International Journal for Housing Science and Its Applications
Publish August 4, 2025. Volume 46, Issue 3 Pages 3306-3317

https://doi.org/10.70517/ijhsa463279

Research on Optimization and Application of Genetic Algorithm Based on Intelligent Class Scheduling System

Junliang Hou[1,*]
[1] Geely College, Chengdu, Sichuan, 610000, China
Corresponding authors: (e-mail: xg15680883257@163.com).

Abstract In order to improve the shortcomings of the current school scheduling system which is inefficient, this paper meticulously researches the scheduling problem and constructs a mathematical model of school scheduling. Genetic algorithm and ant colony algorithm are applied to intelligent class scheduling respectively. Combine the single algorithms to construct an intelligent scheduling model based on genetic-ant colony algorithm. After verifying the superiority of the scheduling performance of the genetic-ant colony algorithm, the scheduling quality, efficiency, and the satisfaction rate of the settable rules of the algorithm are tested. The optimal solution adaptation and average adaptation of the genetic-ant colony algorithm in this paper are better than the comparison algorithm, and although it takes more time than other algorithms, the algorithm in this paper has the best overall scheduling performance. The genetic-ant colony algorithm is better than other algorithms in the overall performance of class scheduling. The genetic-ant colony scheduling model in this paper has a high satisfaction rate of 100% for students' class selection, the mean value of overall rule satisfaction rate and the mean value of scheduling satisfaction rate are more than 95%, the classroom utilization rate reaches 88.1%, and the course uniformity is 0.8, which obtains a good scheduling effect.

Index Terms genetic algorithm, ant colony algorithm, intelligent scheduling, scheduling effect

I. Introduction

Scheduling is an important work of teaching management in colleges and universities, the essence of which is to rationally arrange teachers, students, courses, classrooms, etc. under the existing software and hardware environment of the school so that the teaching planning of the school can be carried out in a planned and orderly manner [1]-[3]. With the increase in the number of students in colleges and universities, the scale of courses offered by faculties has risen dramatically, and scheduling has become an extremely complex task [4]. Since manual scheduling does not have any theoretical guidance or mathematical modeling, and mainly relies on the experience of the scheduler, it has great disadvantages, and this method is suitable for schools with smaller scale and fewer courses [5]-[7]. At the same time, just relying on manual scheduling, then it will cost a lot of human, material and financial resources, the workload is very large, and it can not exclude easy to adjust the schedule [8], [9]. Based on the above problems, by introducing intelligent algorithms to implement modeling of class scheduling, it can be solved while satisfying the constraints of time, teachers, classrooms, courses, students, etc., effectively realizing automated class scheduling, and improving the correctness of class scheduling and class scheduling efficiency [10]-[13]. It is conducive to promoting the management and development of class scheduling, and also indirectly promotes the informatization of teaching management, which is of great significance [14], [15].

Genetic algorithm, as an optimization algorithm that simulates biological evolutionary process, has outstanding performance in many optimization problems, and many scholars have incorporated it into scheduling optimization models to generate intelligent scheduling results. Zhang, Q designed an improved genetic algorithm based on co-evolution and applied it to an intelligent scheduling system in universities, which showed a faster convergence rate as well as a better quality solution compared to the traditional genetic algorithm model [16]. Wen-jing, W incorporated hard and soft constraints into a genetic algorithm to form an improved adaptive genetic algorithm suitable for scheduling assistance, which is computed iteratively by confirming the variances of crossover and random columns in the initial population and iteratively in order to generate optimal scheduling schemes [17]. Chen, X et al. researched and tested a variant genetic algorithm applied to student scheduling system, which not only solved the limitations brought by human scheduling, but also was more effective than other intelligent scheduling algorithms, providing a strong support for teaching management [18]. Han, X. and Wang, D fully consider the impact of joint scheduling on the overall scheduling results, combining genetic algorithms and dynamic programming algorithms to construct a mixed-integer linear programming model that can optimize the scheduling of independent and joint courses simultaneously, and the proposed algorithm still achieves efficient scheduling under the complexity constraints [19]. Haider, S. A et al. proposed a self-learning scheduling genetic algorithm GAGDRL based on deep reinforcement learning, which significantly improves the model's scheduling accuracy and optimization seeking ability by introducing a deep Q-neural network compared to other genetic algorithms, and plays an important role in educational scheduling systems [20]. Xiang, K et al. introduced a mixed-integer linear programming paradigm with course time and scheduling as constraints, which is combined with the designed two-stage meta-heuristic algorithm to complete clustering and assignment of courses, and obtain the optimal scheduling plan for colleges and universities that satisfy the constraints [21]. The algorithms designed in the above research provide valuable improvement suggestions for the scheduling scheme of colleges and universities to a certain extent, but there are still problems such as large influence of parameter settings, easy to fall into local optimal solutions, etc., and there is a large optimization space.

The article incorporates factors such as teachers, classes, courses, classrooms and time into the school scheduling system, and constructs a mathematical model of school scheduling with full consideration of the degree of optimization of the time period, the balanced distribution of class time, the degree of utilization of classrooms, and the scheduling of sports courses. Explore the application of traditional genetic algorithm and ant colony algorithm in the school intelligent scheduling problem, combine the two, optimize the single scheduling model, and construct the intelligent scheduling model based on hybrid genetic-ant colony algorithm. Relevant experiments are designed to test the scheduling performance of the genetic-ant colony algorithm through the indexes of optimal solution adaptability, average adaptability and average computing time of the algorithm. The scheduling model of this paper is used in practice to test the scheduling quality, scheduling efficiency, and scheduling setable rule satisfaction rate of the hybrid genetic-ant colony algorithm.

