Virtual Creatures Controlled by Developmental and Evolutionary CPM Neural Networks

Zhi-Hua Zhou* and Xue-Hua Shen

National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

Abstract In this paper, a system where virtual creatures called *bugs* navigating a grid-based environment, which is controlled by developmental and evolutionary CPM neural networks, is presented. Each bug is born with a certain amount of energy that decreases in the navigation and increases only when the bug gets food. The bug can accumulate experience, i.e. training instances, in its life, which is used to incrementally tune its CPM network to improve the chance of making good decisions in later navigation. If two bugs meet then they may fight each other or produce an offspring, which is determined by their gender. The controlling organ, i.e. the CPM neural network, of the offspring is inherited from its parents in a specific way that the experience, i.e. the training instances, of its parents instead of the knowledge, i.e. the architectures or the weights, of them is genetically transmitted. Simulations show that the CPM networks are valuable to the longevity of the bugs, which exhibits not only the importance of the interaction of the developmental and evolutionary processes to virtual creatures, but also the feasibility of introducing evolution at the level of training instances into artificial neural networks.

Keywords Artificial life, artificial neural networks, evolution, virtual creatures, coulomb potential model

1. Introduction

In the mid of 1990s, Dyer [4] propose several open problems for the research of artificial life, including establishing a developmental process and an evolutionary process, by which artificial neural networks could be automatically created and modified via interaction with the environment, so that a group of interacting agents controlled by those artificial neural networks are able to demonstrate behaviors relevant to life. Those open problems emphasize the importance of developing virtual creatures controlled by artificial neural networks, which is definitely helpful to the investigation of *life-as-it-could-be*.

During last decades, many researchers have focused on this area, and many kinds of virtual creatures have been developed. Collins and Jefferson [3] evolved a group of virtual ant colonies controlled by artificial neural networks for the problem of *central place food foraging*, where the objective of the virtual ants is to walk along entire pheromone trails in order to collect all the food in an virtual environment. Fullmer and Miikkulainen [6] built virtual creatures controlled by artificial neural networks whose genetic encoding was loosely based on the marker structure of biological DNA, which had the ability of recognizing some simple objects through developing high-level finite-state exploration and discrimination strategies. Werner and Dyer [13] constructed diversified virtual creatures called *biots* controlled by artificial neural networks, which evolved more and more complex and effective strategies for

* Corresponding author. Tel.: +86-25-359-3163, fax: +86-25-330-0710.

E-mail addresses: zhouzh@nju.edu.cn (Z.-H. Zhou).

prey and predator avoidance in a virtual environment called *BioLand*. Kerce [7] created virtual marine snails controlled by artificial neural networks trained by Hebbian rule, which lived in a virtual ecosystem where food, sunshine, and turbulence existed. Saunders and Pollack [12] utilized recurrent neural networks to realize virtual ants that evolved a communication scheme over continuous channels which conveyed task-specific information when the virtual ants were learning to follow broken trails of food. Cangelosi and Parsi [2] realized a virtual environment where a simple language emerged in the inhabitant artificial neural networks when they were evolving the perceptual ability of categorizing edible and poisonous mushrooms in order to decide whether to approach or avoid encountered mushrooms. Möller *et al.* [9] developed a specific neural network architecture which demonstrated how the visual landmark navigation could be implemented in the brains of central-place foragers such as bees for accurately returning to important locations.

Note that these cases are only a little piece of the iceberg of previous works on creating virtual creatures controlled by artificial neural networks. Here we do not hope, and unable, to provide an exhaustive list for all the previous works in this area. Moreover, although beyond the scope of this paper, it is worth mentioning that besides these works, there are many researchers focusing on designing real organisms, such as robots, controlled by artificial neural networks [5, 8].

The key of the success of the virtual creatures mentioned above is that their controlling organs, i.e. the artificial neural networks, can change both phylogenetically due to the evolutionary process and ontogenetically due to the learning process. As Nolfi and Parisi [10] indicated, evolution may select good starting conditions that enhance the learning process or canalize it in the right directions, learning may help evolution to find good solutions and to adapt to fast changing environments that cannot be tracked by evolution alone.

