

# Decision-making models for resource allocation in multi-population co-evolutionary genetic algorithms implemented on networks

Gong Dun-wei and Sun Xiao-yan

School of Information and Electronics Engineering,  
China University of Mining and Technology, Xuzhou, 221008, P. R. China

## Summary

It is quite necessary that optimizing complicated systems using multi-population coevolutionary genetic algorithms implemented on networks. But there is no effective way to allocate computational resources to improve the performance of algorithms. A decision-making model for resource allocation in multi-population coevolutionary genetic algorithms implemented on networks is established, considering the case that all the populations coevolve synchronously. In this model, the optimized objective is to minimize the maximal computational time for average one-step iteration of a computational node, and the constraint is that each evolutionary population is allocated to one computational node at most. A method for solving the presented model based on genetic algorithms is also given. The validity of the model is shown through comparative examples. The results obtained in this paper provide a feasible mode for resource allocation in multi-population coevolutionary genetic algorithms implemented on networks.

## Key words:

*genetic algorithms, coevolution, multi-population, implemented on networks, resource allocation.*

## 1. Introduction

Genetic algorithms (GAs) are population based global probabilistic search algorithms presented by professor Holland in 1975 inspired by biological evolutions and genetics, whose idea is that a population evolves continuously towards better solutions by applying genetic operators until optima are achieved [1]. GAs have been widely used in function optimization, parameter identification, robotic control, neural network learning, fuzzy logic systems, and many other areas because of their vivid features such as outstanding parallelity, generality, global optimization and manipulability et al. [2].

However, it is hard for traditional genetic algorithms to solve complicated optimization problems with large dimensional variables, large search areas, complicated objectives and multiple objectives because they are apt to premature convergence and have low search efficiency in evolutionary anaphase. It is efficient for multi-population coevolutionary genetic algorithms (MPCEGAs) to solve the above problems. There are many manners of cooperation, and island multi-population coevolutionary genetic

algorithms are considered in this paper. In this model, multiple evolutionary populations are generated in a certain manner in same search space. After a number of evolutionary generations (one calls the number of the above evolutionary generations populations' evolutionary period), optimal individuals of all populations are exchanged and new populations are formed. The evolution in the next populations' evolutionary period continues based on the obtained new populations until a termination condition is met [3]. Kamiura et al considered a MPCEGA with distributed environment scheme and applied it to solve multi-knapsack problems successfully. In this algorithm, different populations search for different regions by adopting different evolutionary schemes and parameters [4]. Tongchun et al presented a MPCEGA with parameter adaptation. After a number of evolutionary generations, besides of individuals' migration, parameters that control different populations' evolutions are adjusted based on evolutionary performance [5]. Two populations are involved in the algorithm proposed by Liu et al. One is a traditional niche population for global search, and the other is an elitist population for local search. Only elitist individuals of the niche population are immigrated to the elitist one. The algorithm was applied to a portfolio investment model successfully [6]. Li et al proposed a MPCEGA for multi-mode function optimization. In this algorithm, all populations' optima are saved in an archive, and the distances between the individuals of each population and the archive are considered. Individuals' fitness is adjusted based on niche technique so that one population's optima affect other populations' evolutions. The efficiency of the algorithm was validated by typical multi-mode function optimization [7]. The algorithms in [4]-[7] were implemented on a single computational node, which prolongs the algorithms' runtime.

For complicated optimization problems, one expects to obtain their optima in short time by using MPCEGAs. Because of complexity resulted from individuals' evaluation, migration and substitution, as well as limitation of a single node's computational capacity, it is often difficult for MPCEGAs to achieve optimal solutions when they are implemented on a single node. The fast developments and prevalent applications of computer networks establish

foundation for MPCEGAs implemented on networks. The long computational time will be conquered effectively if different populations evolve on different computational resources and coevolutionary information is transmitted by high-speed communication networks, therefore the performance of such algorithms will be improved and their application scope will be enlarged greatly. Veldhuizen et al presented a platform of MPCEGAs implemented on networks. The formula for the algorithms' runtime was given assuming homogeneous computational nodes and same computational task in different computational nodes [8]. It was not considered that how to allocate computational resources reasonably to keep the whole runtime short in heterogeneous paradigms, which is a resource allocation problem. It is important for resource allocation problems in MPCEGAs' implementation on networks. The rational resource allocation can reduce the runtime and improve the performance of algorithms greatly. Ripe conclusions on such problems are absent up to now. So, it's quite necessary to research the above problems.