II. Mathematical modeling of the school scheduling problem

II. A. Mathematical model description

Teacher, class, course, classroom and time are the five main factors in a school scheduling problem and each factor entity has its own unique number.

In a school scheduling system, apart from these five major factors, there are other factors that affect school scheduling. For example, the courses that are offered on a bi-weekly basis, the type of classroom, the number of people that the classroom can accommodate, the optimal time period of the day for learning, and the weighting of the sessions during the week all affect the quality of this scheduling solution.

Assuming that there are j teachers, b classes, k courses, and s classrooms in the school, the mathematical model of the main factors of school scheduling is described as follows:

(1) Teachers: the set of teachers is denoted as $P=\left\{p_1,p_2,p_3,\ldots,p_j\right\}$.

(2) Classes: The set of classes is denoted as $C=\left\{c_1,c_2,c_3,\ldots,c_b\right\}$. For different classes the number of students included is denoted as $Cv=\left\{Cv_1,Cv_2,Cv_3,\ldots,Cv_b\right\}$.

(3) Courses: the set of courses is denoted as $L=\left\{l_1,l_2,l_3,\ldots,l_k\right\}$.

(4) Classrooms: The set of classrooms is denoted as $R=\left\{r_1,r_2,r_3,\ldots,r_s\right\}$. For each classroom the number of students that can be accommodated is denoted as $Rv=\left\{Rv_1,Rv_2,Rv_3,\ldots,Rv_s\right\}$.

(5) Time: The set of time is denoted as $T=\left\{t_1,t_2,t_3,\ldots,t_{25}\right\}$.

The solution space of the scheduling problem then consists of the Cartesian product of the five elements $P\times C\times L\times R\times T$, specifying that when the teacher p_j gives the class c_b to teach the course l_k in the classroom r_s at time period t_i, denoted $p_jc_br_st_il_k=1$, otherwise $p_jc_br_st_il_k=0$. See equation (1) for a detailed description:

$$p_jc_br_st_il_k=\begin{cases}1, & \begin{array}{l}\text{Time period } t_i\text{, teacher } p_j \text{ gives class } c_b\\ \text{a lesson in classroom } r_s\ \ l_k\end{array}\\ 0, & \text{Other}\end{cases} \tag{1}$$

II. B. Constraints in the model

Constraints are divided into hard constraints and soft constraints, hard constraints are the guidelines that must be followed in the scheduling process, if violated, this program is directly discarded. Soft constraints for the icing on the cake of the constraints, in the case of meeting the hard constraints, to meet more soft constraints, will generate a more reasonable scheduling program, soft constraints also serve as the objective function.

In summary, the hard constraints of the school scheduling system are as follows five, where b represents class, k represents course, s represents classroom, and j represents teacher.

(1) Only one classroom can be assigned to the same teacher and only one course can be taught during the same class period. This can be expressed as equation (2):

$$\sum_{b=1}^{B}\sum_{k=1}^{K}\sum_{s=1}^{S} p_j c_b r_s t_i l_k \leq 1 \tag{2}$$

(2) Only one classroom and only one course can be assigned to the same class during the same class period. This can be expressed as formula (3):

$$\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{s=1}^{S} p_j c_b r_s t_i l_k \leq 1 \tag{3}$$

(3) Only one course can be assigned to the same classroom during the same class period. This can be expressed as formula (4):

$$\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{b=1}^{B} p_j c_b r_s t_i l_k \leq 1 \tag{4}$$

(4) The number of people that can be accommodated in the classroom in which the instruction takes place is greater than or equal to the number of classes attending the classroom at the same time. This can be expressed as formula (5):

$$c_v \leq r_v \tag{5}$$

(5) The type of classroom should match the type of course. This can be expressed as formula (6):

$$roomType = courseType \tag{6}$$

The soft constraints are as follows 6:

a) In order to improve the effectiveness of classes, more difficult courses such as required courses and specialized courses are scheduled in the first morning session or the first afternoon session as much as possible.

b) The scheduling of classes for teachers is rationalized according to their willingness to attend classes at various times of the week.

c) Each class should have an even distribution of courses during the week.

d) The same course for the same class should be distributed as evenly as possible during the week. For example, if two English courses are scheduled for a certain class in a week, the two courses should be separated by one or two days as much as possible.

e) Classroom capacity and the number of students in the classroom as close as possible to reduce the waste of classroom resources.

e) Physical education courses should be scheduled in the second period of the morning or the second period of the afternoon as much as possible.

II. C.Modeling of scheduling

II. C. 1) Degree of time period optimization

Assuming that there are n courses in the whole university, the time period weight of the i th course is t_i, the degree of time period optimization is denoted by F_1, and α_j denotes the weight corresponding to the type of the course, such as compulsory courses, specialized courses, elective courses, and sports courses. Then the degree of time period optimization can be expressed as equation (7):

$$f_1 = \frac{1}{n}\sum_{i=1}^{n} \alpha_j t_i \tag{7}$$

The formula is the average of the time slot weights of all courses multiplied by the accumulated addition of the weights of the corresponding course types and then divided by the number of courses in the whole university. The larger the F_1, the higher the degree of optimization of the corresponding scheduling plan's time slots. The smaller the F_1, the lower the degree of optimization of the corresponding scheduling plan time slot.

