Improved Adaptive Genetic Algorithm for Course Scheduling in Colleges and Universities

https://doi.org/10.3991/ijet.v13i06.8442

Wang Wen-jing
Business College of Shanxi University, Taiyuan, Shanxi, China
13754802471@163.com

Abstract—Traditional artificial intelligence and computer-aided course scheduling schemes can no longer meet the increasing demands caused by the informatization of teaching management in colleges and universities. To address this problem, this study designed an improved adaptive genetic algorithm that is based on hard and soft constraints for course scheduling. First, the mathematical model of the genetic algorithm was established. The combination of time, teacher, and course number was regarded as the gene coding. The weekly course schedule of each class was a chromosome, and the course schedule of the entire school was the initial population. The fitness was designed according to the priority of each class, curriculum dispersion, and teacher satisfaction. Local columns between individuals were selected through the roulette principle for a variation of crossover and random columns. Iterative calculation was implemented on the basis of the default mutation and crossover rates to study the optimal course scheduling scheme. Experimental results demonstrate that the improved adaptive genetic algorithm is superior to the original genetic algorithm. When the number of iterations is 150, population evolution is optimal and the fitness does not increase. When the population size is 150 classes, the average scheduling time is the shortest. The basic, adaptive, and improved adaptive genetic algorithms are compared in terms of the number of average iterations required for convergence, maximum individual fitness, and average individual fitness. Comparison results show that the improved adaptive genetic algorithm is superior to the two other algorithms. This study provides references for the model building and evaluation of course scheduling in colleges and universities.

Keywords—course scheduling, mathematical model, improved adaptive genetic algorithm

1 Introduction

Course times and places and teachers should be arranged every semester in colleges and universities according to the teaching plan and curriculum structure. Many rules are observed in course scheduling, which should consider such factors as the times and places of instruction and the teachers. The reasonable allocation of these factors forms a multi-constraint professional resource optimization problem. Course scheduling in most colleges and universities is presently manually implemented by teaching staff, who encounter numerous difficulties because of the large number of courses and teachers, required instruction places, and multiple constraints[1][2].

To address the general requirements of colleges and universities, this work regards each factor in the course scheduling problem as inputs into a genetic algorithm, which is a multi-objective optimization problem with constraints. Many constraints and combination factors are used in course scheduling, thereby increasing its complexity. A genetic algorithm is a parallel random search optimization algorithm that simulates natural and biological genetic and biological evolution. The algorithm establishes a model of biological evolution and implements the relevant calculation. The genetic algorithm can realize global optimization and parallel processing to optimize the configuration of various resources[3][4].

This study first establishes a mathematical model of course scheduling in colleges and universities. The improved adaptive genetic algorithm corresponds to the problems of college scheduling individually, and the optimal solution is searched by simulating natural evolutionary processes. The course scheduling in colleges and universities is designed and optimized. Finally, the NP-hard combinatorial problem is solved effectively[5].

2 State of art

After the 1990s, Arabinda Tripathy of the School of Management, Vastuper University, India, proposed course scheduling that is based on individuals and solved conflicts in course scheduling through the multi-class group method. Jean Aubin of Montreal University, Canada decomposed the problem of course scheduling into timetable and grouping[6][7]. The course scheduling decision support system was developed according to these two issues and includes modules of data processing, automatic optimization, and interactive optimization. Several scholars realized course scheduling and scheduling decision support systems through Lagrange relaxation technique-based branch and bound techniques[8][9]. In 1991, D. Whitey proposed a crossover operator that is based on domain crossover, which represents the individual crossover of genes through serial numbers, and applied the operator to the knapsack problem. D. H. Ackley et al. proposed an iterative genetic algorithm with hill-climbing methods of a complex probability election mechanism. The method determines the value of a new individual through m “voters” [10][11]. Experimental results indicated that the solution speed of the random iterative genetic algorithm with hill-climbing methods was higher than that of genetic algorithm, simulated annealing algorithm, tabu search, and a combination of several intelligent algorithms. Colomi et al. applied a matrix representation scheme with crossover and mutation operators to course scheduling in high schools in Milan. Since 2002, Lalescu has been devoted to studying the application of genetic algorithms in course scheduling systems and developed the software FET. However, the rules and conditions of course scheduling in foreign countries differ from those in China[12][13].