In this paper, we present a BUG system where virtual creatures called *bugs* navigate a 2-D environment where food and roadblocks exist. Each bug is born with a certain amount of energy. The energy of a bug decreases as the bug navigating the environment. Only when the bug gets food, its energy increases. If the energy of a bug decreases to zero then the bug dies. The bug is controlled by a developmental and evolutionary CPM neural network [1], the inputs of which are the information of the bug and the occupancy of its neighboring grids, and the output of which is the decision for the next movement. The bug can accumulate experience, i.e. training instances, in its life, which is used to incrementally tune its CPM network to improve the chance of making good decisions for later movements. When two bugs meet, if they are with the same gender then they will fight each other; otherwise they will produce an offspring. The controlling organ, i.e. the CPM neural network, of the offspring is inherited from its parents. But different to prevailing styles, here the learned knowledge, i.e. the architectures or the weights of the CPM networks, of the parents is not directly passed down. Instead, the experience, i.e. the training instances of the CPM networks, of the parents is genetically transmitted. Simulations show that bugs controlled by the developmental and evolutionary CPM neural networks almost always live far longer than those controlled by random strategies do. This exhibits not only the usefulness of the control of artificial neural networks to virtual creatures, but also the importance of the interaction of the developmental and evolutionary processes to virtual creatures. The success of our virtual creatures also shows that the evolution could be introduced into artificial neural networks at the level of training instances.

The rest of this paper is organized as follows. In Section 2, the BUG system is described. In Section 3, the controlling organ of the bugs, i.e. the developmental and evolutionary CPM neural network, is presented. In Section 4, simulations on the BUG system are reported. Finally in Section 5, the main contributions of this paper are summarized and several issues for future works are indicated.

2. The BUG system

The BUG system realizes a 2-D grid-based virtual environment where bugs, food, and roadblocks exist, as shown in Fig. 1.

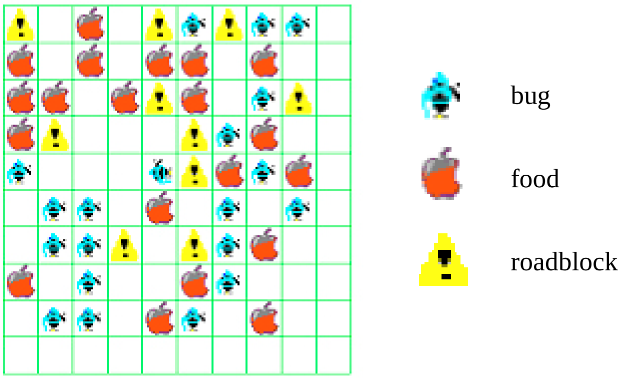

Fig. 1 2-D grid-based virtual environment realized in the BUG system.

The size of the environment is $N \times N$ where N is an integer. When the environment is initialized, a certain amount of bugs, food, and roadblocks are randomly distributed in the environment. The amount of food will be increased at a certain rhythm while that of the roadblocks won't change. Each bug, food, and roadblock occupies a grid. The size of the environment can be set in the system. The percentage of the grids occupied by the bugs, the food, and the roadblocks can also be set, which could be used to craft the amount of those items appearing in the environment, as shown in Fig.2.

