Hybrid Multipopulation Cellular Genetic Algorithm and Its Performance

Li Ming(黎明)[1,2], *Lu Yuming*(鲁宇明)[1,2*], *Jie Lilin*(揭丽琳)[1]

1. Key Laboratory of Nondestructive Testing, Ministry of Education, Nanchang Hangkong University, Nanchang, 330063, P. R. China;
2. College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, P. R. China

(Received 11 October 2013; revised 19 December 2013; accepted 23 December 2013)

Abstract: The selection pressure of genetic algorithm reveals the degree of balance between the global exploration and local optimization. A novel algorithm called the hybrid multi-population cellular genetic algorithm (HCGA) is proposed, which combines population segmentation with particle swarm optimization (PSO). The control parameters are the number of individuals in the population and the number of subpopulations. By varying these control parameters, changes in selection pressure can be investigated. Population division is found to reduce the selection pressure. In particular, low selection pressure emerges in small and highly divided populations. Besides, slight or mild selection pressure reduces the convergence speed, and thus a new mutation operator accelerates the system. HPCGA is tested in the optimization of four typical functions and the results are compared with those of the conventional cellular genetic algorithm. HPCGA is found to significantly improve global convergence rate, convergence speed and stability. Population diversity is also investigated by HPCGA. Appropriate numbers of subpopulations not only achieve a better tradeoff between global exploration and local exploitation, but also greatly improve the optimization performance of HPCGA. It is concluded that HPCGA can elucidate the scientific basis for selecting the efficient numbers of subpopulations.

Key words: cellular genetic algorithm; particle swarm optimization; multispecies; selection pressure; diversity

CLC number: O321 **Document code**: A **Article ID**: 1005-1120(2014)04-0405-08

1 Introduction

Intelligent algorithms proposed in recent years are grounded in various biological phenomena and laws. These intelligent algorithms are widely used to solve optimization problems in science and engineering. In practice, however, these optimization problems by themselves are inadequate for solving complex problems, and the results are often deficient. Therefore hybrid algorithms, which combine the desirable features of different algorithms, have attracted much interest.

The cellular genetic algorithm (CGA) is a type of decentralized GA in which each individual is fixed in a tutorial grid, usually of dimension 2, regardless of parallel execution. Genetic operators are applied locally to the neighborhood of each individual, which enables slow diffusion of favorable individuals. While CGA encourages diversity in the population, it can delay the convergence speed of the algorithm. The CGA exhibits higher global exportation ability than GA, but converges more slowly.

Particle swarm optimization (PSO), an optimized algorithm based on swarm intelligence, simulates the social behavior of cooperative groups such as ants, fishes and birds. The swarm

Foundation items: Supported by National Natural Science Foundation of China (61262019); the Aeronautical Science Foundation of China (2012ZA56001); the Natural Science Foundation of Jiangxi Province (20114BAB201046); the Science and Technology Research Project of Jiangxi Provincial Department of Education (GJJ12435); and the Open-End Foundation of the Key Laboratory of Nondestructive Testing (Ministry of Education).

* **Corresponding author**: Lu Yuming, Assosiate Professor, E-mail: luyuming69@163.com.

develops a collective intelligence that facilitates its search for a global optimum. Desirable features of the PSO algorithm are simple rules, few parameters and rapid convergence speed. However, global search ability of the algorithm is poor.

The evolutionary rules of cellular automata have been extensively documented [1-2]. An extension of cellular automata, namely, genetic algorithms with evolutionary rules can improve population diversity. Li, et al[3] analyzed the convergence rate of canonical CGA using absorbing-state Markov chain. Many widely-used neighbor structures have been analyzed and researched in detail [4-6]. Spatial states with a cell having four different types of neighbors were simulated and the effects of each neighbor were analyzed [7]. Some algorithms introduced disastrous events into CGA[8-10] and proposed a hierarchical CGA, while a hybrid CGA/distribution estimation algorithm was proposed in Ref. [11]. Hybrid algorithms combining GA with local searching proved effective in solving multi-objective optimization problems[12-13]. From the above citations, it is apparent that improving the local search ability and convergence speed of CGA was neglected.

