Hindawi
Security and Communication Networks
Volume 2023, Article ID 9865853, 1 page
https://doi.org/10.1155/2023/9865853

Retraction

Retracted: Design of College Scheduling Algorithm Based on Improved Genetic Ant Colony Hybrid Optimization

Security and Communication Networks

Received 11 July 2023; Accepted 11 July 2023; Published 12 July 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] T. Li, Q. Xie, and H. Zhang, "Design of College Scheduling Algorithm Based on Improved Genetic Ant Colony Hybrid Optimization," *Security and Communication Networks*, vol. 2022, Article ID 2565639, 13 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 2565639, 13 pages
https://doi.org/10.1155/2022/2565639

Research Article

Design of College Scheduling Algorithm Based on Improved Genetic Ant Colony Hybrid Optimization

Ting Li,[1] Qiang Xie,[2] and Hua Zhang[3]

[1]*Center for Evaluation and Faculty Development, Hunan First Normal University, Changsha, Hunan 410205, China*
[2]*Academic Affairs Office, Hunan First Normal University, Changsha, Hunan 410205, China*
[3]*School of Computer Science and Engineering, Hunan University of Information Technology, Changsha, Hunan 410151, China*

Correspondence should be addressed to Hua Zhang; zhanghua@hnuit.edu.cn

Received 18 April 2022; Revised 16 May 2022; Accepted 30 May 2022; Published 14 June 2022

Academic Editor: Chin-Ling Chen

With the gradual expansion of college scale, the professional categories in colleges and universities are becoming more and more complete, and the volume of courses is becoming more and more huge. In the meantime, the number of students is growing by leaps and bounds, and the teaching resources are subject to more and more complicated teaching tasks. The workload and the difficulty of scheduling in teaching management are also on the rise year by year. This paper proposes a design of a college scheduling algorithm based on an improved genetic ant colony hybrid optimization algorithm. Firstly, the fitness-enhanced elimination law is proposed to improve the selection process of traditional genetic algorithms. Subsequently, the gene infection crossover method is proposed to ensure the increase of the average fitness value in the evolutionary process. Next, the unnecessary replication operation in the traditional genetic algorithm is removed to enhance the operation speed of the algorithm. Finally, the parallel mechanism of fuzzy adaptive is introduced to improve the convergence and stability of the algorithm. For the ant colony optimization algorithm, a nonuniform pheromone distribution is used according to the position of the current raster relative to the starting point, which makes the initial pheromone concentration of the dominant raster higher and avoids blind search by ants. The ant movement rules are redefined by the directional neighborhood expansion strategy to further shorten the path. The experimental results indicate that the hybrid optimization algorithm outperforms other algorithms in terms of performance in terms of scheduling success and scheduling time, and it can be applied in practical scheduling because of the high quality of courses schedule.

1. Introduction

Education is a fundamental part of the sustainable development of the country and society, and it is also a long-term basic national policy of China to develop the country through science and education. Education is an important means to improve the education level of society and the quality of the nation, especially higher education. Hence, China's colleges have been expanding their enrollment scale each year and continuously increasing the number of people receiving higher education [1, 2]. In the meantime, the level of scientific research and management is constantly improved and gradually transformed from quantitative change to qualitative change. The process of education and training of talents with high precision technology and skills is long and strict, and each link needs reasonable planning and arrangement. With the continuous refinement of college majors, colleges and universities are constantly developing and innovating in the talent training system. In terms of course categories and link settings, they all strive to cultivate personalized talents with both professional depth and professional span, which undoubtedly has new requirements and challenges for the scheduling work in colleges and universities [3–5]. In-depth research and discussion on the scheduling system of colleges and universities is the basic work of implementing the national talent cultivation strategy and plays a significant role in the teaching management of colleges and universities.