Although domestic scholars began studying course scheduling relatively late, scheduling algorithms are now being studied by many colleges and universities. In

1984, Lin Zhangxi and Lin Yaorui of Tsinghua University conducted an experimental research and designed a timetable scheduling system called TISER（Tmaetable Schedul ER）. The university timetable scheduling system of the Nanjing Institute of Technology and the intelligent teaching organization management and curriculum scheduling system of the Dalian Institute of Technology use artificial intelligence expert systems and decision support systems to simulate artificial course scheduling. Southwest Jiaotong University proposed a course scheduling algorithm that uses the calculation of the class element and the candidate space–time slice of the course element as the core. Yanbian University developed an automatic course scheduling algorithm with a computer-based data structure[14][15]. In 2002, Dai Xiaoming et al. presented the parallel evolution of multi-population heredity; different mutation operators search a variable space, various populations use different genetic strategies, and genetic information is exchanged through a population migration operator to address the convergence of classical genetic algorithms to local optima[14][15]. In 2004, Zhao Hongli et al. proposed a parallel genetic algorithm for gene block coding considering the low search efficiency of a Simple Genetic Algorithm (SGA) in large-scale combinatorial optimization problems. In 2005, Jiang Lei et al. used the parallel genetic algorithm to solve the knapsack problem and attempted to maintain the diversity of the population through an elastic strategy; the authors obtained good experiment results; the algorithm crossed the obstacle of local convergence and evolved in the direction of global optimization. On the basis of analyzing and summarizing the experience of the original course arrangement, Shanxi University presented a formal description to solve the problem of course scheduling and realized an automatic course scheduling system that is based on the knowledge reasoning of this idea. Liu Hong et al. proposed the principle of artificial intelligence to realize the course scheduling of colleges and universities. In recent years, greedy and backtracking algorithms have been used to address the problem of course scheduling in China. However, the greedy algorithm is not necessarily the optimal solution, and the backtracking algorithm should be used with other algorithms because of high time complexity[16][17]. Therefore, complex course scheduling, which involves considerable calculation, should not only rely on pure mathematical methods but also plan and solve the problem step by step by drawing lessons from operational research.

The course scheduling schemes in the abovementioned studies are based on the curriculum characteristics and teaching resources of the corresponding colleges and universities and thus cannot be popularized on a large scale. Each university has its own characteristics of teaching and operation. Courses should be scheduled according to the curriculum characteristics and teaching resources of schools. In the meantime, the following problems also occur. (1) As in the case of combined classes, the constraints considered are not comprehensive enough. (2) The objective function considers only the number of times the constraint is violated and not the importance of various courses and the different teaching effects in varying periods. (3) Each university should consider its actual teaching implementation when formulating a teaching plan. In view of these shortcomings, an improved method is proposed.

3 Methodology

3.1 Constraints on course scheduling

The rules in course scheduling are divided into hard and soft rules. Hard rules are immutable and must be executed in accordance with plans, talent programs, and teaching objectives. Soft rules are satisfied as much as possible if conditions permit, thereby improving the feasibility of the scheme[3]. The hard and soft rules of the proposed method are as follows.

The hard rules are the following. (1) The number of courses offered and the corresponding hours are fixed according to the teaching plan and talent training objective. (2) Each course can be offered in only one class at the same time. (3) Each teacher can teach only one course at the same time. (4) A course can be offered in only one classroom at the same time for several classes together. (5) The capacity of a classroom must be greater than the number of students in a class.