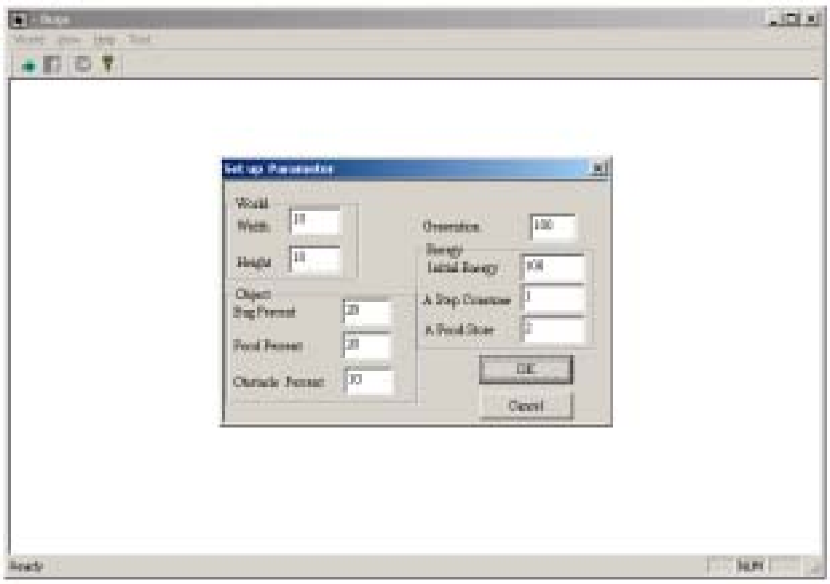

Fig. 2 Configure the environment through setting parameters.

The bugs are controlled by the developmental and evolutionary CPM neural networks that will be presented in Section 3. Each bug has local perceptual ability, that is, it could perceive the occupancy of its neighboring eight grids, as shown in Fig. 3. In each epoch, a bug could take one of five navigational operations, that is, go one step up, down, left, right, or stay, as shown in Fig. 3.

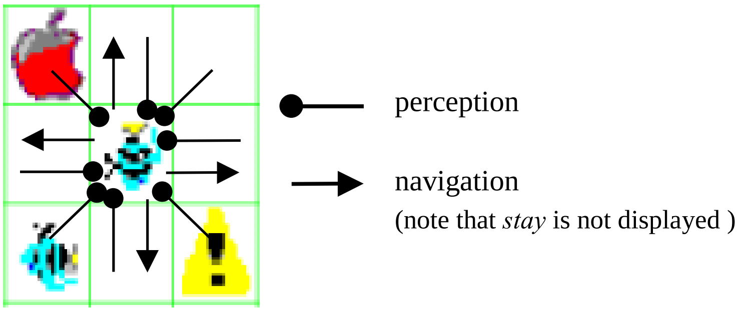

Fig. 3 Perceptual and navigational ability of the bugs.

Each bug is born with a certain amount of energy, which decreases in the navigation of the bug and increases only when the bug gets into a grid occupied by a food. In the latter case, the food is eaten by the bug and removed from the grid. If a bug tries to get into a grid occupied by a roadblock, then it will be rebounded and its energy will doubly decrease in this epoch. If the energy of a bug decreases to zero then the bug dies. The initial energy, the energy decreases in each epoch, and the energy increases for each food, can also be set when the environment is initialized.

Each bug is born with a gender, i.e. male or female. The proportion of bugs with different gender is set to a random value close to 0.5. If two bugs with same gender meet in a grid then they will fight each other. The result is that the stronger bug survives while the weaker one dies. This is realized in the way that for each bug, its energy after the fighting is computed through subtracting its opponent's energy from its own energy. If the resulted energy is not greater than zero then the bug dies. Otherwise the bug survives but is seriously hurt because its energy greatly decreases. On the other hand, if two bugs with different gender meet in a grid then they will mate. The result is that an offspring is generated, which is randomly put into an empty grid in the environment. The controlling organ of the offspring is inherited from its parents in a specific way to be explained in Section 3. Moreover, the weaker parent dies and the stronger parent survives with energy greatly decreasing, which looks like that they have fought each other. This could be viewed as a cost for reproduction.

The BUG system also realizes bugs controlled by random strategies where the bugs randomly pick a navigational operation in each epoch. So, there are in fact two kinds of bugs living in identical environments initialized at the same time, as shown in Fig. 4, where the right environment is of bugs controlled by the developmental and evolutionary CPM neural networks while the left one is of bugs controlled by random strategies.