Scheduling work is essentially the optimal use of various teaching resources in universities to ensure that basic teaching can be carried out in an orderly manner [6]. It is a proposition that needs to solve how to arrive at an optimal solution under multidimensional constraints. That is, various constraints such as time, space, and people need to be satisfied simultaneously in the system design to ensure that no conflicts arise between various constraints in the process of classroom teaching [7]. Avoiding various types of conflict problems such as the same batch of students needing to appear at different teaching sites at the same time or a particular classroom teacher being assigned different teaching tasks at the same time results in the classroom teaching work not being carried out properly [8, 9]. The more humane demands such as not too many hours of classes in a day for one teacher and not too many hours of classes in the same course for the same batch of students are also a challenge to the newer scheduling work. At present, the traditional manual scheduling method commonly used in domestic universities is mainly based on the previous scheduling experience. However, because the traditional manual scheduling method does not have complete systematic theoretical support, let alone data modeling by computer, it often has disadvantages in actual operation such as multiple involvement, large communication volume, slow speed, low efficiency, and error-prone and strong subjective consciousness. This approach is obviously no longer applicable to contemporary higher education institutions. For multiple departments and multiple course categories (basic courses, specialized courses, compulsory and elective courses, etc.), a large number of students need to complete various types of teaching schedule arrangements each semester. If the course arrangement is carried out manually, the workload is so large that it will definitely take a lot of effort and time of teaching administrators, and it is difficult to give a better schedule that is reasonable and feasible and can make full use of teaching resources [10, 11].

Multidimensional information collection and high-speed data processing are realized and gradually popularized and applied currently. Before the scheduling process in colleges and universities, the courses are mathematically modeled, and constraints such as courses, instructors, teaching locations, and number of students are set, and computer algorithms are used to complete the solution in order to realize computer-aided decision-making automated scheduling. It not only solves the rationality and feasibility of course arrangement in space and time and improves the efficiency of scheduling work, but also enhances the level of informationalization of teaching management, which is of great importance.

Algorithms for computer-aided decision scheduling systems are commonly used in domestic universities presently. These methods include graph coloring [12], which transforms the university course scheduling problem into a graph with vertices representing courses and edges representing constraints. The number of colors corresponds to feasible time slots. The graph coloring method assigns a finite number of colors to vertices, and no two adjacent vertices that are linked by an edge are of the same color. Genetic Algorithms (GA) and improved GA [13–15] assess course satisfaction with penalty function by genetically coding course scheduling constraints. Linear programming [16], simulated annealing (SA) algorithm [17], taboo search (TS) algorithm [18], etc., these algorithms are very popular now. The shortcoming of these algorithms is that they have difficulty in dealing with the constraints of the course scheduling process and can only produce feasible solutions with unsatisfactory results. There are also course scheduling methods on deep learning [19, 20], but the computational complexity of deep learning is high and takes longer time.

Several studies have shown that hybrid algorithms show better results in solving college course scheduling problems. Integration of local search algorithm (LS) into particle swarm optimization algorithm (PSO) [21, 22] was used to construct the optimal solution for college course scheduling. Constraint propagation is integrated with genetic algorithm to obtain the approximate optimal solution for college course scheduling. The disadvantage of genetic algorithm is the long computational time. The particle swarm optimization algorithm converges faster than the genetic algorithm and does not require much parameter adjustment. PSO alone cannot solve the constraint satisfaction problem. The curriculum scheduling problem is a constraint satisfaction problem, so it is necessary to find a way to deal with the constraint conflicts in the control curriculum scheduling problem.

This paper proposes an improved genetic ant colony based hybrid optimization algorithm for college scheduling. The algorithm gives constraints on the scheduling problem and applies the improved genetic ant colony hybrid optimization algorithm to solve the scheduling problem. The empirical verification demonstrates that the proposed algorithm achieves good results in practical scheduling problems.

Section 2 of this paper is the state of the art of the problem of scheduling. Section 3 is the methodology of proposed algorithm. Section 4 is the implementation of the scheduling algorithm. Section 5 is the result analysis and discussion, and Section 6 is the conclusion.

2. State of the Art

The core task is to arrange courses, classes, teachers, and classrooms without conflicts in each lecture period and to ensure that they meet the constraints set by teachers in advance. The introduction of the class system in universities has made the scheduling problem more complicated. The constraints of class scheduling are further increased, and the lack of teaching resources in schools is further highlighted.