The soft rules are the following. (1) Teachers who have requirements for class scheduling should be considered. (2) PE classes should not be the first or second class in the morning. (3) Public courses take precedence over professional courses. (4) Teachers' course schedules should be as concentrated as possible to allow them to rest or have spare time for studying and preparing lessons. (5) Courses with many periods should be prioritized. (6) The schedules of theoretical and practical courses should be considered in certain majors and courses. (7) Teaching times of the same course should be determined such that hectic schedules are avoided.

Every university can formulate soft rules according to their own situation. On the basis of fulfilling the hard conditions, this study aims to arrange a course schedule in accordance with the soft rules without serious conflict or error to improve the working efficiency of teaching staff.

3.2 Course scheduling through genetic algorithm

To correspond course scheduling with SGA, the definition of the genetic algorithm should be considered. The most important factors in genetic algorithm are the coding, fitness function, initial population, population size, crossover rate, operator, mutation rate, and abort condition[18][19]. Each course has a course number, teaching time, and teacher, and the combination of these three objects is considered a gene. A chromosome is a combination of genes; that is, the combination of course scheduling or the solution of the problem. Initial population refers to a variety of scheduling programs. Individuals select operators through roulette selection. The selection probability of an individual is distributed on a roulette. The greater the fitness, the larger the probability of being selected as a parent individual[20][21].

Population establishment and conflict detection. A genetic algorithm needs to create an initial population composed of character strings. The initial population comprises individuals, which are composed of chromosomes. The chromosomes are made up of genes. The gene coding is as follows. The coding of teaching task can be written as the time + teacher number + course number. First, the teaching tasks are written into the course schedule individually. The course schedule is a two-dimensional table. The columns indicate the time (Monday–Friday), and the lines indicate the number of courses every day (1–6). T1–T2 represent the first and second classes on Monday. If a time is occupied, a new period will be generated. Finally, the substitute teacher numbers are placed in Course schedule 2 without repetition, thereby forming a class schedule. This scheduling method meets Constraints 1, 3, and 4. Course schedules are arranged for N classes via the same operation, thereby obtaining N chromosomes that are individuals. The weekly schedules of N classes form a two-dimensional weekly schedule of classes and teaching times, as shown in Table 1. Meanwhile, the weekly teaching tasks in Table 2 generate the initial population. The population size is defined as 15, thereby resulting in 15 schedules.

Table 1. Initial population table

Time	Class 1	Class 2	Class ….	Class n
T1–T2	24	16	15	12
T3–T4	6		7	4
T5–T6	9	13	9	
T7–T8	16	24	21	
T9–T0				
….	….	….	….	….

Table 2. Course schedule for Class 1

Time	Monday	Tuesday	Wednesday	Thursday	Friday
T1–T2	24	16	14	11	
T3–T4	6			17	19
T5–T6	9	8	12		

Conflict should be checked once the initial population is established. The teacher number on the first column on the first row in Table 1 (1, 1) should be checked and compared with that on the second column on the first row. The chromosome coding values are compared if the numbers are the same. The teacher is teaching in a combined class if the coding values are the same. Errors exist in course scheduling, and a random number i is generated if the code systems of other bits in the chromosomes are different. Cells (1, 1) and (1, i) in Table 1 are exchanged for recomparison. Data comparison in the first line is completed. Subsequently, Table 1 (2, 1) should be checked, and the above steps should be repeated until the end. Consequently, the conflict in which the same teacher teaches multiple courses at the same time is solved; that is, Constraint 2 is satisfied. Courses are allocated at the same time period in Table 1 that require m language laboratories, n machine rooms, 1 small classroom, and x large multimedia classrooms, which are distributed according to the number of available classrooms. Consequently, Constraint 5 is satisfied.

Fitness function. Fitness affects the iterative direction and convergence speed of a genetic algorithm and thus reflects the advantages and disadvantages of the course schedule.

