第 29 卷 第 2 期
2003 年 3 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 29, No. 2
Mar., 2003

Genetic Algorithm with Dynamic Regional Multi-Species[1)]

LI Ming   YANG Xiao-Qin   ZHOU Lin-Xia

(*Department of Test & Control Engineering, Nanchang Institute of Aeronautical Technology, Nanchang   330034*)
(E-mail: limingl@public. nc. jx. cn)

Abstract   A novel genetic algorithm with dynamic regional multi-species is proposed in this paper. Each of those genetic species occupies a dynamic region that is determined by the weight vector of a fuzzy adaptive Hamming neural network. Through learning and classification of genetic individuals in the evolutionary procedure, the neural network distributes multi-species into different regions of the search space. Furthermore, the neural network dynamically expands each search region or establishes new region for good offspring individuals to continuously keep the diversification of genetic population. As a result, the premature problem inherent in genetic algorithms is alleviated and better tradeoff between the ability of exploration and exploitation can be obtained. The experimental results on the optimization computation of typical multi-modal functions also have shown good performance of the proposed genetic algorithm.

Key words   Genetic algorithms, multi-species, neural networks, premature problem

1   Introduction

The evolutionary procedure of the biological organisms has been a vitally important feature over thousands of years on earth. This nature feature has led to the genetic algorithms which has been used as an alternative method to the traditional optimization strategies. The theoretical research and engineering application for the genetic algorithms have shown its peculiar and excellent performances over other traditional optimization methods[1~4]. However, the premature convergence is still the outstanding problem in genetic algorithms. Premature convergence implies that the genetic algorithms converge to local optima, which is mainly caused by the following reasons: 1) the key alleles may be probably lost in the procedure of genetic selection; 2) the key alleles may be broken by the genetic crossover operation, and genetic population could not reserve enough diversified genes, so the genetic population is occupied by a few local-optimized individuals; 3) the unsuitable control parameters may cause the unbalanced relationship between global and local searches. Various heuristic schemes for alleviating the premature problem and making the genetic algorithms faster have been proposed in the concerned literature[5~10]. They focused on the selection operator, crossover operator, parameter setting, encoding, search procedure, multiple species, etc.

Genetic algorithm based on multiple species involves more than two species and the immigration between different species. The different species evolves with different control parameters and the immigration among species grabs and reserves better individuals. So it can help to maintain the diversification for gene pool and to obtain higher global convergence rate. For example, GAMAS[7] is a typical genetic algorithms based on multiple species. The genetic algorithms based on sharing fitness distribute individuals into different peaks in the fitness map, it is also helpful for diversifying the genetic population.

A novel genetic algorithm with dynamic regional multi-species is proposed in this paper. It includes multiple species which are located in the different regions of the search

space. The region occupied by each species is determined by classifying all individuals, and the region is dynamically changed in the evolutionary procedure. The classifying is based on fuzzy adaptive Hamming neural networks.

The evolutionary operations consist of the genetic operators in each region and between the regions, in which the former presents better local search ability while the later presents better global search ability, so the good tradeoff between global exploration and local exploitation has been obtained. The proposed genetic algorithm can guarantee the population diversification and effectively alleviate the premature problem inherent in the genetic algorithm. Also the experimental results have shown that the proposed method is better than the general genetic algorithms based on general multi-species and that based on sharing fitness.