The scheduling problem is described as follows: the courses set $SC = \{s_1, s_2, \cdots, s_s\}$. The class set $CC = \{c_1, c_2, \cdots, c_c\}$. The set of teachers $TC = \{n_1, n_2, \cdots, n_n\}$. The classroom set $RC = \{r_1, r_2, \cdots, r_r\}$. The time period set $PC = \{u_1, u_2, \cdots, u_u\}$. Before scheduling a course, it is necessary to set up a schedule for different grades and classes and to determine the relationship between courses, classes, teachers, and classrooms.

There are two types of constraints to consider when scheduling classes. The hard constraints are the conditions that must be followed in the scheduling process, and the schedule can only be arranged in accordance with the hard constraints to ensure that the scheduling resources do not conflict with each other, as follows:

(1) Only 1 course can be scheduled for the same class in the same teaching period:

$$\sum_{w=1}^{CC}\sum_{x=1}^{PC}\sum_{y=1}^{SC} c_w u_x s_y n_t r_z \leq 1. \tag{1}$$

Formula (1) indicates that in the same class c_m in the same class period p_i only 1 course can be scheduled at most s_y, by the teacher n_t in the classroom r_z classroom.

(2) Only a maximum of one course can be scheduled in the same classroom during the same teaching period:

$$\sum_{z=1}^{RC}\sum_{x=1}^{PC}\sum_{y=1}^{SC} c_w u_x s_y n_t r_z \leq 1. \tag{2}$$

Formula (2) denotes the same classroom r_z in the same lecture period u_x only 1 course can be scheduled at most s_y, by the teacher n_t in the class c_w class.

(3) A maximum of 1 course can be scheduled by the same instructor in the meantime, i.e.,

$$\sum_{t=1}^{TC}\sum_{x=1}^{PC}\sum_{y=1}^{SC} c_w u_x s_y n_t r_z \leq 1. \tag{3}$$

Formula (3) indicates that the same instructor n_t in the same class period u_x can only be assigned at most 1 course s_y and in the classroom r_z for the class c_w classroom.

Soft constraints are nonmandatory rules before scheduling a class. These rules are not necessary to be met, but they can have a significant impact on the rationality of the schedule and user satisfaction, as follows:

① The weekly class schedule of the same course is spread out as much as possible

② If a course is scheduled to have a priority, the course should be scheduled in the session that has the highest priority, such as the main course in the morning

③ Teacher continuity setting is the maximum number of consecutive lessons that can be taught by one teacher

④ Some courses are offered and others are not scheduled afterwards

3. Methodology

3.1. Enhanced Infection Genetic Algorithm

3.1.1. Enhancement of the Law of Elimination. Conventional genetic algorithms have small differences in fitness values among individuals in the late evolutionary stage, and the selection process is weakly competitive, resulting in stagnant population evolution, low accuracy in finding superiority, and slow convergence. After obtaining the individual fitness value, the average fitness of the population is calculated and then the average fitness of the individual and the population is compared. This operation enables quickly eliminating more individuals with low fitness in the early stage of population evolution. The culling criterion varies with the number of generations. At later stages of evolution, this enhances competition among similar individuals and prevents the population from stagnating. The mathematical description of the fitness value enhanced elimination law is as follows.

Let there be n individuals in a population, and the phenotype of the xth individual in the ath generation is x and the fitness value is $f_a(i_x)$. The mean fitness value of the ath generation is $1/t\sum_{x=1}^{t} f_a(i_x)$. The recalculated fitness value of the xth individual is

$$f_t(i_x) = \begin{cases} \left(f_a(i_x) - \dfrac{1}{t}\displaystyle\sum_{x=1}^{t} f_a(i_x)\right)^z, f_a(i_x) > \dfrac{1}{t}\displaystyle\sum_{x=1}^{t} f_a(i_x), \\ 0, f_a(i_x) \leq \dfrac{1}{t}\displaystyle\sum_{x=1}^{t} f_a(i_x), \end{cases} \tag{4}$$

where z is the reinforcement competition coefficient. z directly affects the evolution speed and the accuracy of the algorithm for finding the best performance; in order to study the influence of z value on the value of individual fitness, the curves are plotted based on $j = i^z\ (i \geq 0)$ function based on $z = 0.3, 0.2, 1, 2, 3$, respectively.