Weight of course interval. In general, teaching in the morning is more effective than that in the afternoon because students are more focused in the morning. The expected priority value of each class is obtained according to years of teaching experience. Separate expected values are required for special courses, such as PE. The priority weight distribution for each class is shown in Table 3.

Weight of course interval. The weights of different time intervals of the same course within one week are also different. The weight distribution of the theoretical course interval is shown in Table 4.

The fitness function is defined as Fit = k1*w1+k2*w2+punish. k1 and k2 are the weights for each class priority and class combination priority, respectively, where k1+k2 =1. "Punish" indicates teacher satisfaction. Before course scheduling, the priority for each teacher's teaching time should be established. Teachers select the time they prefer to schedule classes according to their requirements, as shown in Table 5.

Table 5 indicates the time that teachers do not want to schedule classes. Teachers aim to concentrate their courses because the campus is far from urban areas. Therefore, this work uses the punish weight value, teachers like +5, do not like −5, dislike −10, and do not know 0.

Table 3. Priority weight distribution for each class

Number of courses	1, 5, 9, 13, 17	2, 6, 7, 11, 14, 18	3, 8, 10, 15, 19	4, 12, 16, 20
Expected value	0.8	0.6	0.4	0.1

Table 4. Weight of course interval

Time difference	1, 2, 3, 19, 18	4, 5, 15, 16, 17	6, 7, 11, 12, 13	8, 9, 10, 14
Expected value	0.2	0.4	0.6	0.8

Table 5. Teachers' satisfactory scheduling time

Time	Monday	Tuesday	Wednesday	Thursday	Friday
T1–T2	−1			−1	
T3–T4			−1		
T5–T6	−1				

Selection, crossover, and compilation. Selection is based on the roulette method, and the probability that chromosomes are selected is proportional to fitness. This method cannot guarantee the selection of a chromosome with a high fitness value, but the possibility is relatively large. This study adopts the single-point crossover method, and the probability of individual destruction is relatively small. The predetermined crossover probability is compared with a random value n. Crossover exists if n<PC. A cross point is randomly selected. A column, namely, the course schedule of a class, is selected from Table 1 and exchanged with the same column of another individual. Teaching time is constant during the exchange. For example, Parents 1 and 2 cross to generate the next-generation Individuals 1 and 2, as shown in Tables 6, 7, 8, and 9.

Table 6. Parent 1

Time	Class 1	Class 2	Class 3	Class….	Class n
T1–T2	21	12	3		
T3–T4	3	0	5		
T5–T6	0	9	11		
T7–T8	5	7	8		
T9–T10	8	11	22		
….	….	….	….	….	….

Table 7. Parent 2

Time	Class 1	Class 2	Class 3	….	Class n
T1–T2	21	7	4	….	
T3–T4	8	0	5	….	
T5–T6	0	12	11	….	
T7–T8	7	11	8	….	
T9–T10	9	12	23	….	
….	….	….	….	….	….

Table 8. Individual 1

Time	Class 1	Class 2	Class 3	….	Class n
T1–T2	21	7	3		
T3–T4	3	0	5		
T5–T6	0	12	11		
T7–T8	5	11	8		
T9–T10	8	12	22		
….	….	….	….	….	….

Table 9. Individual 2

Time	Class 1	Class 2	Class 3	….	Class n
T1–T2	21	12	4		
T3–T4	8	0	5		
T5–T6	0	9	11		
T7–T8	7	7	8		
T9–T10	9	11	23		
….	….	….	….	….	….

The probability of mutation is generally small enough to avoid the destruction of the optimal solution. Schaffer recommended that the optimal mutation rate is 0.001–0.05. On the basis of the mutation principle, one or more bits are selected to reverse the bit string of an individual in the population in mutation probability. The mutation that corresponds to course scheduling refers to the random position of several column exchanges with the code of another random position in the same column to ensure that the teaching task of the mutation class is invariable. Mutation is implemented if r<pm. The mutation of Individual 2 into Individual x yields Tables 10 and 11.