2   Genetic algorithm with dynamic regional multiple species

2. 1   Evolutionary diagrams

The proposed algorithm first establishes M regional species by random initialization, then carries on evolution inside single species and evolution among different species. Finally the algorithm tests if it has converged. The evolutionary diagrams are shown in figures $1 \sim 3$.
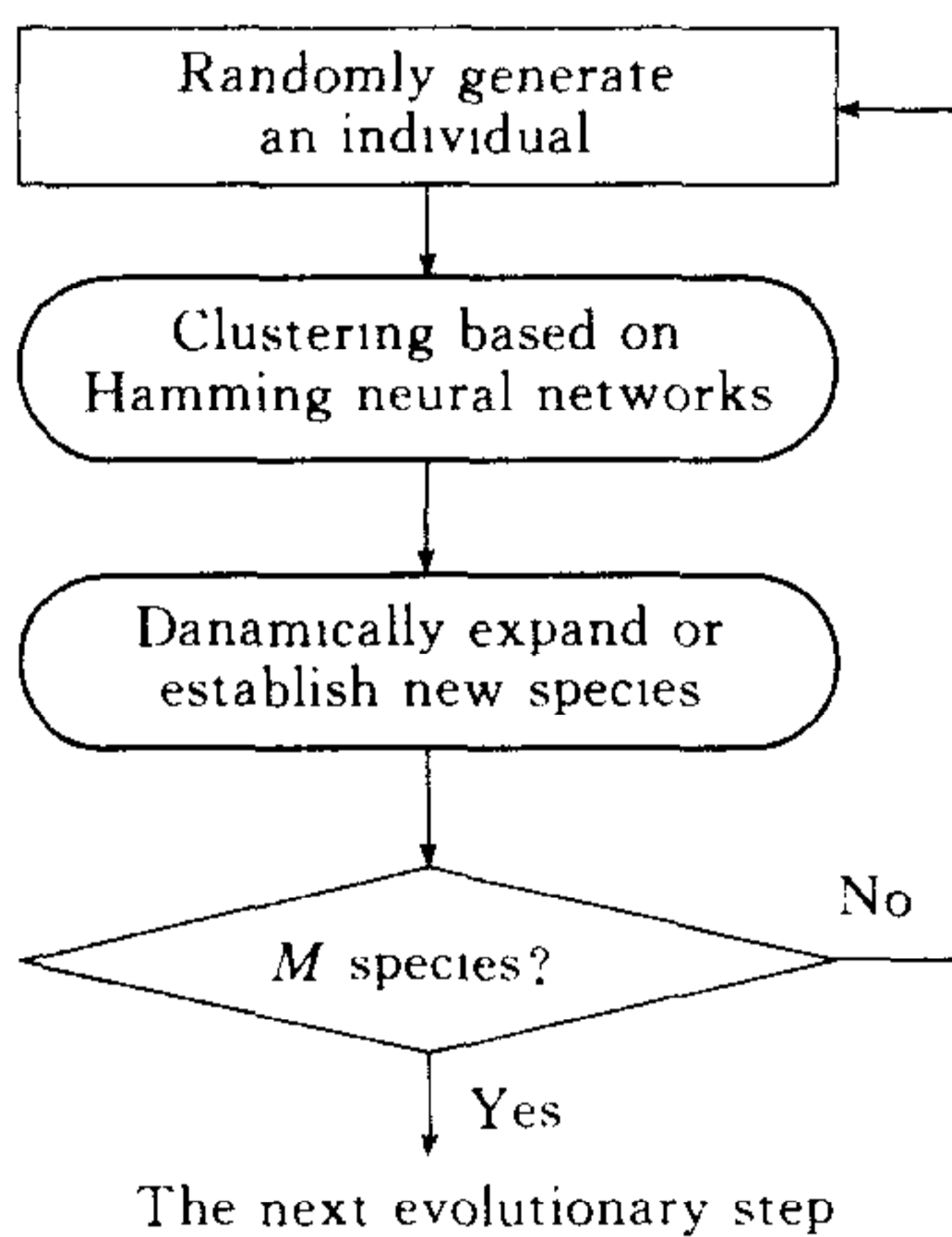
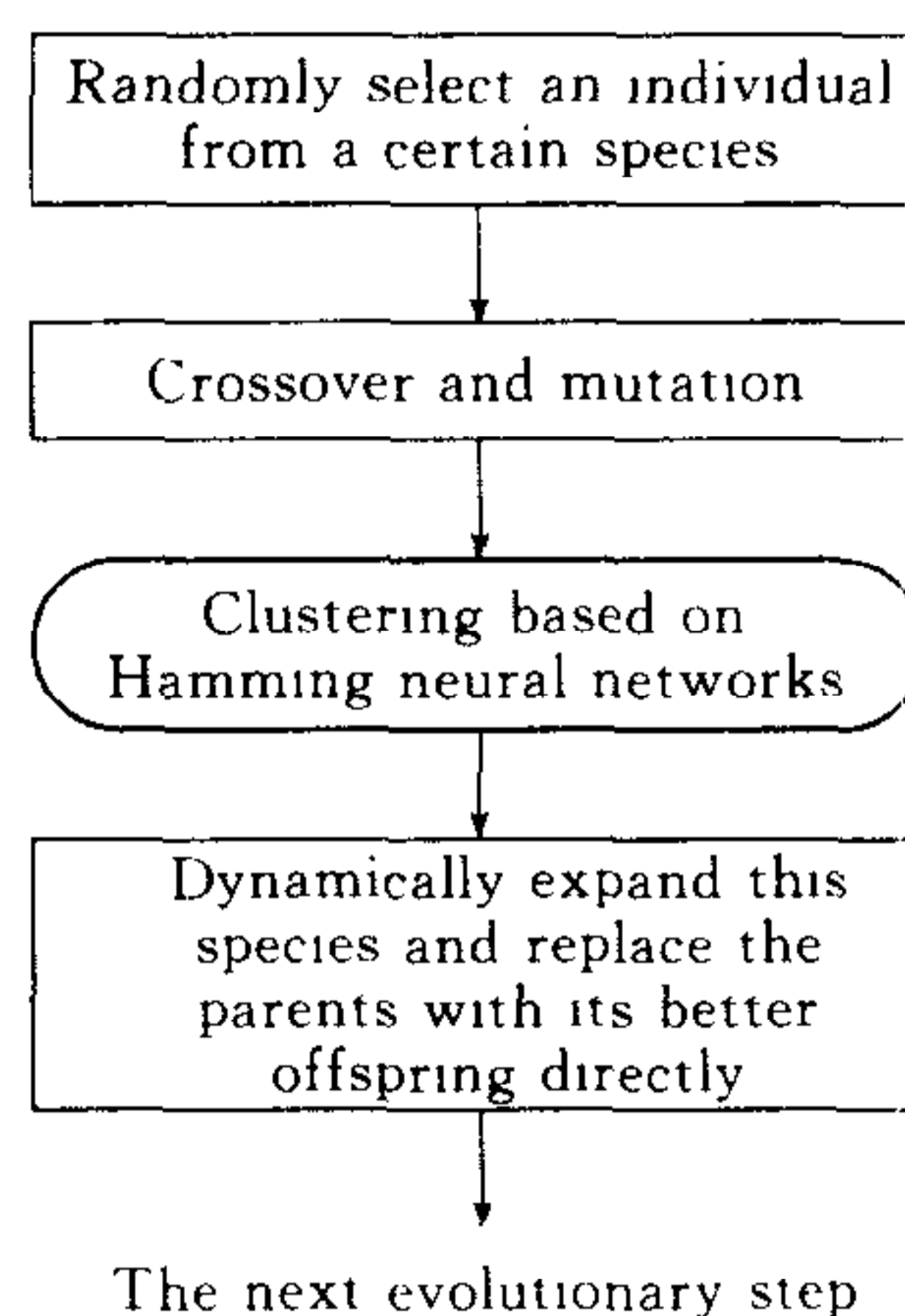Fig. 1   Random initialization
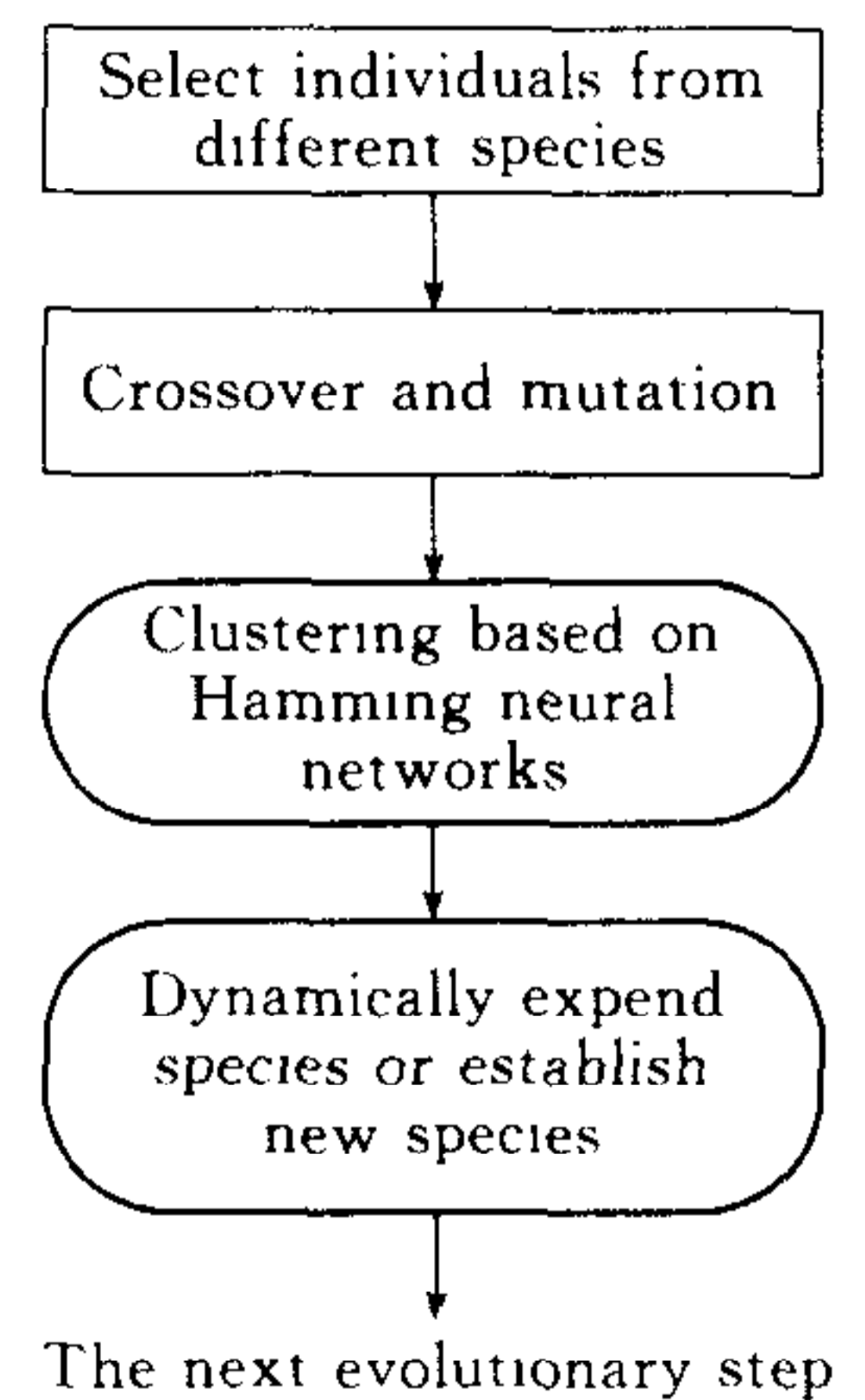
Fig. 2   Evolution inside single species