3. The developmental and evolutionary CPMs

CPM is the abbreviation of *Coulomb Potential Model* [1], which is a relaxation neural network model minimizing an energy function for supervised classification. In this model, each training instance is regarded as a negative charge fixed in the space, and each test instance is regarded as a positive charge that could move in the space. A test charge, i.e. test instance, is attracted by the training charges, i.e. training instances, so that it moves in the electric field formed by the training charges until it is captured by a training charge. Then the class label of the training charge is assigned to the test charge.

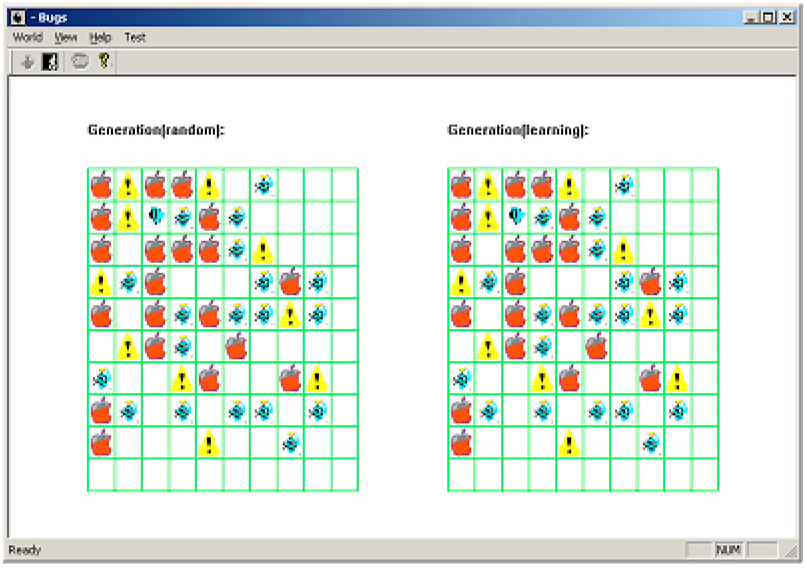

Fig. 4 Two kinds of bugs living in identical environments in the BUG system.

Suppose the magnitude of all the charges are 1.0, and the number of training charges is m, then the electric field strength on the test charge u can be computed according to Eq.(1), where x_i is the i-th training charge, ‖ … ‖ is Euclidian distance, L is a constant determining the property of the electric field (for electrostatic field, L = 2).

$$E_u = -\sum_{i=1}^{m} \left\| u - x_i \right\|^{-L} \frac{(u - x_i)}{\left\| u - x_i \right\|} \tag{1}$$

The test charge is attracted toward a training charge or a group of training charges. Its velocity and location can be computed according to Eq.(2) and Eq.(3) respectively, where λ is a step constant, *mindist* is the distance between the test charge and its nearest training charge, and ε is a small random number such as 1.0×10^{-12}.

$$v(n+1) = v(n) + \lambda \, mindist \, \frac{E_u}{\left\| E_u \right\|} \tag{2}$$

$$l(n+1) = l(n) + v(n) + (\varepsilon \, mindist) \tag{3}$$

The predicting process can be approximately shown in Fig. 5, where each training instance generates a basin of attraction, and the test instance is a freely moving ball. If the ball coming into a basin then it is assigned to the class of the training instance that generates the basin. It is worth mentioning that the test instance may not converge to its nearest training instance because it may be attracted toward a group of training instances with larger distance. So, CPM is similar to Bayesian classifier rather than nearest neighbor classifier.

In the BUG system described in Section 2, each bug is controlled by a CPM neural network. Each CPM network has ten input units. The first input unit represents the energy of the bug. The second one represents the navigational operation that the bug took in the latest epoch. The remaining eight units represent the occupancy of the neighboring

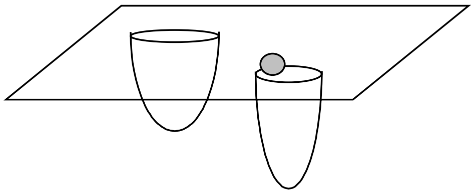

Fig. 5 Predicting process of CPM neural networks.

eight grids, i.e. whether the grids are occupied by food, roadblocks, or other bugs. Each CPM network outputs a decision for the next movement of the bug, that is, go one step up, down, left, right, or stay.