II. C. 2) Equalization of the distribution of lesson time

A good scheduling plan lesson distribution should be balanced, rather than all the lessons are concentrated in a certain period of time, which is not conducive to students' digestion and absorption of classroom knowledge, nor is it conducive to maintaining the quality of teachers' teaching. Suppose C_i denotes a class, d denotes a certain teaching day from Monday to Friday, m_d denotes the number of classes in class C_i on the d teaching day, the number of days of class per week is set to n, and m_{avg} denotes the average number of classes in the class per day. Then the average daily lesson distribution of class C_i with balanced degree lessons is expressed as equation (8):

$$C_i = \frac{1}{\sqrt{\sum_{d=1}^{n}\left(m_d - m_{avg}\right)^2}} \tag{8}$$

where m_{avg} can be expressed as equation (9):

$$m_{avg} = \frac{\sum_{d=1}^{n} m_d}{n} \tag{9}$$

The above is a way to represent the equilibrium of the distribution of hours for a class, for the whole school with w classes, the equilibrium of the distribution of average daily hours for all classes in the school can be expressed as equation (10):

$$f_2 = \frac{\sqrt{\sum_{i=1}^{w} c_i}}{w} \tag{10}$$

where the larger the value of F_2, the better the equalization of the class distribution of the scheduling scheme, and the worse the opposite.

II. C. 3) Extent of classroom utilization

If a class with a very small number of students is scheduled to occupy a classroom with a much larger capacity than the class, this not only greatly reduces the utilization of classroom resources, but may also result in larger classes being scheduled without a suitable classroom. Classroom utilization can be determined by the ratio of the number of classes in the current classroom for the instructional activities being conducted to the number of people the classroom holds. The degree of classroom utilization can be expressed as equation (11):

$$f_3 = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{r}\frac{N(i,j)}{capacityR_j} \tag{11}$$

where n denotes the full number of courses for all classes in the school, $capacity_{R_j}$ denotes the classroom capacity of the classroom occupied by the course, and $N(i,j)$ denotes the number of people in the class corresponding to the course i that is being taught in the R_j classroom. r denotes the number of classrooms and n denotes the number of all courses.

II. C. 4) Physical education course scheduling

Physical education courses require a lot of exercise, after meals and the physical state of inactive time period is obviously inappropriate, try to arrange the physical education courses in the second period of the morning or the second period of the afternoon, the five time periods of the day to set the weight of physical education courses, the average weight value of the scheduling of the physical education courses for all classes to measure the physical education courses, which can be expressed as the formula (12):

$$f_4 = \frac{1}{TY}\sum_{i=1}^{TY}\varepsilon_i \tag{12}$$

where TY denotes the number of physical education courses in all classes and ε_i denotes the weight corresponding to the i th physical education class.

Taking the above discussion together, the modeling of the school scheduling problem can be expressed as Equation (13):

$$\begin{cases} max, & f(x)=(f_1,f_2,f_3,f_4) \\ s.t., & \text{Satisfy hard constraints } 1\sim 5 \end{cases} \tag{13}$$

III. Construction of class scheduling model based on hybrid genetic-ant colony algorithm

III. A. Application of Genetic Algorithms to Scheduling Problems

Genetic algorithm is a type of heuristic algorithm [22], which simulates the natural evolutionary process of survival of the fittest and survival of the fittest in the biological world. Genetic algorithms start with a population in which potential solutions to a problem may be possessed, and each population consists of coded N individuals, each of which is a chromosomal entity with certain characteristics. In each generation, individuals with high fitness are selected for cultivation, and then genetic operators perform various operations in genetics, such as crossover, mutation, etc., to produce a new genetic population for reproduction of the next generation, and the result obtained after many evolutions can be regarded as a near-optimal solution to the problem.

To solve the class scheduling problem using genetic algorithm, the first thing is to establish a correspondence between the factors involved in the class scheduling process and the genetic algorithm operator, the specific correspondence is as follows:

(1) Chromosomes

Genetic algorithms need to encode the solution space to solve the problem, that is, the actual problem is converted into a chromosome genetic code. In the scheduling problem, a chromosome represents one possible scheduling solution.

(2) Genes

Genes are components of chromosomes. Teachers have a fixed curriculum, and a pair such as “teacher-curriculum” can be a gene.

(3) Initial population

Before executing the genetic algorithm, it is necessary to generate a population with a number of initial solutions based on the size of the solution to the scheduling problem.

(4) Fitness function

The fitness function is used to calculate the chromosome eigenvalues and also to measure the degree of merit of the class schedule.

(5) Basic operations

The basic operations of genetic algorithms include selection, crossover and mutation, by which the gene segments of chromosomes are changed.

III. B. Application of Ant Colony Algorithm to Scheduling Problems

III. B. 1) Modeling the Ant Colony Algorithm

Ant colony algorithms were first used to solve the TSP (Traveler's Problem), which has a wide range of representations [23]. The TSP is often described as [24]: Suppose a professor is going to give lectures in n cities, and he chooses a certain city as the starting point, then arrives at all the cities in turn, and finally returns to the starting city. If only one lecture is to be given in each city, how does the professor plan the order in which the lectures are to be given in order to minimize the time required for the lecture trip. Many problems can be solved analogously to the TSP solution, therefore, the TSP can be used to describe the mathematical model of the ACO algorithm. The following concepts are first introduced:

$$m=\sum_{i=1}^{n} b_i(t) \tag{14}$$

where m is the number of ants in the colony. n is the size of the problem (i.e., the number of cities to be reached). $b_i(t)$ is the number of ants located in i cities at time t.

$$\mu_{ij}(t)=\frac{1}{d_{ij}} \tag{15}$$

where d_{ij} is the distance between i city and j city. μ_{ij} is the heuristic function that represents the expected value of the ants from i to j city.