Resource allocation problems of MPCEGAs implemented on networks are studied in this paper. A decision-making model for resource allocation is established, considering the case that all populations coevolve synchronously. In this model, the optimized objective is to minimize the maximal computational time for average one-step iteration of a computational node, and the constraint is that each evolutionary population is allocated to one computational node at most. A method for solving the presented model based on GAs is also given. The validity of the model is shown through comparative examples.

## 2. Decision-making Models for Resource Allocation in Multi-population Coevolutionary Genetic Algorithms Implemented on Networks

MPCEGAs' implementation on networks is more complicated compared with their implementation on a single computational node. A main reason is that factors to be considered increase greatly. The corresponding factors include system's framework and constitution for implementation, strategies for saving, accessing and exchanging coevolutionary information, design of computational node interface and implementation of each population's evolution et al. Among these, the focus is a resource allocation problem, which will be solved in this section. The purpose for establishing decision-making models in MPCEGAs is to allocate computational resource to all evolutionary populations so that expected performance indexes are optimal in the case that some constraints are met.

The following factors will be considered for establishing decision-making models.

- (1) The time for evolutionary populations to perform genetic operators, which is decided by genetic strategies. Different genetic operators and control parameters will result in different runtime.
- (2) The time for exchanging representative individuals from different populations, which varies with the number of coevolutionary populations. The larger the number of populations, the longer the time is.
- (3) The time for transmitting representative individuals in networks, which is related to network medium and transmission mechanism. Different network media and different computational resource positions will result in different communicating time for same information to be transmitted.
- (4) Capacities of computational resources, which vary with different configurations.

### 2.1 Assumptions

The following hypothesis and stipulations are made in order to establish decision-making models.

- (1) Let  $p_1, p_2, \dots, p_n$  be coevolutionary populations, whose population size is  $|p_1|, |p_2|, \dots, |p_n|$ , respectively.
- (2) Let  $r_0, r_1, \dots, r_m$  be available computational resources whose computational capacities are  $c_0, c_1, \dots, c_m$  respectively, where  $r_0$  is a supervisory node and the rests are computational nodes. The distances between a computational node and the supervisory node are  $d_1, d_2, \dots, d_m$ , respectively.
- (3) All populations exchange representative individuals synchronously after  $g$  generations. The time for exchanging is correlated with the number of evolutionary populations. It is assumed that the time for every exchange is  $t_e \cdot n$ .
- (4) It is assumed that the time for transmitting representative individuals is proportional to the distance between a node and the supervisory node. The time for transmission of unit distance is  $\tau$ .

### 2.2 Decision-making Models for Resource Allocation

There are many kinds of objectives for decision-making models in MPCEGAs implemented on networks. The ob-

jective here is chosen to minimize the maximal computational time for average one-step iteration of a computational node.

For all populations evolving synchronously on different computational nodes, it is meaningful to expect the maximal computational time minimal. In the case that the number of evolutionary generations is determined, the above objective is equivalent to minimize the maximal computational time for average one-step iteration of a computational node. Such objective is adopted in the following decision-making model.

Let  $(a_{ij})_{n \times m}$  be a decision variable that describes allocation of nodes.  $a_{ij} \in \{0,1\}$ ,  $i=1,2,\dots,n$ ,  $j=1,2,\dots,m$ , where  $n$  and  $m$  represent the number of coevolutionary populations and computational nodes respectively. If a population  $p_i$  is allocated to a computational node  $r_j$ , then let  $a_{ij} = 1$ , otherwise let  $a_{ij} = 0$ . Let  $t_j$  be the time for average one-step iteration of a computational node  $r_j$ , therefore,

$$t_j = \sum_{i=1}^n a_{ij} (\alpha_{ij} |p_i| \cdot c_j + \tau \cdot d_j / g + t_e \cdot n / g \cdot c_0) \quad (1)$$

Where  $\alpha_{ij}$  is a coefficient describing the time for one-step evolutionary of  $p_i$  on  $r_j$ . The objective of this paper is:

$$\min \max_{j \in \{1,2,\dots,m\}} \{t_j\}$$

For simplicity, one only considers a simple constraint case that each evolutionary population is allocated to one computational node at most. Of course, there may be some nodes without allocated evolutionary populations. In this case, one has

$$\sum_{j=1}^m a_{ij} = 1$$

Based on these, the decision-making model established in this subsection can be expressed as:

$$\min \max_{j \in \{1,2,\dots,m\}} \left\{ \sum_{i=1}^n a_{ij} (\alpha_{ij} |p_i| \cdot c_j + \tau \cdot d_j / g + t_e \cdot n / g \cdot c_0) \right\} \quad (2)$$

$$\text{s.t. } \sum_{j=1}^m a_{ij} = 1 \quad (3)$$

$$a_{ij} \in \{0,1\}, i = 1,2,\dots,n; j = 1,2,\dots,m$$

### 2.3 Explanations for Models

Some further explanations for the model established in subsection 2.2 will expound in detail in this subsection.