The GA is based on the tradeoff between global exploration and local exploitation, which reflects selection pressure. Refs. [14-16] investigated the selection pressure of CGA on neighborhood structure, breeding strategies and selecting operation. The selection pressure imposed by CGA with disaster on size and period of disasters is also investigated [17]. Selection pressure was found to be lower following a large disaster, and to occur over a shorter time period. Ref. [18] proposed a new adaptive algorithm that aims to dynamically control the exploration/exploitation trade-off, based on three-dimensional CGAs. According to their results, selection pressure varied if certain parameters were varied. This finding provides valuable insights into the tradeoff between global exploration and local exploitation.

Recognizing that PSO possesses strong local searching ability, this paper proposes a hybrid multipopulation cellular genetic algorithm (HCGA) that combines GA with PSO. The performance of the algorithm is evaluated on four typical test functions. Selection pressure and population diversity are assessed by varying the population size and the number of subpopulations. We demonstrate the superiority of HCGA in terms of global convergence rate and convergence speed. The algorithm operates most effectively when the number of subpopulation is $n=2^m (m=3)$.

2 Description of Cellular Genetic Algorithm and Particle Swarm Optimization

2.1 Cellular genetic algorithm description

In CGAs, individuals are placed on a toroidal d-dimensional grid (the algorithm is usually implemented in two dimensions). Each occupied grid element (or cell) contains a single individual. Genetic reproduction and crossover can occur only between an individual and its nearest neighbors (see Fig. 1).

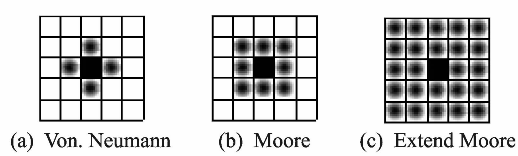


Fig. 1 Structure of a neighborhood

We adopt the CGA presented in Ref. [2]. In the CGA, individuals are randomly classified as "active" or "inactive" (see Fig. 2). Under an evaluative rule, all individuals simultaneously change state. An "active" cell is the one that can interact with its neighborhood to select and crossover.

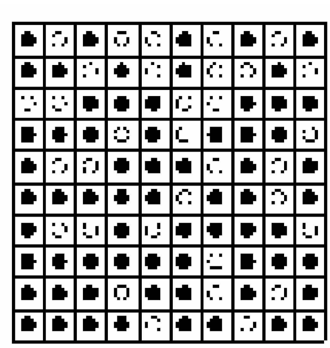


Fig. 2 Distribution of individuals in CGA

The pseudo-code of the CGA algorithm is provided below:

Step 1 To classify an individual as living or dead on the $L \times L$ grid at random.

Step 2 To set the stop condition.

Step 3 To calculate fitness of individuals.

Step 4 To select the living individual and obtain its neighborhood as parents.

Step 5 To implement parents' recombination.

Step 6 To evaluate fitness and replace existing individual if fitness is improved.

Step 7 To implement individual mutation.

Step 8 To update states synchronously according to evolution rule.

Step 9 When the stop condition is satisfied, end.

In the above algorithm, the current population is replaced after synchronously applying crossover and mutation to all individuals.

2.2 Particle swarm optimization description

The PSO searches a global optimum by simulating movement and interaction of swarming particles. A population of particles is initialized with random position and velocities. The position of a particle corresponds to one possible solution of the problem. The objective value of each particle is computed by an objective function. In the next iteration, the position and velocity of each particle is updated through tracking its own experience and that of other particles.

3 Hybrid Multipopulation Cellular Genetic Algorithm

3.1 Population division and immigration of individuals

Population diversity can be maintained by dividing the population into several equally-proportioned subpopulations that do not depend on each other. Each subpopulation evolves independently, i.e., genetic operations cannot occur between subpopulations.