From the experimental results it is clear that the function y is convex when $0 < z \leq 1$. And as the value of k decreases, the curve trend tends to flatten. When $z > 1$, the function y is concave and the curve tends to steepen as the value of z increases. It can be seen that, for the enhanced elimination law, a smaller value of z decreases the difference in fitness values between individuals and retards population evolution. A larger value of z increases the difference in fitness values between individuals and accelerates population evolution. Therefore, the larger the z value, the greater the difference in fitness values among similar individuals at the later stage of population evolution, the more intense the competition, and the better the search accuracy.

3.1.2. Genetic Infection Crossover Method. Conventional genetic algorithms use crossover and mutation for random search. When the chromosomes of two individuals are crossed over to produce new individuals, this approach is likely to produce offspring with lower fitness values when the fitness value of the parent individual is larger, thus reducing the average fitness of the population and deviating from the optimal solution. In contrast, gene infection crossover effectively prevents the reduction of fitness of new individuals by replacing most of the genes of the

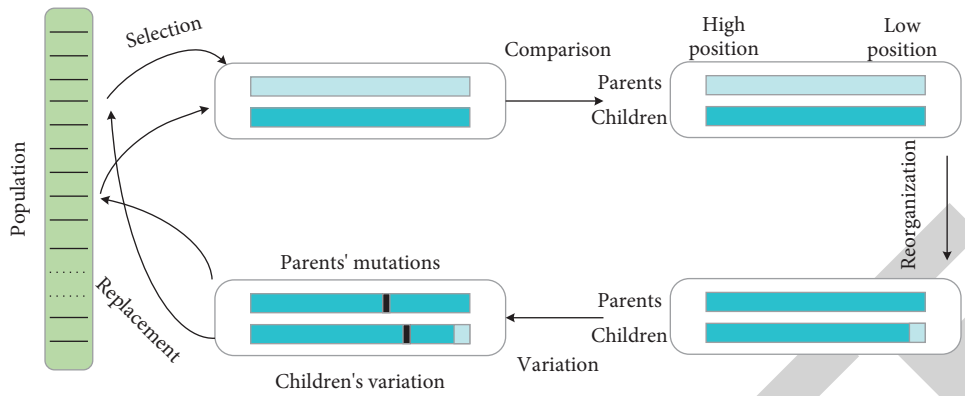

Figure 1: Genetic infection cross-operation.

parent with those of individuals with low fitness. As shown in Figure 1, the gene infection crossover method is described as follows:

Step 1. The fitness values of individuals are recalculated using the augmented elimination law such that individuals with lower fitness are not selected when the selection operation is performed.

Step 2. For each individual in the selected population, a parent is selected according to the principle of proportional selection, and the fitness of the individual is compared with the fitness of the parent. When the individual's fitness is less than the parent's fitness, a lower point in the parent's chromosome is selected and the corresponding gene segment in the individual's chromosome is replaced with the higher gene infection before this point. When the individual's fitness is greater than the parent's, no further crossover of gene infection is performed.

In the linear problem, when the parent replaces most of the high genes with infected individual genes, the individual's phenotype will rapidly approach the parent, preventing the generation of individuals with lower fitness values and thus preventing the reduction of the average fitness value of the population. Moreover, by retaining their own low genes, individuals retain a certain search ability and are able to search near the parent's phenotype, increasing the probability of searching for the optimal solution. In contrast, in nonlinear problems, a search based solely on the parent's phenotype can easily lead to a local optimal solution. Therefore, when performing gene crossover, the principle of proportional selection is applied to each individual to select the parent to ensure that the parental phenotypes can be adequately selected and compared. This ensures the speed of search, while preserving the diversity of parental expressions and preventing the emergence of local optimal solutions.