In formula (2), the sum item contains 3 parts. The first one  $\alpha_{ij} |p_i| \cdot c_j$  is termed as an item for a single population evolution, which represents the time for  $p_i$  to perform one-step genetic operators. The bigger  $|p_i|$ , the bigger the value of this item is.  $\alpha_{ij}$  is different for different genetic strategies or computational nodes. If both genetic strategies adopted by all populations and capacities of all computational nodes are same, then one has  $\alpha_{ij} = \alpha$ . The value of this item is also related to  $c_j$ .

The second part  $\tau \cdot d_j / g$  expresses the average time for representative individuals to transmit in network, which is called an item for network transmission. The longer the distance between  $r_j$  and  $r_0$ , as well as the more the frequency of exchanging representative individuals, the longer the time is. The value of this item is usually small because of fast information transmission of modern communication networks.

The third item  $t_e \cdot n / g \cdot c_0$  describes the average time for representative individuals' exchange, which is called a coevolutionary item. Different exchange methods result in different  $t_e$ . The more the number of coevolutionary populations, as well as the more the frequency of exchanging representative individuals, the longer the time is. Besides the above, the value of this part is also related to  $c_0$ .

The above three parts are summed up because there may be many evolutionary populations running on same computational node. Which populations evolve on  $r_j$  is determined by  $a_{ij}$ . The rational resource allocation is just performed through  $a_{ij}$  in this subsection.

An extreme case that  $m$  equals 1 is considered. In this case, there is only one computational node besides the supervisory node. All populations evolve on this computational node, but still exchange representative individuals on the supervisory node. The time for average one-step iteration of this node is:

$$t = \sum_{i=1}^n (\alpha_{ij} |p_i| \cdot c_j + \tau \cdot d_j / g + t_e \cdot n / g \cdot c_0) > \sum_{i=1}^n a_{ij} (\alpha_{ij} |p_i| \cdot c_j + \tau \cdot d_j / g + t_e \cdot n / g \cdot c_0) \quad (4)$$

It can be seen from formula (4) that less time will spend on each computational node in case that MPCEGAs perform on many computational nodes than that on a single computational node, and the same is true for the maximal time. The results well explain why adopting many computational nodes can save computational time.

## 2.4 Solutions of Decision-making Models

The model given in subsection 2.2 is a typical combinational optimization problem, hence corresponding methods such as traditional optimization techniques, intelligent computation, et al can be applied to solve it [9]. Because combinational optimization problems are NP-hard, it will take much time for traditional optimization techniques to solve them when the size of models (namely  $n$ ,  $m$ ) increases, whereas intelligent computational methods can reduce the computational time efficiently. GAs are typical intelligent computational methods, therefore one applies GAs to solve them.

When one applies simple genetic algorithms to solve the above model, binary encoding is adopted to encode a variable whose length is  $n \times m$ . Initial populations with certain size that satisfy formula (3) are produced randomly. Fitness with punishment items to evaluate evolutionary individuals is established based on formula (2) and (3). Proportional selection, one-point crossover and one-point mutation operators are utilized, and their probabilities are  $p_c$  and  $p_m$  respectively. Elitist strategies are adopted for individuals' replacement. The termination condition is that the number of evolutionary generations reaches to a set value. The algorithms were expounded in detail in [2].

## 3. Examples

In order to validate the efficiency of the presented model, a complicated system is optimized by a MPCEGA implemented on local network. The parameters' values in formula (2) are set as table 1. The computational capacity of a node is characterized with the time consumed for computing  $\pi$  until 0.52 million bits after radix point. Computational capacities of nodes and distances between computational nodes and the supervisory node are given in table 2.

The following three cases are considered. The first one is that all populations adopt same genetic strategies. The second one is that all populations are divided into many groups, and different groups adopt different genetic strategies. The last one is that different populations adopt different genetic strategies. They are expounded in detail as follows.