Population division usually causes isolated islands that cannot interact with other islands. To enable information exchange between subpopulations, one or a few reproductive individuals in a subpopulation are allowed to immigrate to another island according to the immigration rate when the interval generation ΔT meets a specified value. Here in this paper, ΔT is 20.

3.2 Construction of new operations

The existing CGA imposes random mutations that are irrelevant to past and present individual states, thereby ignoring the distance between each individual and the fittest individual. Furthermore, excessively high mutation rates will destroy favorable genes, while low rates will reduce the search speed. Very low rates will stagnate the evolutionary process. In addition, since mutation is directional, the probability of low fitness will be increased.

In this study, mutation in CGA (Step 7) is replaced by a new operation based on neighboring structures in PSO. Following the operation, the individual in the next iteration is calculated as

$$x_{i(t+1)} = x_{it} + v_{i(t+1)} \tag{1}$$

where t is the generation, n the population size, i the position order of the individual in the cell space, and x_{it} the gene of the ith individual. The population is denoted as $Q_t = \{x_{1t}, x_{2t}, \cdots, x_{it}, \cdots, x_{nt}\}(1 \leqslant i \leqslant n)$, and $v_{i(t+1)}$ is calculated as

$$v_{i(t+1)} = w \cdot v_{it} + c_1 \cdot r_1 \cdot (x_{i\max}^{\text{History}} - x_{it}) + c_2 \cdot r_2 \cdot (x_{i\max}^{\text{Neighbor}} - x_{it}) \tag{2}$$

where w is the inertia coefficient, $x_{i\max}^{\text{History}}$ the fittest gene acquired by an individual, and $x_{i\max}^{\text{Neighbor}}$ the fittest neighboring gene identified by the individual. r_1 and r_2 are the uniformly distributed random numbers in the interval $[0,1]$. c_1, c_2 are the cognitive and social learning factors, respectively. v_{it} is the mutating velocity at generation t, calculated as

$$v_{it} = \left[\sum_{k=2}^{t} (x_{i\max}^{\text{History}}(k) - x_{i\max}^{\text{History}}(k-1))\right] / t \tag{3}$$

Since Eq. (2) uses amplitude and directional information to forecast mutation of an individual, it improves the local searching ability, and eliminates the indiscriminate mutating operations that occur in CGA.

4 Computational Experiments

4.1 Test problems

The algorithm is evaluated on four test functions, as summarized below:

(1) F_1 Schaffer's $f6$ function

$$f(x,y)=0.5-\frac{(\sin\sqrt{x^2+y^2})^2-0.5}{1+0.001(x^2+y^2)^2}$$
$$-10\leqslant x,y\leqslant 10 \quad (4)$$

Eq. (4) has a single maximum at 1. This global optimum is surrounded by a few local optima, including one at 0.990 284 and another at 0.962 776. Implemented on F_1, most algorithms easily reach a local optimum from which they cannot escape.

(2) F_2 Needle function

$$f(x,y)=\frac{\sin(\sqrt{(x-50)^2+(y-50)^2}+2.718\ 28)}{\sqrt{(x-50)^2+(y-50)^2}+2.718\ 28}+1$$
$$0\leqslant x,y\leqslant 100 \quad (5)$$

Function F_2 is similar to F_1. One of its local maxima (at 1.128 4) is extremely close to the global maximum (at 1.151 1). Most algorithms reach the local optimum at 1.128 4.

(3) F_3 Griewank's function

$$f(x_1,x_2,\cdots,x_n)=\frac{1}{4\ 000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$$
$$-600\leqslant x_i\leqslant 600 \quad (6)$$

This paper adopts $30-d$. Function F_3, which is a multimodal function with a single global optimum surrounded by many nearby local optima.

(4) F_4 Sphere function

$$f(x_1,x_2,\cdots,x_n)=\sum_{i=1}^{n}x_i^2,\ -100\leqslant x_i\leqslant 100 \quad (7)$$

F_4 is a unimodal function with a minimum of 0 at (0, 0, $\cdots$, 0). Its dimension is the same as F_3. High-dimensional versions of this function are more difficult to solve because of the strong constraints between variables.