$p_{ij}^{k}(t)$ denotes the probability of ant k transferring from i to j at moment t, $\tau_{ij}(t)$ is the pheromone on the path (i,j) at moment t, and the ants $k(1 \le k \le m)$ are determining the next walking route based on the result of $\tau_{ij}(t)$, and will generally choose the route with high value of $\tau_{ij}(t)$. In order to avoid ants reaching a city repeatedly, the taboo table $tabu_k(1 \le k \le m)$ is used to record the cities that the ants pass by. Then the probability that ant k moves from city i to city j at moment t is:

$$p_{ij}^{k} = \begin{cases} \left[\tau_{ij}(t)\right]^{\alpha} \mu_{ij}(t)^{\beta} \Big/ \sum\limits_{seallowed} \left[\tau_{ij}(t)\right]^{\alpha} \mu_{ij}(t)^{\beta}, j \in allowed_k \\ Otherwise \end{cases} \tag{16}$$

where $allowed_k$ is used to record the cities that the ants have not yet passed through. α is the information heuristic factor, representing the effect of the amount of information on the ants when picking a route. β is the expectation heuristic factor, which represents the effect of the heuristic information on the ants when picking a route.

After the ants have traveled through all the cities, the pheromone content on each route will change somewhat, so the pheromone should be updated, and the algorithmic formula is as follows:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{17}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{18}$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} Q / L_k, If\ the\ k-th\ ant\ passes\ through \\ (i,j)\ between\ t\ and\ t+1 \\ 0, Otherwise \end{cases} \tag{19}$$

where ρ is the pheromone volatilization coefficient $(0<\rho<1)$ and $1-\rho$ represents the amount of pheromone left on the road. $\Delta\tau_{ij}(t)$ is the amount of pheromone added to the road from i to j city, with a starting value of 0. $\Delta\tau_{ij}^{k}(t)$ is the amount of pheromone left by ant k on the road from i to j city. Q denotes the pheromone concentration, expressed as a constant, and L_k is the distance of the route traveled by ant k.

III. B. 2) Dichotomous graphical model of the scheduling problem

The ACO algorithm can only solve the problem described by the graph structure, so the scheduling problem can be converted into a bipartite graph maximum matching problem [25]. The graph structure is given in the following equation:

$$G = (N, S, G) \tag{20}$$

where N is the set of vertices of the graph. S is the set of edges. G is the set of weights associated with the set of edges.

The course scheduling process involves five main factors, according to the analysis of the scheduling problem has been determined that the teacher, the course, the class is a one-to-one correspondence between the relationship, and now only need to be reasonable arrangements for the class time and classroom. Therefore, in the ant colony algorithm, the relationship between these five elements can be transformed into "course, teacher, class" relationship and "time, classroom" relationship, the scheduling problem is to solve these two relationships composed of bipartite graph maximum matching problem.

III. C. Application of Genetic-Ant Colony Hybrid Algorithm to Scheduling Problems

The execution steps of the hybrid genetic-ant colony algorithm are as follows [26]:

(1) Initialize the parameters of the genetic algorithm, randomly generate the initial population N, and set the initial number of iterations to 0.

(2) Calculate the fitness value of all individuals in the population.

(3) Perform the basic genetic operation for the individuals in the population, and the number of iterations is added to 1.

(4) If the termination condition is satisfied, continue the execution, otherwise jump to step (2).

(5) Transform the approximate optimal solution generated by the genetic algorithm into the initial pheromone distribution required by the ACO algorithm, and construct the bipartite graph model of the scheduling problem.

(6) Initialize the parameters of the ant colony algorithm and set the number of iterations $gen = 0$.

(7) Place m ants in the vertex set GLSC on the left side of the bipartite graph.

(8) Select a path for each ant based on transfer odds and scheduling constraints.

(9) Record m routes and complete the iteration.

(10) Calculate the length of m paths and record the current optimal path value.

(11) Update the amount of pheromone on the upper edge of the optimal path and volatilize the pheromone on the remaining edges.

(12) Determine whether the constraints of the termination condition are met, if so, stop the operation and solve for the current optimal path value, otherwise $gen = gen + 1$, return to step (7) and start a new iteration.

IV. Analysis of the effectiveness of scheduling

IV. A. Experimental design

IV. A. 1) Setting up the experimental environment

In order to verify the effectiveness of the algorithms proposed in this paper, some real data of undergraduate teaching at University A are used for testing in this paper. The experimental environment and the development tools used are as follows:

(1) Hardware environment: Intel i5-2450M+8G RAM

(2) Operating system: Windows 10

(3) Experimental environment: Eclipse

(4) Database tool: MySQL 5.6

(5) Development language: Java

IV. A. 2) Experimental data sets

The experimental data in this paper come from the undergraduate teaching data of the School of Civil Engineering of University A and the personnel training programs of some majors published by University A. These specialized courses are usually taught by two or three classes together. Each instructor can teach two or three classes, while each instructor can also teach multiple courses. In this paper, we use random assignment to determine the students for public courses. For specialized courses, on the other hand, we pre-assign multiple classes to attend. Finally, we set a series of constraints on the scheduling of classes so that the constraints can verify the effectiveness of the algorithm in this paper.