Table 1: Parameter settings

Parameters	Values
$\tau$	0.1
$g$	10
$t_e$	0.1
$n$	40
$m$	7

Table 2: Computational capacities and distances from supervisory node

Nodes	Computation Capacities	Distances from supervisory node
$r_0$	19	0
$r_1$	21	24
$r_2$	18	6
$r_3$	22	7
$r_4$	29	5
$r_5$	17	12
$r_6$	27	17
$r_7$	23	20

### 3.1 Case 1: All Populations with Same Genetic Strategies

In this subsection, all populations adopt same genetic strategies, namely, roulette selection, two-point crossover and two-point mutation. All populations' size is 300, namely  $|p_i| = |p| = 300$ . The coefficients for one-step evolutionary time of all populations are same on same computational node, because same genetic strategies are adopted for all populations, which means that  $\alpha_{ij} = \alpha_j, i = 1, 2, \dots, n; j = 1, 2, \dots, m$ , where  $\alpha_j$  is a coefficient for one-step evolutionary on  $r_j$ . Because  $\alpha_j$  is related to  $c_j$ , for not lost generality, it is assumed that a coefficient for one-step evolutionary of a population on  $r_0$  is  $\alpha_0$ . One has  $\alpha_{ij} = \frac{c_j}{c_0} \alpha_0$ . The formula (2) is transformed as:

$$\min_{j \in \{1, 2, \dots, m\}} \max_{i \in \{1, 2, \dots, n\}} \left\{ \sum_{i=1}^n a_{ij} \left( \frac{c_j}{c_0} \alpha_0 |p_i| \cdot c_j + \tau \cdot d_j / g + t_e \cdot n / g \cdot c_0 \right) \right\} \quad (5)$$

$\alpha_0$  is determined as follows. The above system is optimized on the supervisory node, and let  $t_0$  be runtime for a population to evolve  $T$  generations. Then  $\alpha_0 = \frac{t_0}{T}$ . The average value of  $\alpha_0$  for many times of running is taken and one has  $\alpha_0 = 0.0248$ .

The allocation of populations to nodes is shown in table 3 when the method presented in this paper is adopted. The corresponding maximal average one-step iteration time for the whole coevolution is 20.1164.

Table 3: Populations' allocation to nodes

Nodes	Serial Number	Population Numbers
$r_1$	4,14,19,24,34,38	6
$r_2$	6,7,8,9,10,13,29,36	8
$r_3$	3,20,22,26,31,32	6
$r_4$	16,35,37	3
$r_5$	5,11,15,17,23,25,27,28,30	9
$r_6$	2,39,40	3
$r_7$	1,12,18,21,33	5

Table 4: Populations' allocation to nodes

Nodes	Serial Number	Population Numbers
$r_1$	1,2,3,4,5	5
$r_2$	6,7,8,9,10,11,12,13,14	9
$r_3$	15,16,17,18,19,20,21	7
$r_4$	22,23,24,25,26,27	6
$r_5$	28,29,30,31,32,33	6
$r_6$	34,35,36	3
$r_7$	37,38,39,40	4

The maximal average one-step iteration time for the whole coevolution is 33.9758 based on the allocation shown in table 4. This scheme is given based on the capacities of all computational nodes and the distances between computational nodes and the supervisory node without utilizing the presented model in this paper.

### 3.2 Case 2: Division of Populations into Many Groups and Different Groups with Different Genetic Strategies

In this subsection, all populations are divided into four groups and different ones adopt different genetic strategies, such as different selection, crossover or mutation operators. Therefore, different groups' coefficients  $\alpha_j$ s for one-step evolutionary time are different on same computational node. The number of populations and their serial number are factors affecting the efficiency of algorithms. Assuming the size of all populations is same, and its value is 300. Among 40 populations, the former 20 populations adopt roulette selection, while the later 20 populations utilize tournament selection and the size of tournament is 2. One-point crossover and one-point mutation are adopted in the populations whose serial number is from 1 to 10. Two-point crossover and two-point mutation are adopted in the populations whose serial number is from 11 to 20. Three-point crossover and three-point mutation are for the populations whose serial number is from 21 to 30 and the last 10 populations utilize four-point crossover and four-point mutation. Four populations whose serial number is 1,11,21 and 31 perform on the supervisory node for several times

to gain their coefficients for average one-step evolutionary time. The results are shown in table 5. Other parameters are same as those in subsection 3.1.