4.2 Parameter setting

The parameters are as follows: number of runs is 100, cellular space size 20×20, population size 400, crossing rate 0.8, mutation 0.05. In HCGA, learning factors c_1 and c_2 are both set to be 2. Immigration rate is 0.2 and the inertial weight is 1.

5 Experimental Results

5.1 Analysis of selection pressure

To some extent, selection pressure represents the balance between exploration and exploitation. Selection pressure is measured by the takeover time [3], defined as the required time for a single (best) individual to occupy the entire population using the selection operator only, and ignoring crossovers and mutation. The shorter the takeover time, the higher the selection pressure.

Fig. 3 plots the proportion of the best individual in the population as a function of time in CGA. Fig. 4 is an equivalent plot generated by HCGA, but varying the subpopulation number and population size.

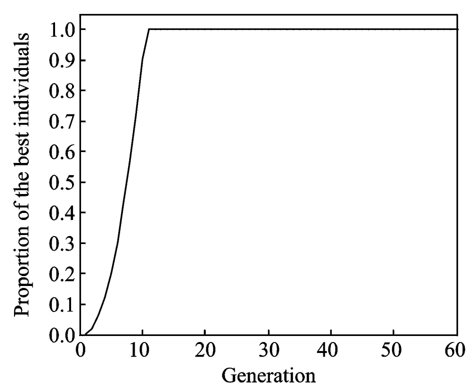
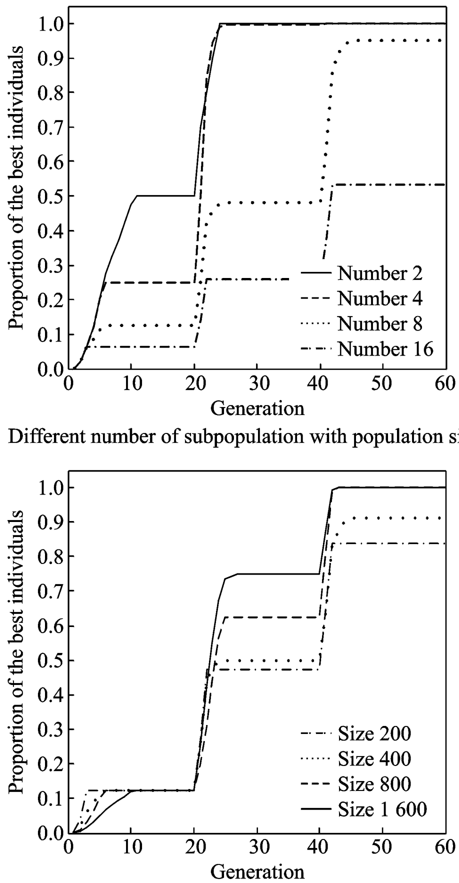


Fig. 3 Growth curve of the best individual (CGA)

In Fig. 3, the curve gradually ascends to 1 and remains constant thereafter. When the proportion of the best individual reaches 1, information of the best individual cannot be spread. Then the selection pressure demonstrates the saturated condition.

The curve of Fig. 4 similarly ascends but less smoothly. The proportion of the best individuals firstly gradually ascends to $1/n$ before 10 generations, then stays stable for a period of time and goes up after 20 generations. Furthermore, the similar curve jump can be observed in the later evolution, such as 40 generations. The jumps are observed when $\Delta T=20$. The phenomenon is caused by individual migration strategy, which



(a) Different number of subpopulation with population size of 400

(b) Different population size with number of sub-population of 8

Fig. 4 Growth curve of the best individual (HCGA)

provides potential for the best individual information in a subpopulation exchanging into other subpopulations. Information is thus disseminated between subpopulations. In this way, a fit individual can spread its genes into other subpopulations, and thereby spread more widely. But when the proportion of the best individuals is 1, the best individual cannot spread.

Varying subpopulation numbers The population size is retained at 20 × 20 and the subpopulation number is set to 2, 4, 8 and 16. The resulting selection pressure is displayed in Fig. 4(a). In Fig. 4(a), the proportion of best individuals in the population increases more slowly when more subpopulations exist. Namely, the proportion of fittest individuals decreases as subpopulation number increases; equivalently, the selection pressure decreases as the number of subpopulations increases.