The hidden units of a CPM neural network are adaptively determined by the basins of attraction generated by the training set. In other words, there is no hidden unit before training, and the number of hidden units is growing when the training instances are gradually fed. It is obvious that the training process of the CPM neural networks is in an incremental style, which is valuable to the control of the bugs in the BUG system.

When a bug navigates the environment, it could memorize its perception, and after an epoch it could get a delayed reward for its decision, i.e. energy increased or decreased. Thus, the perception and the delayed reward constitute a training instance that could be used for the bug to tune its controlling organ to improve the chance of making good decisions in the future. When the training instance is fed to the CPM neural network, if it is appropriately captured by an established basin of attraction, then the basin may be strengthened to reflect a good decision or weakened to reflect a poor decision. Otherwise a new basin of attraction may be established to reflect a good decision. Therefore the controlling organs, i.e. the CPM neural networks, of the bugs are developmental throughout the lives of the bugs. This corresponds to the accumulation of experiences through learning during the life of the bugs.

Since the training speed of the CPM neural networks is very fast and the basins of attraction are determined by the training sets, each bug in the BUG system is designed to keep all the training instances that it collects during its life. These training instances are helpful to not only the developmental process described above but also the evolutionary process described as follows. When two bugs with different gender meet, an offspring is generated whose controlling organ is inherited from that of its parents. Different to prevailing styles where the architectures or the weights of the networks of the parents are passed down, here the training instances of the parents are genetically transmitted. In detail, for each parent a random subset is picked out from its training set, whose size is roughly half of that of the training set. Then, those two subsets are combined and used to train the CPM network of the offspring. We believe that the architectures or the weights of the CPM networks reflect the learned knowledge of the bugs, while the training instances reflect the experience accumulated during the lives of the bugs. So, in the evolutionary process, it is the experience of the parents instead of their knowledge is genetically passed down.

Note that we do not claim that genetically transmitting the experience is superior to transmitting the knowledge. In fact, in the past it is believed that evolution could be introduced into artificial neural networks at three levels, i.e. weight training, architecture adaptation, or learning rule [14]. Now we hope to append another level, i.e. training instances, and to explore in the virtual environment of the BUG system that whether evolution introduced into artificial neural networks at this level works well or not.

It is also worth mentioning that in order to realize the developmental and evolutionary CPM neural networks, the entire training set of each neural network must be kept, which may result in considerable storage cost. Moreover, when using CPM networks to predict test instances, since the distances between the test instance and all the training instances must be computed, the required computational cost may not be overlooked. However, since the virtual environment in the BUG system is quite simple, the size of the training set and the computational cost are not problems for the quick navigation of the bugs, which is the fact at least in our simulations.

4. Simulations

As shown in Fig. 4, besides the bugs controlled by the developmental and evolutionary CPM neural networks, the BUG system also realizes bugs controlled by random strategies, which live in an identical virtual environment. In this section, we report on the simulations of comparing the longevity of those two kinds of bugs.

We have performed simulations on twenty kinds of virtual environments with different sizes, different number of bugs, and different amount of food. In detail, the simulations are performed on environments with the size ranging from 10×10, 15×15, 20×20, 25×25, to 30×30; with 10% or 20% grids occupied by the bugs at the beginning; and with 10% or 20% grids occupied by the food at the beginning. For each kind of environment, we perform 100 runs and record the percentage of the runs where bugs controlled by the developmental and evolutionary CPM neural networks *win*, *lose*, or *tie*. For each run, if a bug controlled by the developmental and evolutionary CPM network lives the longest life then the percentage of *win* is increased by one point; if a bug controlled by random strategy lives the longest life then the percentage of *lose* is increased by one point; otherwise the percentage of *tie* is increased by one point. In all runs, 10% of the grids are occupied by roadblocks, the amount of food is doubled in every 30 epochs, and the bugs are born with 100 units of energy which is decreased in one units in each epoch and is increased for two units for each food.