Table 5: Populations' coefficients for average one-step evolutionary time

Serial Number	Average one-step evolutionary time coefficients
1	0.0109
11	0.0248
21	0.0565
31	0.0823

The allocation scheme given in table 6 is gained by optimizing the presented model, and the corresponding maximal average one-step iteration time for the whole coevolution is 32.2526.

Table 6: Populations' allocation to nodes

Nodes	Serial Number	Population Numbers
$r_1$	3,7,9;12,16,19;27;40	3+3+1+1=8
$r_2$	13;24,26,28;32,35	0+1+3+2=6
$r_3$	2,10;15;22,25;36	2+1+2+1=6
$r_4$	5;11,14,18;29	1+3+1+0=5
$r_5$	6;21;31,33,34,38	1+1+4+0=6
$r_6$	1,4,8;20;37	3+1+1=5
$r_7$	17;23,30;39	0+1+2+1=4

The maximal average one-step iteration time for the whole coevolution is 35.3820 if an allocation scheme shown in table 7 is used.

Table 7: Populations' allocation to nodes

Nodes	Serial Number	Population Numbers
$r_1$	1;21,22;31,32	1+0+2+2=5
$r_2$	11,12,13,14;23,24,25;33	0+4+3+1=8
$r_3$	15;34,35,36	0+1+0+3=4
$r_4$	2,3;16;37	2+1+0+1=4
$r_5$	4,5,6;17,18,19;26;38,39	3+3+1+2=9
$r_6$	7,8,9,10;27,28	4+0+2+0=6
$r_7$	20;29,30;40	0+1+2+1=4

### 3.3 Different Populations with Different Genetic Strategies

In this part, the case that all populations use different genetic strategies is considered. The population size of different populations differs from each other, which is given as follows according to serial number.

165,108,330,175,16,84,82,220,311,490,  
177,440,104,361,262,382,449,142,206,18,  
420,237,239,202,212,440,60,142,369,223,  
84,310,270,81,262,327,341,253,91,122.

All populations' coefficients for one-step evolutionary time are as follows:

0.0493,0.0387,0.0692,0.0194,0.0035,  
 0.0173,0.0184,0.0322,0.055,0.0273,  
 0.0245,0.0251,0.0314,0.0132,0.0266,  
 0.0226,0.0415,0.0390,0.0255,0.0263,  
 0.0298,0.0128,0.0094,0.0124,0.0192,  
 0.0187,0.0454,0.0474,0.0199,0.0276,  
 0.0301,0.0161,0.0166,0.0244,0.0172,  
 0.0253,0.0246,0.0157,0.0406,0.0283.

Other parameters are same as those in subsection 3.1. The allocation scheme shown in table 8 is obtained by optimizing the presented model, and the corresponding maximal average one-step iteration time for the whole coevolution is 16.4065.

Table 8: Populations' allocation to nodes

Nodes	Serial Number	Population Numbers
$r_1$	6,11,12,16,37	5
$r_2$	3,5,7,14,25,30,32,33,34	9
$r_3$	4,10,13,15,18,24	6
$r_4$	8,22,27,31,38	5
$r_5$	1,17,20,21,23,26,28	7
$r_6$	29,35,36,39	4
$r_7$	2,9,19,40	4

It is difficult to determine which allocation scheme will minimize the maximal average one-step iteration time for the whole coevolution because different populations adopt different evolutionary strategies and population size. Therefore the efficiency of the algorithm presented in this paper is illuminated by compared with the outcome of a random allocation scheme. The maximal average one-step iteration time for the whole coevolution is 20.4817 in case of an allocation scheme shown in table 9, which is much larger than 16.4065.

Table 9: Populations' allocation to nodes

Nodes	Serial Number	Population Numbers
$r_1$	1,2,3,4,5	5
$r_2$	6,7,8,9,10,11,12,13,14	9
$r_3$	15,16,17,18,19,20	6
$r_4$	21,22,23,24,25	5
$r_5$	26,27,28,29,30,31,32	7
$r_6$	33,34,35,36	4
$r_7$	37,38,39,40	4

It can be seen from the above 3 cases that the third one is the most difficult to determine the allocation scheme, whereas the maximal average one-step iteration time of the whole coevolution is the least. The reason is that different evolutionary strategies and different population size adopted by different populations result in different one-

step evolutionary time. It is shown from formula (2) that  $\alpha_{ij}|p_i| \cdot c_j$  will be small if all populations are allocated correctly. Based on these, one can get shorter evolutionary time than the former two cases. It can be concluded from the result that all populations utilizing different evolutionary strategies will result in short evolutionary time when topology frameworks of algorithms, computational capabilities of nodes, and migration strategies are determined. Therefore, when MPCEGAs are applied, in order to improve their efficiency, different evolutionary strategies should be adopted by different populations before allocation optimization. In addition, when MPCEGAs implemented on networks are utilized to optimize complicated systems, it is informed from the comparative results that rational allocation of populations to computational nodes is very efficient for improving algorithms' performance and reducing algorithms' runtime.

