$[AsCl_4]^+$	422	156	456.5	147.6
$[ZnCl_4]^{2-}$	276	80	258.4	104.7
$[CdCl_4]^{2-}$	261	84	245.3	82.8
$[HgI_4]^{2-}$	126	35	114.7	42.1
$[TiF_4]$	712	185	743.5	206.0
$[TiBr_4]$	231.5	68.5	232.1	68.1
$[TiI_4]$	162	51	135.1	49.0
$[ZrCl_4]$	377	98	345.3	96.3
$[ZrBr_4]$	225.5	60	212.8	59.2
$[ZrI_4]$	158	43	129.1	44.6
$[HfBr_4]$	235.5	63	229.0	61.1
$[HfI_4]$	158	55	142.8	44.9
$[CrBr_4]$	224	60	234.1	68.5
$[FeCl_4]^{2-}$	266	82	254.5	98.1
$[MgBr_4]^{2-}$	150	61	185.4	64.1
$[MnBr_4]^{2-}$	195	65	180.3	58.3
$[MnI_4]^{2-}$	108	46	116.9	44.9
$[FeCl_4]^-$	333	114	326.9	98.3
	Prediction Set			
$[BBr_4]^-$	243	117	262.5	129.5
$[AlCl_4]^-$	348	119	342.3	113.9
$[GaCl_4]^-$	343	120	331.8	102.6
$[InBr_4]^-$	197	55	206.7	54.6
$[CCl_4]$	460	214.2	474.0	181.4
$[SiCl_4]$	423.1	145.2	433.8	136.6
$[SiI_4]$	168.1	62	144.7	58.8
$[GeCl_4]$	396.9	125	396.0	109.1
$[SnI_4]$	147.7	42.4	135.1	45.1
$[PCl_4]^+$	458	178	532.7	206.5
$[CdBr_4]^{2-}$	161	49	174.5	53.72
$[TiCl_4]$	389	114	384.6	106.7
$[ZrF_4]$	662.5	175	685.4	148.2
$[HfCl_4]$	382	101.5	358.3	105.2
$[CrCl_4]$	373	116	387.9	107.5
$[TlCl_4]^-$	312		301.5	
$[TlBr_4]^-$	192		205.4	
$[TlI_4]^-$	130		133.3	
$[ZnBr_4]^{2-}$	171		183.9	
$[ZnI_4]^{2-}$	118		118.9	
$[CdI_4]^{2-}$	116	39	115.0	43.4
$[MnCl_4]^{2-}$	256		251.6	
$[FeBr_4]^-$	200		215.3	
$[FeBr_4]^{2-}$	162		182.0	
$[NiCl_4]^{2-}$	264		263.0	
$[NiI_4]^{2-}$	105		120.2	
$[CoCl_4]^{2-}$	269		259.8	
$[CoBr_4]^{2-}$	166		185.0	
$[CoI_4]^{2-}$	118		119.3	
$[HgCl_4]^{2-}$	267		236.5	

Table 2. Correlation Coefficients between the Calculated and Experimental Values and the Root Mean Square Errors (RMS)

		ν_1		ν_2	
method		train. set	pred. set	train. set	pred. set
corr coeff	PECNN	0.9943	0.9907	0.9837	0.9584
	PLS	0.9691	0.9441	0.9287	0.8671
RMS (cm^{-1})	PECNN	21.9	19.6	15.5	15.0
	PLS	50.6	44.2	31.8	25.7

PECNN over PLS is remarkable embodied by the data in Table 2.

5. CONCLUSION

In this work the newly proposed prepotency evolution (PE) coupled with chaotic initiation in network training is examined with the frequencies of tetrahedral vibration modes of tetrahalides. PE has a higher convergence speed than the conventional GA. It does not waste searching time surrounding the testing minima like the GA. The logistic dynamic mapping is an uncommon seeding-machine to produce different initial chromosome populations never appeared before for PE algorithm. The combination of logistic mapping and PE used in ANN training makes PECNN be able to test lots of minima rapidly and effectively. This greatly enlarges the opportunity to find the global minimum. Besides the ANN training, the combination of PE and chaotic mapping also can be used in other optimization schemes. Because the PE does not tend to over fit the training samples, so it will be useful for treating the data having a large random error like the quantitative-structure−activity-relationship (QSAR) studies in medicine activity studies and related fields.

ACKNOWLEDGMENT

This work was supported by grants from the natural science foundations of China (Grant No. 20075006).

REFERENCES AND NOTES

(1) Soil, D.; Kanaka, V.; Hospital, J. Introduction to Multi-Layer Feed-Forward Neural Networks. *Chem. Intel. Lab. Sys.* **1997**, *39*, 43−62.

(2) Hebert, D. B. Genetic Algorithms in Chemistry. *Chem. Intel. Lab. Sys.* **1993**, *19*, 277−293.

(3) Levine, B. K.; Moores, A. J. Genetic Algorithm in Analytical Chemistry. *Anal. Let.* **1999**, *32*, 433−445.

(4) Lardy, R. Genetic algorithms in chemometrics and chemistry: a review *J. Chemo.* **2001**, *15*, 559−569.

(5) Hunger, J.; Hutter, G. Optimization and analysis of force field parameters by combination of genetic algorithms and neural networks. *J. Comput. Chem.* **1999**, *20*, 455−471.

(6) Tarsal, A.; Grand jean, B. P. A.; Larachi, F. Integrated Genetic Algorithm-Artificial Neural Network Strategy for Modeling Important Multiphase-Flow Characteristics. *Ind. Eng. Chem. Res.* **2002**, *41*, 2543−2551.

(7) So, S.-S.; Karplus, M. Evolutionary Optimization in Quantitative Structure−Activity Relationships: An Application of Genetic Neural Networks. *J. Med. Chem.* **1996**, *39*, 1521−1530.

(8) Burden, F. R.; Rosewarne, B. S.; Winkler, D. A. Predicting maximum bioactivity by effective inversion of neural networks using genetic algorithms. *Chemo. Intell. Lab. Sys.* **1997**, *38*, 127−137.

(9) Lu, Q.-Z.; Sheen, G.-L.; Yu, R.-Q. Genetic Training of Network Using Chaos Concept: Application to QSAR Studies of Vibration modes of Tetrahedral Halides. *J. Comput. Chem.* **2002**, *23*, 1357−1365.

(10) Navasota, K. I*nfrared and Raman Spectra of Inorganic and Coordination Compounds*, 4th ed.; Wiley: New York, 1986; Part 2.

(11) May, R. M. Simple mathematical models with very complicated dynamics. *Nature* **1976**, *26*, 459−467.

(12) Feigenbaum, M. J. Qualitative universality for a class of nonlinear transformations. *J. Statist. Phys.* **1978**, *19*, 25−52.

CI020068T