Table 1 Simulation results in different kinds of virtual environments.

size of environment	*bug : food*	*win*	*tie*	*lose*
10×10	10% : 10%	87%	2%	11%
10×10	10% : 20%	40%	1%	59%
10×10	20% : 10%	93%	1%	6%
10×10	20% : 20%	77%	2%	21%
15×15	10% : 10%	90%	1%	9%
15×15	10% : 20%	81%	2%	17%
15×15	20% : 10%	93%	0%	7%
15×15	20% : 20%	77%	1%	22%
20×20	10% : 10%	88%	2%	10%
20×20	10% : 20%	94%	0%	6%
20×20	20% : 10%	93%	0%	7%
20×20	20% : 20%	72%	1%	27%
25×25	10% : 10%	94%	1%	5%
25×25	10% : 20%	91%	3%	6%
25×25	20% : 10%	93%	4%	3%
25×25	20% : 20%	54%	0%	46%
30×30	10% : 10%	90%	2%	8%
30×30	10% : 20%	93%	1%	6%
30×30	20% : 10%	95%	1%	4%
30×30	20% : 20%	43%	0%	57%

Table 1 shows that the bugs controlled by the developmental and evolutionary CPMs almost always live a longer life than those controlled by random strategies except on two environments. This reveals that the developmental and evolutionary CPM neural network is very valuable to the longevity of the bugs. We believe that these simulation results illustrate not only the importance of the interaction of the developmental and evolutionary processes to virtual creatures, but also the feasibility of introducing evolution at the level of training instances into artificial neural networks.

5. Conclusion

In this paper, virtual creatures navigating under the control of developmental and evolutionary artificial neural networks are presented. The developmental process is realized based on the incremental learning ability of CPM neural networks. The evolutionary process is realized through introducing evolution into the CPM neural networks at the level of training instances. Simulations show that the developmental and evolutionary CPM neural network is valuable to the longevity of the virtual creatures in the virtual environment. This exhibits not only the usefulness of the control of artificial neural networks to virtual creatures, but also the importance of the interaction of the developmental and evolutionary processes to virtual creatures.

Moreover, in the past it is believed that evolution could be introduced into artificial neural networks at three levels, i.e. weight training, architecture adaptation, or learning rule [14]. The success of our virtual creatures shows that, at least there is another level, i.e. training instances, for the evolution to be introduced into artificial neural networks. Note that in our virtual creatures such kind of evolution is only preliminarily utilized by saving all the training instances of the artificial neural networks, which may result in great storage cost for complicated tasks. However, we believe that other kinds of effective and efficient approaches could be developed based on the recognition that the evolution could be introduced at the level of training instances. One promising way for exploring is to store some *key* training instances instead of all the training instances for the artificial neural networks, which is an interesting issue for future works.

Furthermore, the controlling organs of our virtual creatures are only tuned by considering the decisions and the delayed rewards in successive two epochs in the developmental process. As Nolfi and Parisi [11] indicated, the behavior of artificial neural networks in an artificial life environment should be evaluated not in terms of single outputs but in terms of entire sequences of outputs. If a virtual creature controlled by artificial neural network is approaching food, although it is only the terminal action of reaching the food rewards the virtual creature, the succession of decisions that bring the virtual creature to the food should be given some sort of reward by this final action. Trying to incorporate such kind of temporal reward to the virtual creatures is another interesting issue for future works.

Acknowledgements

The National Natural Science Foundation of China and the Natural Science Foundation of Jiangsu Province, China, supported this research. The comments and suggestions from the anonymous reviewers greatly improve this paper.

References

[1] Bachmann C. M., Cooper L., Dembo A., and Zeitouni O. “A relaxation model for memory with high storage density.” *Proceedings of the National Academy of Sciences, USA*. 1987, vol.84, no.21, pp.7529-7531.