Varying population size Retaining the subpopulation number at 8, the population size is set to 200, 400, 800 and 1 600, respectively. The results are plotted in Fig. 4(b). From Fig. 4(b), we observe that selection pressure decreases as population size increases, up to the 10th generation. Between generations 10 and 20, it is relatively constant, because the fittest individual is not spread until the conditions favor migration. Beyond the 20th generation, selection pressure again increases with population size.

The above analysis reveals that by segmenting the population, HCGA reduces the selection pressure relative to CGA, and improves the global convergence of the algorithm.

5.2 Performance of HCGA and CGA

HCGA is compared with CGA with respect to global convergence rate (P), average convergence generation (G), average run time (T), and the average and standard deviation (STD) of the best value.

The results implemented on F_1—F_4 are shown in Table 1. The global convergence of HCGA on F_1 and F_2 is 100% and the algorithm never becomes trapped in local optima. The convergence generation of HCGA is lower than CGA and the algorithm converges more quickly. Especially on F_2, CGA converges to the global optimum in only 17% of trials, and its convergence speed is three times slower than that of HCGA. On F_3, CGA never converges to the global optimum, while HCGA converges in 100% of trials. On F_4, although both algorithms converge 100% of the time, the convergence speed of HCGA far exceeds that of CGA. The convergence rate of HCGA is attributed to the population segmentation and individual migration, which reduces selection pressure, slows down information dissemination and avoids premature convergence. Moveover, the new operation is directional, and the convergence speed is thus improved.

5.3 Performance under varying population segment number

The number of subpopulations is an important parameter in HCGA. This section compares the algorithm performance for different subpopulations n, where $n=2^m(m=1,2,3,4)$. Population size is retained constant at 400. The other parameters are as specified in Section 4.2.

Table 2 compares the global convergence rate (P), average convergence generation (G), average run time (t), and average and the standard deviation (STD) of the best value.

The larger the number of subpopulations, the lower the selection pressure (see Fig. 4). Hence, on each of the four test functions, the global convergence rate increases as the number of subpopulations increases. Initially, the spending time decreases as the number of subpopulations increases and later increases, except on F_1. When the number of subpopulations is too large, the selection pressure will be too low. It is unfavorable for information dissemination, which can reduce the performance of HPCGA. The STD of the fittest individual is also improved as subpopulation increases, and later decreases.

Table 1 Comparison of performance in terms of HCGA and CGA

Function		F_1	F_2	F_3	F_4
P/%	CGA	85	17	0	100
(Accuracy<10^{-4})	HCGA	100	100	100	100
G	CGA	148.3	4 075.1		6 835.2
	HCGA	134.7	1 465.7	2 606.8	2 068.3
T/s	CGA	7.809 2	265.0744		137.125 2
	HCGA	7.855 7	74.362 2	95.590 7	45.406 2
Average value	CGA	0.999 356 03	1.135 146 71	0.049 831 54	5.476 9E−05
	HCGA	0.999 945 86	1.151 040 78	5.760 6E−05	5.476 6E−05
STD	CGA	5.257 6E−06	1.070 2E−04	2.553 3E−04	6.890 0E−09
	HCGA	6.548 4E−10	7.888 8E−10	7.412 1E−10	6.743 9E−10