IV. A. 3) Evaluation indicators

The output of the algorithm proposed in this paper is the final class scheduling result. In order to verify the effectiveness of the final class scheduling results, this paper proposes some important evaluation indexes: the optimal solution adaptation, the average adaptation, and the average algorithm operation time. Among them, the optimal solution fitness refers to the fitness of the individual with the largest fitness in the population finally generated by the algorithm. This index directly reflects the quality of the algorithm. The larger the optimal solution fitness, the better the quality of the algorithm, and the better the effect of the final class schedule generated. Conversely, the worse the quality of the algorithm, the worse the final class schedule generated. The average fitness measures the algorithm from an overall perspective; the larger the average fitness, the better the optimization efficiency of the algorithm. Conversely, the worse the optimization efficiency of the algorithm. The average operation time of the algorithm reflects the time complexity of the algorithm to a certain extent, the longer the operation time, the more complex the algorithm. The ultimate goal of the scheduling algorithm is to be able to get the individual with a larger degree of adaptation in a certain time, i.e., to get a better scheduling result.

IV. B. Scheduling Performance Analysis

In the algorithm design of this paper, there are many influencing parameters involved, and we will focus on the influence of the genetic algebra on the scheduling algorithm. For this purpose, we set the value of the genetic algebra T to increment from 100 to 200, with each incremental value of 10. The optimal solution adaptation from the final population is observed and analyzed. The experimental results are shown in Table 1.

Table 1 shows that the optimal solution fitness in the final population increases with the increase in the number of genetic generations. This indicates that the population is evolving in a better direction as the number of genetic generations increases. The algorithm gradually gets better results. When the number of genetic generations reaches 170, the rate of increase of the optimal solution fitness of the final population slows down. This indicates that the algorithm's ability to search for better solutions in the solution space is weakened when the genetic number of the population reaches 170 generations. In addition, as the number of genetic generations increases, the computation time of the algorithm increases as well, and the algorithm needs more time to obtain a better solution. In this paper, the genetic generation size is set to 170, and it is considered that the algorithm has reached a better search of the solution space when the genetic generation reaches 170.

Table 1: Optimal fitness of initial population and final population under different generations

Genetic generation T	The optimal solution of the initial population	The optimal solution of the final population
100	0.3202296	0.3592167
110	0.3300736	0.3629442
120	0.3235988	0.3651229
130	0.3334436	0.3687945
140	0.3397791	0.3707533
150	0.3311693	0.372477
160	0.3280099	0.3733431
170	0.3208595	0.3775129
180	0.3208342	0.3776694
190	0.3234856	0.3775332
200	0.3376436	0.3779407

In order to verify the effectiveness of the scheduling algorithm proposed in this paper, the algorithm in this paper will be compared with the traditional genetic algorithm, simulated annealing algorithm and genetic simulated annealing algorithm. Considering the random searchability of the algorithm on the solution space, the experimental comparison will be analyzed by taking the average value through several experiments. In this paper, the number of experiments is set to be from 5 to 50, with an incremental value of 5 each time, and the scheduling effect of each algorithm is compared by the optimal solution fitness, the average fitness and the algorithm's average computing time. The experimental results are shown in Fig. 1, Fig. 2 and Fig. 3.

As can be seen from Fig. 1 and Fig. 2, the optimal solution adaptation of the genetic-ant colony algorithm in this paper is higher than that of other algorithms between 0.0363 and 0.0472, and the average adaptation of the algorithm in this paper is higher than that of other algorithms between 0.0402 and 0.0423. Therefore, both the optimal solution adaptation and the average adaptation, the algorithm of this paper produces better results than the results of other algorithms. Also, we can find that both the optimal solution fitness and the average fitness have different values with the number of experiments, which is because the algorithm in this paper is a randomized search algorithm according to certain rules. The algorithm searches the solution space in a different range and in a different order during each experiment. However, we can find that the optimal solution fitness and the average fitness, although different, will remain in a rough interval. This is because when the algorithm's search of the solution space reaches a certain degree, the better results obtained in the end are always similar.

As can be seen through Figure 3, the algorithm in this paper takes between 4224 and 8850 seconds more on average. Regardless of any number of experiments, the algorithmic average computing time of this paper's algorithm is greater than the average computing time of the other three algorithms. Compared with the traditional simulated annealing algorithm, the algorithm in this paper performs simulated annealing operations on multiple individuals in the population instead of a single individual. Compared with the traditional genetic algorithm, the algorithm in this paper is more complex computationally during the selection operation, and introduces an individual correction mechanism during the crossover and mutation operations. In addition, we have added individual optimization operation to the traditional genetic algorithm, considering both population evolution and individual optimization, so that the algorithm can get better results. Compared with the traditional genetic simulated annealing algorithm, the algorithm in this paper produces multiple individuals instead of a single individual during individual optimization. The idea of branch limiting is also used to prevent the number of individuals from being too large. Therefore, the algorithm in this paper is higher than the other three algorithms in terms of computing time, but the final results obtained are better than the results of the other algorithms.