Table 10. Individual 2

Time	Class 1	Class 2	Class 3	….	Class n
T1–T2		12			
T3–T4		0			
T5–T6		9			
T7–T8		7			
T9–T10		11			
….	….	….	….	….	….

Table 11. Individual x

Time	Class 1	Class 2	Class 3		Class n
T1–T2		12			
T3–T4		0			
T5–T6		11			
T7–T8		7			
T9–T10		9			
….	….	….	….	….	….

3.3 Course scheduling based on improved adaptive genetic algorithm

The course scheduling algorithm based on the improved genetic algorithm is implemented as follows.

1. The course task is coded according to the characteristics of the course scheduling.
2. The population is initialized according to the teaching task and outline, and the constraint condition is satisfied. The hard rules are first satisfied, and the soft rules are subsequently fulfilled.
3. The weight value, reward, and punishment right values are set according to the curriculum combination, time combination, curriculum characteristics, and teacher satisfaction, and individual fitness is calculated on the basis of weight.
4. Fitness is selected using the roulette principle. If no fitness is selected, then Operation (3) is implemented again. If several fitnesses are selected, then cross mutation is applied to generate filial generation and Operation (3) is performed again.
5. When the number of iterations reaches the requirement, the output results are obtained and the algorithm ends.

Courses should be scheduled according to the teaching tasks. Therefore, the condition for terminating the algorithm is the iteration time reaching a. The values of p_c and p_m, which directly influence the convergence, should be set. The new generation of breeding will be fast and the fitness will be disturbed if p_c is extremely high. The progeny reproduction will be slow if p_c is extremely small. The algorithm will be destroyed if p_m is large. A new individual will not be generated if p_m is small, which is unsuitable for the generation of the next generation of outstanding individuals[22][23]. The adaptive genetic algorithm adopts the following formula of p_c and p_m, which was proposed by Schaffer.

$$p_c = \frac{p_{c2}\left(f_m - f^{'}\right)}{f_m - f_a} \qquad \left(f^{'} \geq f_a\right) \tag{1}$$

$$p_c = p_{c_1} \qquad \left(f^{'} < f_a\right) \tag{2}$$

$$p_m = \frac{p_{m2}\left(f_m - f\right)}{f_m - f_a} \qquad \left(f \geq f_a\right) \tag{3}$$

$$p_m = p_{m1} \qquad \left(f < f_a\right) \tag{4}$$

where f_m is the individual with the largest fitness in a population; f_a is the average fitness in a population; $f^{'}$ is the individual with the largest fitness in a cross operation; f is the fitness of variant individuals; p_{c1}, p_{c2}, p_{m1}, and p_{m2} are the values of [0, 1], which are obtained through Formulas (1), (2), (3), and (4). The closer f_m is to f, the smaller the crossover and mutation rates. If f_m and f are equal to the maximum fitness, then the crossover and mutation rates are zero. If the better individual is in an unchanged state, then the solution is not necessarily global optimization. Therefore, this method is unsuitable for population evolution in the early stage. The algorithm is improved. Thus, the crossover and mutation rates of individuals with the largest fitness values in the population will not be zero, and p_m and p_c can be increased. The formula is rewritten as:

$$p_c = p_{c1} - \frac{\left(p_{c1} - p_{c2}\right)\left(f^{'} - f_a\right)}{f_m - f_a} \qquad \left(f^{'} \geq f_a\right) \tag{5}$$

$$p_c = p_{c1} \qquad \left(f^{'} < f_a\right) \tag{6}$$

$$p_m = p_{m1} - \frac{\left(p_{m1} - p_{m2}\right)\left(f - f_a\right)}{f_m - f_a} \qquad \left(f \geq f_a\right) \tag{7}$$

$$p_m = p_{m1} \qquad \left(f < f_a\right) \tag{8}$$

where $p_{c1} = 0.9$, $p_{c2} = 0.6$, $p_{m1} = 0.1$, and $p_{m2} = 0.001$.