Table 2 Comparisons of performance of different numbers with sub-population

Function	m	P/% (Accuracy<10^{-4})	G	T/s	Average value	STD
F_1	1	93	115.5	5.565 0	0.999 759 65	1.833 1E−06
	2	98	124.0	5.949 7	0.999 828 58	9.740 8E−07
	3	100	134.7	7.855 7	0.999 945 86	6.548 4E−10
	4	100	162.4	10.053 0	0.999 943 78	9.071 7E−10
F_2	1	73	1 459.6	163.114 0	1.144 919 70	1.012 5E−04
	2	89	1 562.9	123.486 6	1.148 544 79	5.367 2E−08
	3	100	1 465.7	74.362 2	1.151 040 78	7.412 1E−10
	4	100	1 466.7	81.173 5	1.151 040 49	7.998 4E−10
F_3	1	66	6 435.3	226.709 4	0.001 966 32	3.951 9E−05
	2	98	3 455.7	128.186 7	8.995 5E−05	5.367 2E−08
	3	100	2 606.8	95.590 7	5.760 6E−05	7.412 1E−10
	4	100	2 859.3	111.166 4	5.777 7E−05	7.998 4E−10
F_4	1	82	5 482.3	113.144 6	8.102 2E−05	3.274 5E−09
	2	100	2 543.0	51.900 6	6.041 8E−05	8.063 7E−10
	3	100	2 068.3	45.406 2	5.476 6E−05	6.743 9E−10
	4	100	2 052.8	47.987 4	5.534 5E−05	7.296 8E−10

5.4 Diversity performance

Population diversity is crucial in evolutionary algorithms. Only in a diverse population can the algorithm seek a global optimum. Therefore, maintaining population diversity is guaranteed to improve algorithm performance. This section investigates changes in diversity over time, while varying the number of subpopulations. Diversity is measured as the ratio of population entropy to the maximum of population entropy[17].

The CGA and HCGA algorithms are implemented 100 times on F_1 and F_3. Figs. 5, 6 plot the evolution of diversity calculated by CGA and HCGA, respectively, for the four population segmentation numbers. The population diversity drops dramatically with the increasing generation in CGA. On F_1 and F_4, population diversity is very low at generations 500 and 1 000, respectively. However, in HCGA, the population diversity declines slowly and maintains high over a long period. Most importantly, population diversity is strengthened as subpopulation number increases, up to $m=3$. When $m=3$ and 4, the algorithm can keep population diversity better than that of the case with $m=1$ or 2.

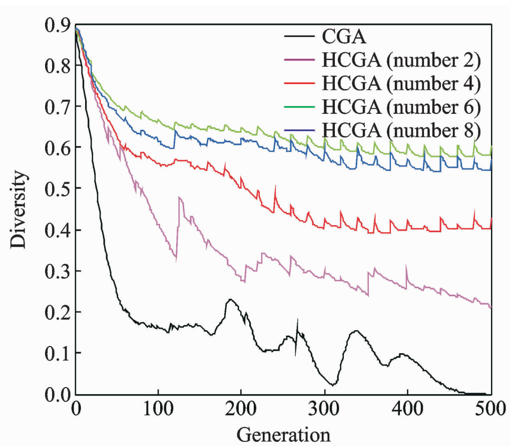


Fig. 5 Diversity change of F_1 on HCGA with different subpopulation numbers

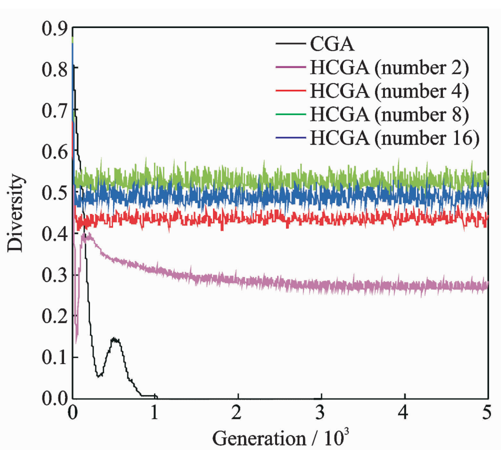


Fig. 6 Diversity change of F_4 on HCGA with different subpopulation numbers

6 Conclusions

The HCGA is proposed, in which GA is combined with a new operation inspired by PSO. The new operation replaces mutation in standard CGA, and enables population segmentation and genetic migration. By enhancing population diversity and reducing selection pressure, HCGA achieves a favorable global exploration/local exploitation balance. It improves not only the convergence rate and speed of conventional CGA, but also its stability. This paper also investigates the effect of subpopulation number on HCGA performance. The algorithm performs most effectively at a critical number of subpopulations. The result demonstrates that HGCA performance can be optimized by selecting an appropriate number of subpopulations. On each of the four test functions, the algorithm performance is optimized at the population segmentation number of 8 ($m=3$).