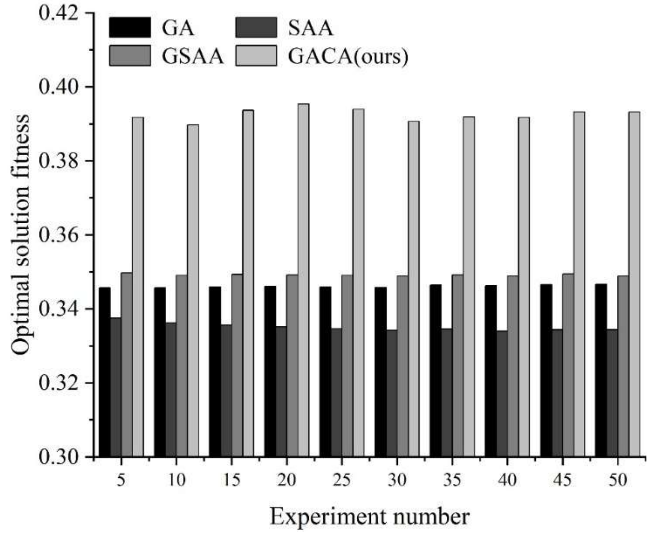

Figure 1: Optimal solution fitness of different number of experiments

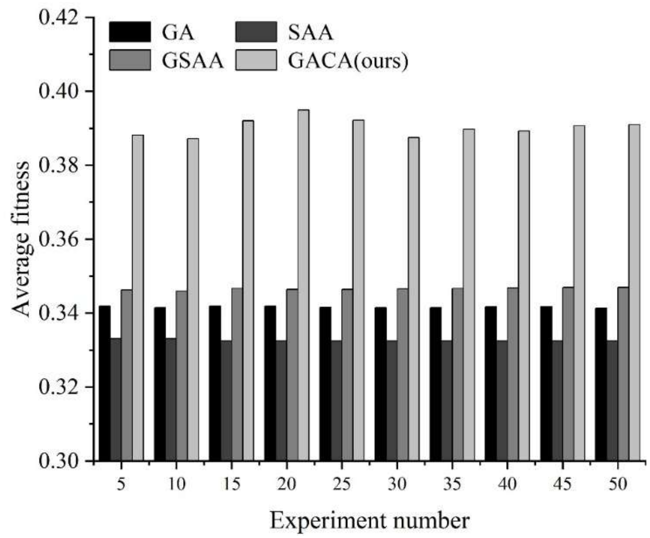

Figure 2: Average fitness of different number of experiments

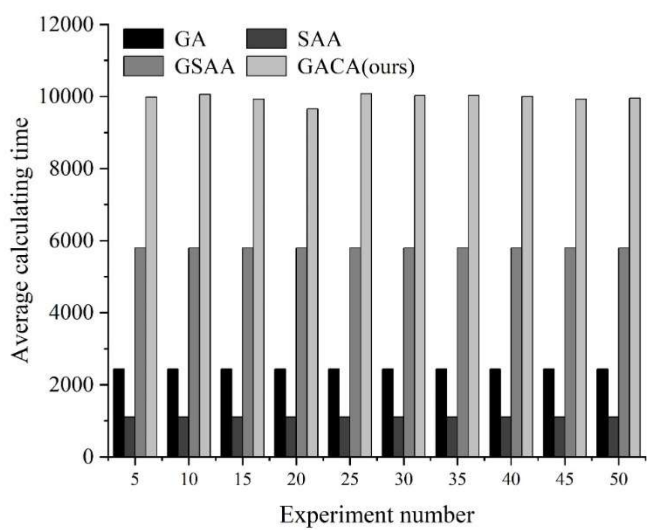

Figure 3: Average calculating time of different number of experiments (s)

IV. C. Analysis of scheduling effects

In this paper, the undergraduate programs in the School of Civil Engineering at University A are selected as the experimental subjects. The undergraduate teaching data of the School of Civil Engineering mainly includes 42 specialized course information, 62 teachers and 796 students. The raw data of this paper's experiments come from the undergraduate course database of the School of Civil Engineering of University A. Taking the undergraduate course scheduling data of the School of Civil Engineering of University A for many years as a reference, we carry out multifold extraction, conversion, and data cleaning in the database to complete the preprocessing operation of the data, and provide a data basis for the experiments of the genetic-ant colony scheduling algorithm in this paper.

IV. C. 1) Quality of scheduling

In this paper Genetic-Ant Colony Algorithm (GACA) on the evaluation of the effectiveness of scheduling, the quality of the scheduling scheme, in terms of section superiority, uniformity of class distribution, the degree of optimization of the course portfolio, classroom constraints, and consumed time, this paper Genetic-Ant Colony Algorithm (GACA) is compared to the original Genetic Algorithm (GA), Simulated Annealing Algorithm (SAA), Genetic Simulated Annealing Algorithm (GSAA) for comparison. The experimental results are shown in Table 2.

From Table 2, it can be seen that the section optimality, uniformity of class distribution, optimization of course portfolio, and teacher constraint satisfaction of the Genetic-Ant Colony Algorithm (GACA) in this paper are 22748, 9889, 9174, 7086, and 7074, which are higher than the original Genetic Algorithm (GA), Simulated Annealing Algorithm (SAA), and Genetic Simulated Annealing Algorithm (GSAA), respectively. Although the GACA algorithm consumes the longest time (6432s), the overall evaluation shows that the performance of the genetic-ant colony algorithm in this paper is better than the performance of other scheduling algorithms.