Fig. 3   Evolution among different species

M output nodes of the adaptive fuzzy Hamming neural networks correspond to M clusters which occupy M hyper-rectangles[11,12] in the vector space. In this paper, the search space of the genetic algorithm is mapped onto the vector space of the neural networks. And these M clusters are attached to M genetic species, so each species corresponds to a unique hyper-rectangle in the search space.

As shown in figure 1, in the initialization step, enough individuals are generated randomly and are classified into M species by the Hamming neural networks. That is, M hyper-rectangles (M species) are generated by initialization, and there are N individuals in each of these species. So the population size is $N \times M$ individuals. Evolution inside a single species and evolution among different species are shown in figures 2 and 3, respectively. The alternative evolutions inside and among species favor a good balance of global and local searches.

2.2 The relationship between the dynamic region of genetic species and clusters of neural networks

Assume a generalized genetic individual is a K dimensional vector: $X = \{x_1, x_2, \cdots, x_K\}$ where $x_k \in [0,1]$; $k = 1 \sim K$. In order to guarantee the convergence of learning for the fuzzy adaptive Hamming neural networks [11,12], every genetic individual is encoded as a complementary vector $I = (X, X^C)$ to the input to the Hamming net, but actually the vector X only corresponds to a unique I. Assume the weight vector of the Hamming net is $W = \{W^j : j = 1 \sim M\}$ where $W^j = \{w^{ij} : i = 1 \sim 2K\}$, the number of the output nodes equals to the number of total species, so each W^j corresponds to one output node, thus corresponds to a species. The W^j determines the location and the size of its hyper-rectangle[11,12], and furthermore determines the region of a species. In the evolutionary procedure, the total number of neural output nodes is constant. So if a new species is established, an old one must be destroyed.

Assume the vigilance parameter of Hamming net is ρ. If input vector I satisfies the following criterion:

$$\frac{|I \wedge W^j|}{|I|} \geqslant \rho \text{ where } j \in \Omega \tag{1}$$

then the input vector is classified into cluster J. J can be calculated by the following equation

$$J = \arg \underset{j \in \Omega}{\text{Maximum}} \frac{|I \wedge W^j|}{\alpha + |W^j|} \tag{2}$$

where α is a positive real-valued number called the choice parameter. For a new generated individual I, if no any cluster J satisfies equations (1) and (2) and the fitness of individual I is higher than that of individuals in an existing species, new species will be established to replace the old existing one.

Similar to the structure of the complementary code, the weight vector of Hamming net can be interpreted as $W^J = (U^J, (V^J)^C)$ where $U^J = \{u_1^J, u_2^J, \cdots, u_K^J\}$ and $V^J = \{v_1^J, v_2^J, \cdots, v_K^J\}$ have the same dimensions as that of individual X and satisfies that $u_i^J \leqslant v_i^J$, $\forall i, J$. Just as shown in Fig. 4 (take 2 dimensional vector X as example), vectors U^J and V^J determine a K-dimensional hyper-rectangle R^J in the K-dimensional space $[0,1]^K$. Vector points U^J and V^J correspond to the corners nearest to and furthest from the origin of the coordinate system, respectively.
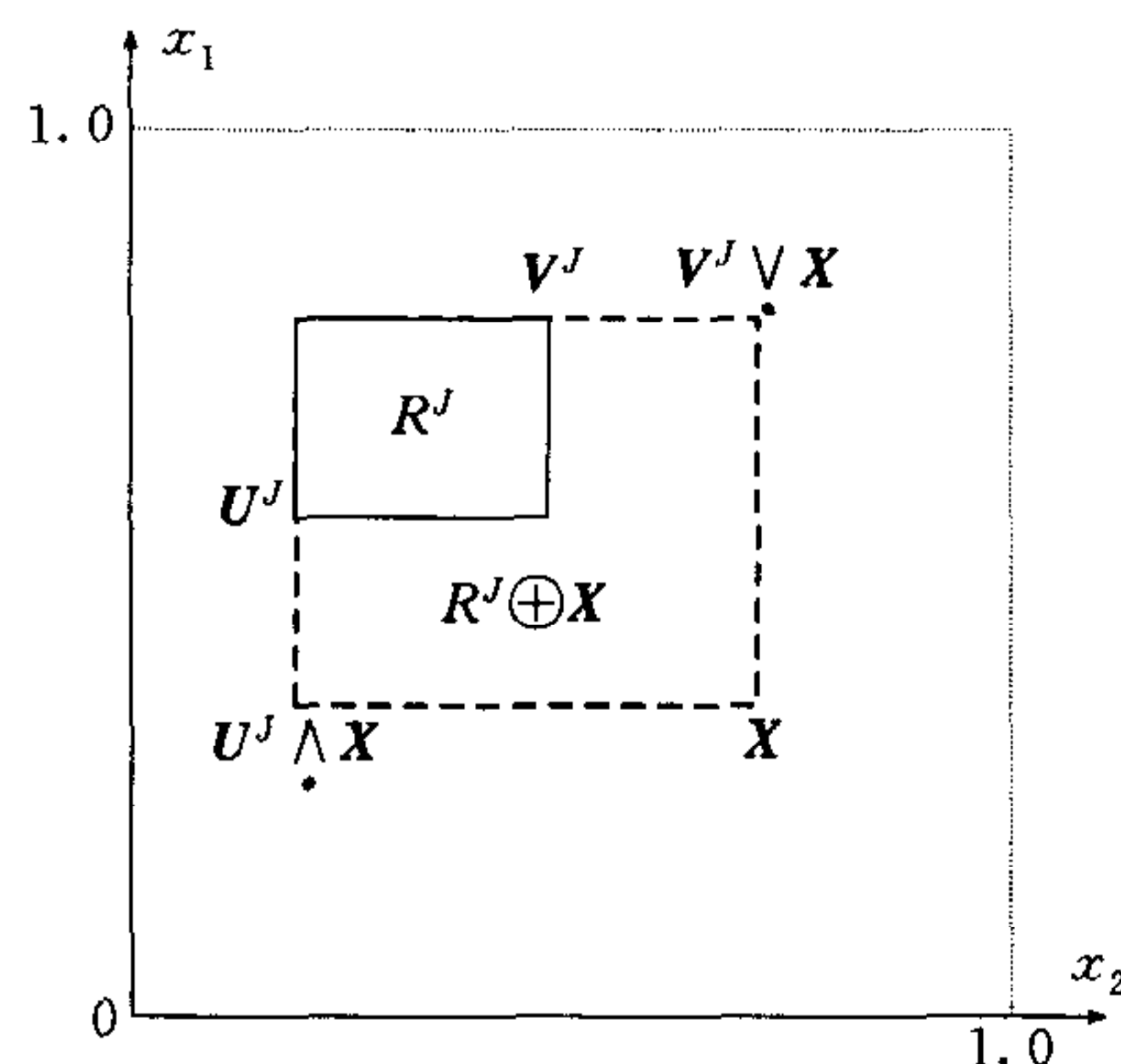