References:

[1] Billings S A, Yang Y. Identification of the neighborhood and CA rules from station-temporal CA patterns [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2003, 33(2): 332-339.

[2] Lu Y M, Li M, Li L. The cellular genetic algorithm with evolutionary rule [J]. Acta Electronica Sinica, 2010, 38(7): 1603-1607. (in Chinese)

[3] Li J H, Li M. Convergence analysis and convergence rate estimate of cellular genetic algorithms [J]. Moshi Shibie yu Rengong Zhineng/Pattern Recognition and Artificial Intelligence, 2012, 25(5): 874-878. (in Chinese)

[4] Alba E, Dorronsoro B. Cellular genetic algorithms [M]. USA: Springer Science and Business Media, LLC, 2008, 21-34.

[5] Ishibuchi H, Sakane Y, Tsukamoto N, NojimaA Y. Implementation of cellular genetic algorithms with two neighborhood structures for single-objective and multi-objective optimization [J]. Soft Computing, 2011, 15(9): 1749-1767.

[6] Alba E, Dorronsoro B. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms [J].

IEEE Transactions on Evolutionary Computation, 2005, 9(2): 126-142.

[7] Billings S, Yang Y. Identification of probabilistic cellular automata [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2003, 33(2):225-236.

[8] Kirley M. A cellular genetic algorithm with disturbance: optimization using dynamic spatial interactions [J]. Journal of Heuristics, 2002, 8(3): 321-342.

[9] Lu Y, Li M, Li L. Improved genetic algorithm based on migration differential individuals [J]. Systems Engineering and Electronics, 2011, 33(3): 1-4. (in Chinese)

[10] Liu Nan, Huang Jinquan. Performance seeking of turbo-shaft engines based on improved particle swarm optimization algorithm[J]. Journal of Nanjing University of Aeronautics and Astronautics,2013,45(3), 303-308.

[11] Keedwele E, Khu S T. A hybrid genetic algorithm for the design of water distribution networks [J]. Engineering Applications of Artificial Intelligence, 2005, 18(4):461-472.

[12] Jiang Yu, Yang Yingbao, Zhou Hang. Innovative predatory search algorithm for aircraft arrival sequencing and scheduling problems[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2010,27(4):361-364.

[13] Rezaeian J, Javadian N, Tavakkoli M R, Jolai F. A hybrid approach based on the genetic algorithm and neural network to design an incremental cellular manufacturing system[J]. Applied Soft Computing Journal, 2011, 11(6): 4195-4202.

[14] Jiradej V, Nasimul N, Hitoshi I. Polynomial selection: A new way to tune selective pressure[C]// Proceedings of The 2nd World Congress on Nature and Biologically Inspired Computing. Fukuoka, Japan: IEEE, 2010, 597-602.

[15] Kaveh A, SHahrouzi M. Dynamic selective pressure using hybrid evolutionary and ant system strategies for structural optimization [J]. International Journal for Numerical Methods in Engineering, 2008, 73(4): 544-563.

[16] Camargo G, Camargo J, Naufal J, Matiussi G. Definition of selective pressure control methods for optimization of genetic algorithms in air traffic control [C]// Proceedings of the 10th IASTED International Conference on Artificial Intelligence and Soft Computing. [S. l.]. ASC, 2006, 304-311.

[17] Chen S, Lu Y, Yang H, et al. Selection pressure study of cellular genetic with disturbances [J]. Computer Engineering and Applications, 2011, 47(27): 32-35. (in Chinese)

[18] Asmaa Al-Naqi , Erdogan A T, Arslan T. Adaptive three-dimensional cellular genetic algorithm for balancing exploration and exploitation processes [J]. Soft Computing, 2013,17(4):1-13.

(Executive editor: Zhang Tong)