[2] Cangelosi A. and Parisi D. “The emergence of a ‘language’ in an evolving population of neural networks.” *Connection Science*. 1998, vol.10, no.2, pp.83-97.

[3] Collins R. J. and Jefferson D. R. "AntFarm: towards simulated evolution." In: C. G. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen (Eds.), *Artificial Life* II, Vol. X of *SFI Studies in the Sciences of Complexity*. Redwood City, CA: Addison-Wesley, 1991, pp.579-601.
[4] Dyer M. G. "Toward synthesizing artificial neural networks that exhibit cooperative intelligent behaviors: some open issues in artificial life." In: C. G. Langton (Ed.), *Artificial Life: An Overview*. Cambridge, MA: MIT Press, 1995, pp.111-134.
[5] Floreano D. and Mondada F. "Evolutionary neurocontrollers for autonomous mobile robots." *Neural Networks*. 1998, vol.11, no.7-8, 1461-1478.
[6] Fullmer B. and Miikkulainen R. "Using marker-based genetic encoding of neural networks to evolve finite-state behaviour." In: *Proceedings of the 1st European Conference on Artificial Life*. Paris, France, 1991, pp.255-262.
[7] Kerce K. F. "Evolution of artificial neural networks grounded in virtual ecosystems." In: *Proceedings of the 7th Florida Artificial Intelligence Research Symposium*. Pensacola Beach, FL, 1994, pp. 1-5.
[8] Michel O. "An artificial life approach for the synthesis of autonomous agents." In: J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers (Eds.), *Lecture Notes in Computer Science 1063*. Berlin: Springer-Verlag, 1996, pp.220-231.
[9] Möller R., Maris M. and Lambrinos D. "A neural model of landmark navigation in insects." *Neurocomputing*. 1999, vol.26-27, pp.801-808.
[10] Nolfi S. and Parisi D. "Learning to adapt to changing environments in evolving neural networks." *Adaptive Behavior*. 1997, vol.5, no.1, pp.75-98.
[11] Nolfi S. and Parisi D. "Neural networks in an artificial life perspective." In: W. Gerstner, A. Germond, M. Hasler and J.-D. Nicoud (Eds.), *Lecture Notes in Computer Science 1327*. Berlin: Springer-Verlag, 1997, pp.733-738.
[12] Saunders G. M. and Pollack J. B. "The evolution of communication schemes over continuous channels." In: *Proceedings of the 4th International Conference on Simulation of Adaptive Behavior*. North Falmouth, MA, 1996, pp.580-589.
[13] Werner G. M. and Dyer M. G. "Evolution of herding behavior in artificial animals." In: *Proceedings of the 2nd International Conference on Simulation of Adaptive Behavior*. Honolulu, HI, 1993, pp.393-399.
[14] Yao X. and Liu Y. "Towards designing artificial neural networks by evolution." *Applied Mathematics and Computation*. 1998, vol.91, no.1, pp.83-90.

Zhi-Hua Zhou received the BSc, MSc and PhD degrees in computer science from Nanjing University, China, in 1996, 1998 and 2000, respectively, all with the highest honor. At present he is an associate professor and director of the AI Lab of the Computer Science & Technology Department, Nanjing University. His current research interests are in machine learning, neural computing, evolutionary computing, pattern recognition, and data mining. In these areas he has published over 30 technical papers in referred journals or conferences. In 1999 he won the Mircrosoft Fellowship Award. He has chaired the organizing committee of the 7th Chinese Workshop on Machine Learning, and served as program committee members for many conferences. He is on the editorial board of the journal *Artificial Intelligence in Medicine*. He is an executive committee member of Chinese Association of Artificial Intelligence (CAAI), chief secretary of the CAAI Machine Learning Council, and is a member of IEEE and IEEE Computer Society.

Xue-Hua Shen received the BSc degree in computer science from Nanjing University, China, in 1999. At present he is working toward his PhD degree in computer science at University of Illinois at Urbana-Champaign. His current research interests are in data management, information access, and machine learning.