4 Result analysis and discussion

In addition to the crossover and mutation rates, the population size and the number of iterations in the genetic algorithm affect the experimental results. The iteration number parameters, namely, T = 30, T = 50, T = 100, T = 150, and T = 300, are tested in the experiment. The maximum fitness for different iterations is calculated when the mutation and crossover probabilities are constant, and the test results are shown in Table 12.

Table 12 shows that the improved adaptive genetic algorithm is closely related to the convergence of the algorithm. The population evolution approaches the optimal and the fitness does not increase when the number of iterations reaches 150. Corresponding tests are conducted when the population sizes are M = 40, M = 90, M = 120, and M = 150. The algorithm converges, and the test results are shown in Table 13. The iteration number is 50. As shown in Table 13, the time consumption for course scheduling is the shortest when M = 150 classes.

The average time is the shortest when population M<=150 and T<=300. However, the optimal course schedule in accordance with the constraints is obtained. The algorithm is compared with the adaptive genetic algorithm in terms of the number of convergent iterations, number of local convergences, maximum individual fitness, and average individual fitness. The improved experimental data are shown in Table 14.

Table 12. Relationship between iteration number and maximum fitness

Group number	Iteration number	Maximum fitness
1	30	1456
2	50	1678
3	100	2109
4	150	2208
5	300	2207

Table 13. Relationship between population scale and maximum fitness

Group number	M = 40 Time (s)	M = 90 Time (s)	M = 120 Time (s)	M = 150 Time (s)
1	4456	4567	3478	1231
2	4678	3473	2789	1345
3	4109	3109	2990	1675
4	4208	3401	2763	1873
Average time	4362.75	3637.5	3005	1531

As shown in Table 14, the number of average iterations for convergence of the improved adaptive genetic algorithm is 50 times fewer than the that of the adaptive genetic algorithm and the maximum individual and average individual fitnesses are significantly increased. The relationships between the number of iterations and the maximum fitness and between the population size and the maximum fitness show that the optimal course scheduling scheme can be obtained when M<=150. Meanwhile, a comparison of the improved algorithm and the adaptive genetic algorithm shows that the average convergence of the former is 50 higher than that of the latter. The maximum and average fitnesses are also increased by 5454 and 6164, respectively.

Meanwhile, the experiment evaluates the quality of course scheduling from five perspectives, namely, the maximum fitness value of the individual in the population, the priority of each class in course scheduling, course dispersion, teacher satisfaction, and operation time. The experimental results are shown in Table 15.

Table 15 is converted to Figure 1, and they show that although the operation time of the improved adaptive genetic algorithm is longer than that of the basic and genetic algorithms, the course arrangement, course dispersion, and teacher satisfaction of the improved algorithm is better than those of the two others. Therefore, the improved genetic algorithm is the best of the three.

Table 14. Results of Experiment 1

Algorithms	**Number of average iterations for convergence**	**Maximum individual fitness**	**Average individual fitness**
Basic genetic algorithm (GA)	Cannot realize convergence		
Adaptive genetic algorithm (GA1)	212	17689	13876
Improved adaptive genetic algorithm (GA2)	162	23143	20010

Table 15. Results of Experiment 2

Evaluation	**GA**	**GA1**	**GA2**
Maximum fitness value		16789	24560
Priority of each class in course scheduling	7890	8890	9980
Course dispersion	5698	7760	8678
Teacher satisfaction	3990	6789	6909
Operating time (seconds)	3490	5990	6785