Fig. 4   The hyper-rectangle of the Hamming net and its expansion

2.3 Dynamic change of the genetic region

Each species has a best individual inside its hyper-rectangle, so M species have M best

individuals. To the j-th species (also the hyper-rectangle R^j), its highest fitness value is f^j, so the fitness values of the best individual set Ψ are f^j ; $j = 1 \sim M$. Assume that the best individual in the S-th species (also in the hyper-rectangle R^S) has the lowest fitness value in the set Ψ, that is,

$$f^S = \underset{j \in \Psi}{\text{Minimum}} f^j \tag{4}$$

When a new offspring individual X is generated, its corresponding complementary vector I is inputted into fuzzy adaptive neural Hamming net. The net will classify the vector I as follows.

I) If I is classified into an existing hyper-rectangle R^j, then expand R^j and carry on learning for the neural weight vector W^j as shown in figure 4 and the following two equations:

$$W^j(new) = W^j(old) \wedge I \tag{5}$$

$$R^j(new) = R^j(old) \oplus X \tag{6}$$

The above equations can be divided as

$$U^j(new) = U^j(old) \wedge X \tag{7}$$

$$V^j(new) = V^j(old) \vee X \tag{8}$$

From the vigilance criterion of equation (1), the largest size of any hyper-rectangle should satisfy the constraint

$$|R^j(old) \oplus X| \leqslant (1 - \rho)K \tag{9}$$

where the size of hyper-rectangle R^j is determined as

$$|R^j| = |V^j - U^j| \tag{10}$$

II) If I does not belong to any existing hyper-rectangle and its corresponding individual's fitness value $f(X) \leqslant f^S$, then individual X is abandoned directly.

III) If I does not belong to any existing hyper-rectangle, but its corresponding individual's fitness value $f(X) > f^S$, a new hyper-rectangle is established to replace the old hyper-rectangle R^S, that is, $R^S(new) = X$. Consequently a new species is set up.

2.4 Evolution inside each species

Take species J as example, select two parents from species J, and generate new offspring individuals by crossover and mutation operations. Each offspring X is classified by Hamming net then.