### 4. Conclusions

In order to improve the performance of MPCEGAs implemented on networks, a decision-making model for computational resource allocation is established in this paper. A method for solving the presented model based on GAs is also given. Finally, examples are detailed. The results show that the model established in this paper is able to allocate computational resource reasonably. Based on the optimized allocation scheme, the runtime for coevolutionary populations to evolve on computational nodes reduces greatly. The outcomes are also suitable for other multi-population coevolutionary algorithms.

It should be declared that only one objective, namely minimization of the maximal computational time for average one-step iteration of a computational node, is considered in this paper. Other objectives can also be considered during establishing decision-making models, and hence corresponding multi-objective decision-making models can be obtained. Good results are expected to achieve based on such models, which are the issues to be further researched.

### Acknowledgments

The authors would like to acknowledge National Natural Science Foundation of China for its financial support under Grant No. 60304016 and 60575046.

## References

- [1] J. H. Holland, *Adaptation in natural and artificial systems*, Ann Arbor: The University of Michigan Press, Michigan, 1975.
- [2] D. E. Goldberg, *Genetic algorithms in search, optimization & machine learning*, Addison-Wesley, 1989.
- [3] P. Siarry, A. Petrowski, M. Bessaou, A multi-population genetic algorithm aimed at multi-modal optimization, *Advances in Engineering Software*, 2002, 33: 207-213.
- [4] J. Kamiura, T. Hiroyasu, M. Miki, et al, MOGADES: Multi-objective genetic algorithm with distributed environment scheme, *Proceedings of the 2nd International Workshop on Intelligent Systems Design and Applications*, 2002:143-148.
- [5] S. Tongchim, P. Chongstitvatana, Parallel genetic algorithm with parameter adaptation, *Information Processing Letters*, 2002, 82: 47-54.
- [6] H. J. Liu, Z. B. Wang, Improved multi-mode genetic algorithms and their application on portfolio investment model, *Control and Decision*, 2003,18(2): 173-176.
- [7] M. Q. Li, J. S. Kou, Coordinate multi-population genetic algorithms for multi-mode function optimization, *Acta Automatic Sinica*, 2002, 28(4): 497-504.
- [8] D. A. V. Veldhuizen, J. B. Zydallis and G. B. Lamont, Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms, *IEEE Trans. on Evolutionary Computation*, 2003, 7(2): 144-173.
- [9] N. K. Bambha, S. S. Bhattacharyya, J. Teich, et al, Systematic integration of parameterized local search into evolutionary algorithms, *IEEE Trans. on Evolutionary Computation*, 2004, 8(2): 137-155.



**Sun Xiao-yan** received the B.S. in automation from China University of Mining and Technology, Xuzhou, P.R.China, in 2001, M.S. in control theory and control engineering, China University of Mining and Technology, Xuzhou, P.R.China, in 2005, and now she is a Ph D. student in control theory and control engineering of China University of Mining and Technology, Xuzhou, P.R.China. Since 2001, she has been an assistant in School of Information and Electrical Engineering, China University of Mining and Technology. Her main research interest is co-evolutionary computation. She has authored and co-authored more than 10 papers.



**Gong Dun-wei** received the B.S. in applied mathematics from China University of Mining and Technology, Xuzhou, P.R.China, in 1992, M.S. in control theory and its applications from Beijing University of Aeronautics and Astronautics, Beijing, P.R.China, in 1995, and Ph.D. in control theory and control engineering from China University of Mining and Technology, Xuzhou, P.R.China, in 1999. From 1995 to 2002, he has been a research associate in School of Information and Electrical Engineering, China University of Mining and Technology. Since 2003, he takes up the director of the Center for Intelligent System and Intelligent Control, China University of Mining and Technology. His main research interests are evolutionary computation and intelligent control.