Table 2: Effect comparison of several different algorithms

	GA	SAA	GSAA	GACA (ours)
The maximum fitness of the population	11583	13856	15296	22748
Number excellence	9024	9268	9475	9889
Class distribution uniformity	7594	7868	8346	9174
Curriculum combinatorial optimization	5648	5873	6625	7086
Classroom constraint satisfaction	4869	5297	5886	7074
Consuming time/s	2785	3246	5876	6432

IV. C. 2) Efficiency of class scheduling

In terms of scheduling efficiency, experiments are conducted in terms of student selection satisfaction, classroom utilization, and overall scheduling rule satisfaction, and the results of the specific experimental data are shown in Table 3.

In Table 3, we select two different datasets to carry out experiments on the application of the genetic-ant colony algorithm in this paper, and the experimental results show that the satisfaction rate of students' class selection reaches 100%, the average value of the overall rule satisfaction rate is 96.4%, the classroom utilization rate reaches 88.1%, and the course homogeneity reaches 0.8, regardless of whether it is in the all-administrative class or the walk-in class.

Table 3: Experiment results in class scheduling efficiency

Dataset	Student selection satisfaction rate	Overall rule satisfaction rate	Classroom utilization rate	Class uniformity
Administrative class	100%	94.28%	91.27%	0.8
	100%	96.42%	89.74%	0.7
	100%	98.23%	86.43%	0.9
	100%	95.27%	88.54%	0.9
	100%	98.07%	90.99%	0.7
	100%	95.82%	87.52%	0.9
	100%	96.29%	86.49%	0.8
Walking class	100%	98.31%	86.99%	0.9
	100%	97.23%	84.73%	0.8
	100%	98.28%	90.09%	0.7
	100%	95.01%	90.01%	0.8
	100%	95.28%	87.08%	0.8
	100%	94.38%	85.45%	0.8

IV. C. 3) Scheduling can set rule satisfaction rate

In the scheduling class can be set rule satisfaction rate, the application of this paper's genetic-ant colony algorithm experimental results are shown in Table 4. Table 4 is a list of the scheduling satisfaction rate under the conditions of setting different scheduling rules, which basically meets the current rule constraints of intelligent school scheduling, the teacher's non-scheduling, the limitation of class time, cross-campus classes, two conflicts before and after the course, two consecutive and more courses, single and double-weekly class schedules, and combined class settings, etc., and the scheduling satisfaction rate reaches the average value of 99%.

Table 4: Experiment results in class scheduling settable rule satisfaction rate

Dataset	Teacher doesn't schedule	Teacher class restriction	Teacher mutex	Teachers cross the campus	Class before-after mutex
Administrative class	99.75%	NO	NO	NO	NO
	99.27%	NO	NO	NO	NO
	99.89%	NO	NO	NO	NO
	NO	NO	NO	NO	NO
	97.86%	99.88%	NO	NO	NO
	98.94%	99.37%	NO	NO	NO
	97.88%	NO	NO	NO	NO
Walking class	100%	NO	NO	NO	NO
	NO	NO	NO	NO	NO
	NO	99.03%	NO	NO	NO
	99.74%	NO	NO	NO	NO
	99.98%	NO	NO	NO	NO
	NO	NO	NO	NO	NO
Dataset	Class same-day mutex	Class time priority	Continuous class	Single-double week	Sum class
Administrative class	NO	97.29%	100%	NO	NO
	92.45%	93.23%	100%	NO	NO
	NO	86.91%	NO	100%	NO
	NO	96.85%	85.92%	NO	100%
	92.32%	91.32%	100%	100%	100%
	NO	98.69%	83.26%	NO	100%
	90.12%	81.54%	92.64%	NO	NO
Walking class	NO	NO	NO	NO	NO
	NO	87.82%	NO	NO	NO
	NO	97.18%	60%	NO	NO
	NO	98.83%	NO	NO	100%
	NO	98.02%	82.76%	NO	100%
	NO	89.02%	NO	NO	100%

V. Conclusion

The author researches the school scheduling problem and constructs a scheduling mathematical model. The genetic algorithm and ant colony algorithm are combined to propose a hybrid genetic-ant colony algorithm to optimize the school intelligent scheduling system.

The hybrid genetic-ant colony algorithm proposed in this paper has the best effect on the comprehensive optimization of intelligent school scheduling when the number of genetic generations reaches 170. Compared with the original genetic algorithm, simulated annealing algorithm, and genetic simulated annealing algorithm, the optimal solution fitness of the genetic-ant colony algorithm in this paper is higher by 0.0363-0.0472, and the average fitness is higher by 0.0402-0.0423, and although the genetic-ant colony algorithm in this paper consumes an average extra time by 4,224 to 8,850 seconds, the algorithm in this paper has the best performance for scheduling classes in general. The Genetic-Ant Colony algorithm is superior to other algorithms in scheduling in terms of section optimality, uniformity of class distribution, optimization of course combination, and satisfaction with teacher constraints, and it is slightly inferior to other algorithms in terms of average time consumed. The mean values of student selection satisfaction rate, overall rule satisfaction rate, classroom utilization rate, course uniformity, and scheduling satisfaction rate of the genetic-ant colony algorithm are 100%, 96.4%, 88.1%, 0.8, and 99%, respectively. The algorithm in this paper performs well on smart scheduling in schools.

References

[1] Liu, R., Shi, Y., Yi, B., Xu, Y., Lu, H., Wang, X., ... & Ji, C. (2020). Humanized computing for mass customization application in curriculum management. Mobile Networks and Applications, 25, 1484-1495.

[2] Tong, M. S., Xie, Y. M., & Wan, G. C. (2022, December). On the Problems and Optimization of Curriculum Arrangement in China's Universities. In 2022 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE) (pp. 765-768). IEEE.