If X still belongs to species J (that is $X \in R^J$), expand the hyper-rectangle R^J as described in the above subsection. Then X is used to replace its parent directly if its fitness value is higher. In fact, the expansion of hyper-rectangle R^J has recorded the evolutionary search history inside species J, so that redundant searches can be avoided and the evolutionary search can be faster by using the direct replacement.

If X does not belong to species J (that is $X \notin R^J$), ignore and discard this new generated offspring X.

2.5 Evolution among different species

First randomly select two individuals from two different species, crossover and mutate them to generate two offspring individuals. Assume X is one of the offspring. Hamming net classifies X then.

If X belongs to species J (that is $X \in R^J$), expand the hyper-rectangle R^J as described in the above subsection. Then X is used to replace its parent directly if its fitness value is higher.

If X does not belong to any existing species, abandon X or establish a new species for X according to the cases II) and III) in subsection 2.3.

2.6 Discussion on dynamic regional multiple species

The using of the dynamic regional multiple species has four main characteristics.

I) Each species occupies a certain region which is a hyper-rectangle of adaptive fuzzy

Hamming net. And all species locates at different regions of the search space. Thus the diversification of genetic population can be maintained, and the premature problem can be alleviated.

II) The direct replacement of parents with their better offspring can enhance the convergence speed.

III) The evolution inside species presents high mutation rate, and plays the role mainly on local search. The evolution among species presents high crossover rate, and plays the role mainly on global search.

IV) The combination of evolution inside each species and among different species balances the tradeoff between exploration and exploitation of the genetic algorithm.

3 The experimental results

The proposed genetic algorithm, standard genetic algorithm, GAMAS and genetic algorithm based on sharing fitness have been carried out optimizing F6 function and Griewangk function in this section. All the experiments are implemented on a PC-PIII800/256MRAM.

3.1 Optimizing F6 function

F6 function is a typical two-dimensional multi-modal function[7], which is

$$f(x,y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2} \tag{11}$$

where its global optimized value locates at $(x=0, y=0)$, and there are infinite local optimized values near the origin $(x=0, y=0)$.

In order to contrast the performance for all mentioned genetic algorithms, the population size for all methods is set to 200 individuals. And each method has carried out 50 times to obtain the average results. The control parameters for the proposed method are: the number of multiple species is 10, the mutation rate inside species is 0.15, the crossover rate among species is 0.50, the vigilance parameter is 0.96. The control parameters for GAMAS are the same as that in [7]. The main control parameters for GA on sharing fitness[8] are $\gamma=0.5$, $\eta_{share}=2.2$. The experimental results are shown in Table 1. From the table, the proposed genetic algorithm obtained 100% global convergence rate which is much higher than other mentioned methods and it does not need more time to converge.

Table 1   The experimental results on F6 function

Genetic algorithms	Global convergence rate(%)	Average convergence speed(Sec)
Standard genetic algorithm	24	7.32
GAMAS	52	9.45
GA on sharing fitness	30	5.77
The proposed GA	100	6.46

3.2 Optimizing Griewangk function

Griewangk function[13] is a multi-dimensional function often used to test the performance of genetic algorithms, Griewangk function is defined as

$$f(X) = \sum_{i=1}^{N} x_i^2/4000 - \prod_{i=1}^{N} \cos(x_i/\sqrt{i}) + 1 \quad -600 < x_i < 600 \tag{12}$$

where N is the dimension of the function. The global optimized value locates at $x_i = 0$; $\forall i$, and the local optimized value locates at $x_i = \pm k\pi\sqrt{i}$; $i,k=1,2,\cdots$. The standard genetic algorithms are always premature at local optimized points.

The optimization results on Griewangk function by the proposed GA and the GA on sharing fitness are shown in Table 2. Fifty times of the implementation for each method have been carried out and Table 2 has shown the average results. The population size for

the two methods is set to 200 individuals. The control parameters for the proposed method are: the number of multiple species is 20, the mutation rate inside species is 0.20, the crossover rate among species is 0.75, the vigilance parameter is 0.90. The control parameters GA on sharing fitness[8] are $\gamma = 2.0$, $\eta_{share} = 3.0$. Table 2 has also shown good performance of the proposed GA.