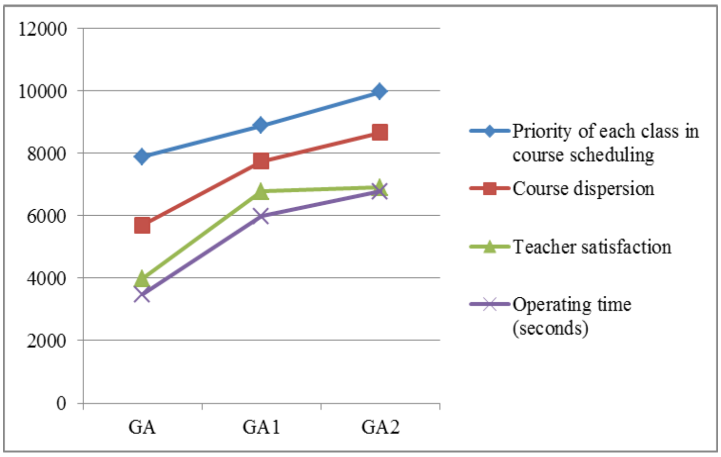

Fig. 1. Comparison of experimental results of three algorithms

5 Conclusions

On the basis of the actual situation of colleges and universities, the improved adaptive genetic algorithm was applied to course scheduling, and the daily goodness of fit, course combination degree, and teacher satisfaction were set through the fitness function. A scheme with high fitness satisfied the course scheduling requirements and reduces the time of manual scheduling. Therefore, applying the improved genetic algorithm to course scheduling was scientific and feasible. The following conclusions could be drawn.

1. The average scheduling time is the shortest when population size is 150 classes.
2. The adaptive genetic algorithm and the improved adaptive algorithm are compared using the same parameters. The experimental analysis shows that the improved genetic algorithm converges faster than the adaptive genetic algorithm, and the effect of course scheduling is good.
3. The improved adaptive algorithm is superior to the adaptive genetic algorithm in terms of the number of convergent iterations, local convergence time, maximum individual fitness, and average individual fitness.
4. The average number of iterations for convergence, maximum individual fitness, and average individual fitness show that the improved adaptive genetic algorithm is superior to the genetic and adaptive genetic algorithms.

This study discusses the application of the genetic algorithm in college course scheduling and proposes a new method, which adopts certain references from the research of college scheduling. However, several problems, such as alternate week class scheduling and decrease in calculation rate caused by the increase in population size, are found. Course scheduling is a complex combinatorial optimization process. Additional factors should be considered in future study and research, and new algorithms should be tested to provide enhanced course scheduling methods.

6 References

[1] Jat S N, Yang S. A hybrid genetic algorithm and search approach for post enrolment course timetabling, Journal of Scheduling, 2008, vol. 14(6), pp. 617-637. https://doi.org/10.1007/s10951-010-0202-0

[2] Li Hongchan, Zhu Haodong. Decimal Immunization GA used to solve UTP, Systems Engineering—Theory & Practice, 2012, vol. 32(9), pp. 2031-2036.

[3] Li Hongchan, Zhu Haodong. University Timetabling Problem Solving Based on the Best Individual Replacement Strategy , Computer Engineering, 2011, vol. 37(19), pp. 186-188.

[4] Shara S. A. Alves. A novel educational timetabling solution through recursive genetic algorithms, 2015 Latin America Congress on Computational Intelligence (LA-CCI), 2015, pp. 1-6.

[5] P Flores. PWiseGen: Generating test cases for pairwise testing using genetic algorithms, IEEE International Conference on Computer Science & Automation Engineering, 2011,vol. 2 (1), pp. 747-752.

[6] Zhang Chaoqun, Zheng Jianguo, Qianjie. Comparison of Coding Schemes for Genetic Algorithms, Application Research of Computers. 2011,vol. 28 (3), pp.819-822.

[7] Chen Minghua, Zhou Benda, Ren Zhe. Genetic Algorithm based on Random Uniform Design, Applied Mathematics A Journal of Chinese Universities, 2010, vol. 3 (3), pp.279-284.

[8] M Wei . Design and implementation of university arrangement system based on the genetic algorithm, IEEE International Conference on Computer Science & Automation Engineering, 2012, vol. 3(1), pp. 591-594.