[3] Junjun, Z., Hexia, Y., Oyam, D. M. A., & Yi, W. (2022). Design and Implementation of Intelligent Course Scheduling System for Deep Integration of Education and Teaching. Frontiers in Educational Research, 5(19).

[4] Yang, D. (2023). Scheduling Model Based on Maximum Satisfaction. Advances in Economic Development and Management Research, 1(1), 122-129.

[5] Shi, B., Aidana, Y., Xu, X., Wang, Y., & Yu, D. G. (2020, December). Systematic Arrangement of Courses to Improve students' Professional Innovation Capability. In 2020 6th International Conference on Social Science and Higher Education (ICSSHE 2020) (pp. 539-543). Atlantis Press.

[6] Raskina, I. I., Shtepa, J. P., Kurganova, N. A., & Korobeynikova, N. A. (2022). Student Learning Arrangement Based On The 'Flipped Classroom'Model'. European Proceedings of Social and Behavioural Sciences.

[7] Suo, X. F., Gao, Y. H., & Han, X. (2014). University course arrangement system based on improved ant colony algorithm design. Applied Mechanics and Materials, 651, 2536-2540.

[8] Zhu, K., Li, L. D., & Li, M. (2021). A survey of computational intelligence in educational timetabling. International Journal of Machine Learning and Computing, 11(1), 40-47.

[9] Sari, R., Ramdhania, K. F., & Purnomo, R. (2022). Team-Teaching-Based Course Scheduling Using Genetic Algorithm. PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic, 10(1), 55-66.

[10] Alghamdi, H., Alsubait, T., Alhakami, H., & Baz, A. (2020). A review of optimization algorithms for university timetable scheduling. Engineering, Technology & Applied Science Research, 10(6), 6410-6417.

[11] Chen, Y., Bayanati, M., Ebrahimi, M., & Khalijian, S. (2022). A novel optimization approach for educational class scheduling with considering the students and teachers' preferences. Discrete Dynamics in Nature and Society, 2022(1), 5505631.

[12] Rezaeipanah, A., Matoori, S. S., & Ahmadi, G. (2021). A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search. Applied Intelligence, 51, 467-492.

[13] Xiao, J. (2024). The application of greedy algorithm in the course selection and scheduling system of college students. Applied and Computational Engineering, 75, 48-53.

[14] Nugroho, A. K., Permadi, I., & Yasifa, A. R. (2022). Optimizing course scheduling faculty of engineering unsoed using genetic algorithms. JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer), 7(2), 91-98.

[15] Ding, C., Chen, L., & Zhong, B. (2019). Exploration of intelligent computing based on improved hybrid genetic algorithm. Cluster Computing, 22(Suppl 4), 9037-9045.

[16] Zhang, Q. (2022). An optimized solution to the course scheduling problem in universities under an improved genetic algorithm. Journal of Intelligent Systems, 31(1), 1065-1073.

[17] Wen-jing, W. (2018). Improved Adaptive Genetic Algorithm for Course Scheduling in Colleges and Universities. International Journal of Emerging Technologies in Learning, 13(6).

[18] Chen, X., Yue, X. G., Li, R., Zhumadillayeva, A., & Liu, R. (2021). Design and application of an improved genetic algorithm to a class scheduling system. International Journal of Emerging Technologies in Learning (iJET), 16(1), 44-59.

[19] Han, X., & Wang, D. (2025). Gradual Optimization of University Course Scheduling Problem Using Genetic Algorithm and Dynamic Programming. Algorithms, 18(3), 158.

[20] Haider, S. A., Ahmad, K. M., Zahid, A., AlGhamdi, A., Keshta, I., & Soni, M. (2024, October). Genetic and Deep Reinforcement Learning-Based Intelligent Course Scheduling for Smart Education. In Proceedings of the 2024 International Conference on Artificial Intelligence and Teacher Education (pp. 117-124).

[21] Xiang, K., Hu, X., Yu, M., & Wang, X. (2024). Exact and heuristic methods for a university course scheduling problem. Expert Systems with Applications, 248, 123383.

[22] S. Bagheri,H. Khalafi & A. A. Bahrami. (2025). An Optimal Radiation Shield Design of a Small Modular Nuclear Power Reactor by Genetic Algorithm. Nuclear Technology,211(5),940-952.

[23] Medhat A. Tawfeek,Ibrahim Alrashdi,Madallah Alruwaili,Leila Jamel,Gamal Farouk Elhady & Haitham Elwahsh. (2025). Improving energy efficiency and routing reliability in wireless sensor networks using modified ant colony optimization. EURASIP Journal on Wireless Communications and Networking,2025(1),22-22.

[24] Petr Stodola & Radomír Ščurek. (2025). Using machine learning in combinatorial optimization: Extraction of graph features for travelling salesman problem. Knowledge-Based Systems,314,113216-113216.

[25] Christos Papadimitriou,Tristan Pollner,Amin Saberi & David Wajc. (2024). Online Stochastic Max-Weight Bipartite Matching: Beyond Prophet Inequalities. Mathematics of Operations Research,49(3),1607-1607.

[26] Xie Xiaolin,Yan Zixiang,Zhang Zhihong,Qin Yibo,Jin Hang & Xu Man. (2024). Hybrid genetic ant colony optimization algorithm for full-coverage path planning of gardening pruning robots. Intelligent Service Robotics,17(3),661-683.