Table 2   The experimental results on Griewangk function

Dimension	GA on sharing fitness			The proposed GA		
	Speed (Second)	generations	Global convergence rate(%)	Speed (Second)	generations	Global convergence rate(%)
10	32.7	43962	24	25.3	7350	92
50	663.5	172934	10	547.5	2087I	78
100	2077.3	240665	2	2746.5	37693	70

When selecting different control parameters for the proposed GA, it still presents good performance both at global convergence rate and at convergence speed. According to these experiments, the mutation rate inside species and the crossover rate among species can be set in the range of 0.1~0.2 and 0.5~0.6, respectively, and the vigilance parameter can be set near 0.9.

Besides the above optimization experiments, the proposed method is also used to find the best solution for TSP problem and to optimize the neural network's structure and weights. The experimental results have also shown that the proposed method is obviously better than standard GA, GAMAS and GA on sharing fitness.

References

1   Zhou Ming, Sun Shu-Dong. Genetic Algorithms: Theory and Applications. Beijing: National Defense Press, 1996 (in Chinese)

2   Xuan Guang, Chen Run-Wei. Genetic Algorithms and Engineering Design. Beijing: Science Press, 2000(in Chinese)

3   Baeck T, Hammel U, Schwefel H P. Evolutionary computation: Comments on history and current state. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 3~17

4   Yao X, Liu Y, Lin G. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 1999, 3(2): 82~102

5   Leung Y, Gao Y, Xu Z-B. Degree of population diversity—A perspective on premature convergence in genetic algorithms and its Markov chain analysis. *IEEE Transactions on Neural Networks*, 1997, 8(5): 1165~1176

6   Grefenstette J. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 1986, 16(1): 122~128

7   Potts J C, Terri D G, Surya B Y. The development and evaluation of an improved genetic algorithm based on migration and artificial selection. *IEEE Transactions on Systems, Man, and Cybernetics*, 1994, 24(1): 73~86

8   Goldberg D E, Richardson J. Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the 2nd International Conference on Genetic Algorithms, Cambridge, MA: Lawrence Erlbaum, 1987. 41~49

9   Li Mao-Jun, Tong Diao-Sheng. A partheno-genetic algorithm and analysis on its golbal convergence. *Acta Automatica Sinica*, 1999,25(1): 68~72(in Chinese)

10 Wang Hong-Gang, Zeng Jiang-Chao, Xu Yu-Bin. An excellent schemas selft-learning genetic algorithm. *Acta Automatica Sinica*, 1999,25(2):375~379(in Chinese)

11 Paul S W, LI Ming. Supervised and unsupervised fuzzy adaptive Hamming net. Pattern Recognition, 1999, 32(10): 1801~1816

12 Hung C-A, Lin S F. Adaptive Hamming net: A fast-learning ART1 model without searching. *Neural Networks*, 1995, 8(4): 605~618

13 Muhlenbein H, Schomisch M, Born J. The parallel genetic algorithm as function optimizer. *Parallel Computing*, 1991, 17(5): 619~632

14 Doo-Hyun Choi, Se-Yong Oh. A new mutation rule for evolutionary programming motivated from backpropagation learning. *IEEE Transactions on Evolutionary Computation*, 2000, 4(2): 188~193

动态区域性多群体搜索的遗传算法

黎 明   杨小芹   周琳霞

（南昌航空工业学院测试与控制工程系   南昌   330034）

（E-mail: limingl@public. nc. jx. cn）

摘 要   提出了一种新的动态区域性多群体搜索的遗传算法. 该方法的各个遗传群体所占据的搜索空间由自适应模糊 Hamming 神经网络的决定，此神经网络通过对遗传个体分类和学习，将不同的遗传群体分配在搜索空间的不同位置，并可以动态地调整遗传群体的搜索区域或建立新的遗传群体，从而确保了遗传群体的个体多样性，有效地抑制了可能发生的早熟收敛现象，而且使得遗传算法具有较强的全局寻优能力和快速局部寻优能力. 本文的实验通过对典型的复杂多模函数的优化计算，也显示了动态区域性多群体搜索的遗传算法的优良性能.

关键词   遗传算法，多群体搜索，神经网络，早熟收敛

中图分类号   TP18