[9] Gang Li, Fuyu Zhao. Load following control and global stability analysis for PWR core based on multi-model, LQG, IAGA and flexibility idea, Progress in Nuclear Energy, 2013, vol. 7(3), pp. 80-89. https://doi.org/10.1016/j.pnucene.2013.03.015

[10] Hongbin Wu.Optimal allocation of microgrid considering economic dispatch based on hybrid weighted bilevel planning method and algorithm improvement, International Journal of Electrical Power and Energy Systems, 2016, pp. 28-37. https://doi.org/10.1016/j.ijepes.2015.08.011

[11] Jin Jiang. A novel recognition system for human activity based on wavelet packet and support vector machine optimized by improved adaptive genetic algorithm, Physical Communication, 2014, pp. 211-220. https://doi.org/10.1016/j.phycom.2014.04.006

[12] T Vidal .A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows, Computers & Operations Research, 2013, vol. 40 (1),pp. 475-489. https://doi.org/10.1016/j.cor.2012.07.018

[13] JL Ponz-Tienda.The Resource Leveling Problem with multiple resources using an adaptive genetic algorithm, Automation in Construction, 2013, vol. 29(1), pp. 161-172. https://doi.org/10.1016/j.autcon.2012.10.003

[14] CW Chen. RETRACTED: Modified intelligent genetic algorithm-based adaptive neural network control for uncertain structural systems, Journal of Vibration & Control, 2013, vol. 19(9), pp.1333-1347. https://doi.org/10.1177/1077546312442232

[15] Y Guo. A Multi-agent Based Self-adaptive Genetic Algorithm for the Long-term Car Pooling Problem, Journal of Mathematical Modelling & Algorithms in Operations Research, 2013, vol. 12 (1),pp.45-66. https://doi.org/10.1007/s10852-012-9175-7

[16] X Zhou. Dynamic RMSA in elastic optical networks with an adaptive genetic algorithm, Global Communications Conference, 2013, pp.2912-2917.

[17] W Neungmatcha. Adaptive genetic algorithm for solving sugarcane loading stations with multi-facility services problem, Computers & Electronics in Agriculture, 2013, vol. 98 (7), pp.85-99. https://doi.org/10.1016/j.compag.2013.07.016

[18] F Wang. An Improved Adaptive Genetic Algorithm for Image Segmentation and Vision Alignment Used in Microelectronic Bonding, IEEE/ASME Transactions on Mechatronics, 2014, vol. 19(3), pp.916-923. https://doi.org/10.1109/TMECH.2013.2260555

[19] Y Du. Frequency-Domain System Identification of an Unmanned Helicopter Based on an Adaptive Genetic Algorithm, IEEE Transactions on Industrial Electronics, 2014, vol. 61 (2), pp.870-881. https://doi.org/10.1109/TIE.2013.2257135

[20] L. De Giovanni . An adaptive genetic algorithm for large-size open stack problems, International Journal of Production Research, 2013, vol. 51(3), pp.682-697. https://doi.org/10.1080/00207543.2012.657256

[21] LY Wei. A hybrid model based on ANFIS and adaptive expectation genetic algorithm to forecast TAIEX, Economic Modelling , 2013, vol. 33(2), pp.893-899. https://doi.org/10.1016/j.econmod.2013.06.009

[22] R Kumar .Optimal Capacitor Placement in Radial Distribution System using Adaptive Genetic Algorithm, IEEE India International Conference on Power Electronics, 2014, pp.1-6.

[23] S Wang. Fitness-scaling adaptive genetic algorithm with local search for solving the Multiple Depot Vehicle Routing Problem, Simulation, 2016, vol. 92(7), pp 601-616. https://doi.org/10.1177/0037549715603481

7 Author

WANG Wen-jing is a lecturer in the College of Information, Business College of Shanxi University, Taiyuan 030031,Shanxi,China (13754802471@163.com).

Article submitted 13 February 2018. Final acceptance 25 April 2018. Final version published as submitted by the author.