3.1.3. Enhanced Infection Genetic Algorithm

Step 1: the algorithm first sets parameters such as crossover rate and variance rate and initializes the population. Initialize the population; i.e., based on the selected coding method, a coding set with completed individual coding is generated based on the selected population size as the initial value for the algorithm to run. The commonly used encoding methods are binary encoding and gray code encoding.

Step 2: calculate the new fitness value. That is, the fitness value is calculated based on the original fitness function, and then the fitness value of individuals in the population is updated by the enhanced infection rule.

Step 3: determine whether the conditions for stopping evolution are satisfied. When the conditions for stopping evolution are satisfied, exit the program and give the result; otherwise proceed to the next step. Commonly used control conditions of the algorithm are evolutionary algebra, planning error, etc.

Step 4: perform gene infection crossover operation, and after that, perform mutation operation to move to step 2. Using the principle of proportional selection to crossover individuals of the population, the evolutionary speed is enhanced while ensuring the computational accuracy.

Compared with the conventional genetic algorithm, the enhanced infection genetic algorithm is improved in two main aspects: first, after obtaining the fitness value of each individual, the fitness value is recalculated according to the enhanced elimination law. It enhances the competition of different genes, increases the elimination of individuals with small fitness values in the early evolutionary stage, and speeds up the convergence rate. It increases the competition of similar chromosomes in the late evolutionary stage and improves the convergence accuracy. Second, by improving the crossover method of gene infection, it ensures that the fitness value of offspring individuals is higher than that of their parents, avoiding the generation of individuals with lower fitness values and preventing the decrease of the average fitness value of the population.

3.2. Adaptive Parallelism Mechanism. The introduction of parallel mechanism in genetic algorithm can effectively combine the natural parallelism of GA and the fast concurrency of computer, which is an important direction to improve the performance of algorithm proposed in recent

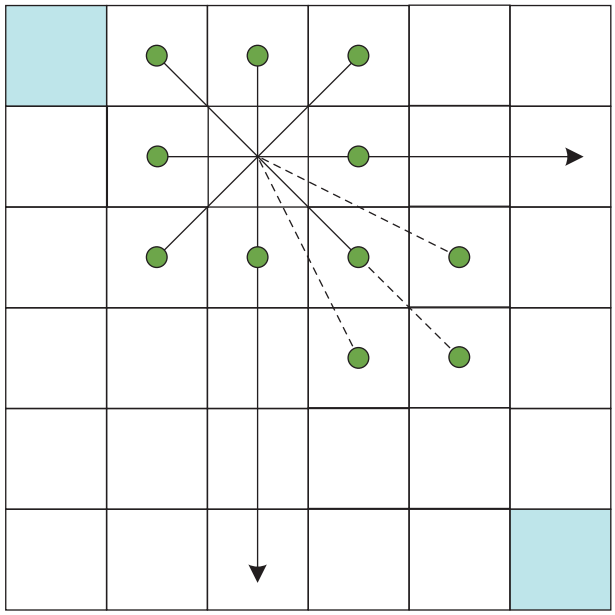

Figure 2: Feasible raster after directional neighborhood expansion.

years. Currently, there are four basic types of parallel mechanism models for genetic algorithms: Master-slave model, fine-grained model, coarse-grained model, and hybrid model. The parallel mechanism proposed in this paper adopts a coarse-grained model.

3.2.1. Evolutionary Strategies for Parallel Mechanisms. The existing genetic algorithm improvements can be summarized as coding, microgenetic strategy (improvement of genetic operator and parameter selection), and macrogenetic strategy (improvement of algorithm operation mechanism). In this paper, macroscopic multigroup parallelism mechanisms are studied rather than specific evolutionary strategies. Specifically, the larger value of P_m and the smaller value of P_c are called the exploration strategy, and the genetic operation is chosen to perform the mutation operation first and then the crossover operation, and finally the selection replication operation is performed. This evolutionary strategy facilitates the population to jump out of the local optimum and explore the new solution space. The smaller value of P_m and the larger value of P_c are called development strategy. This strategy first performs crossover operations, then mutation operations, and finally selective replication operations. This strategy can recombine existing genes to produce new individuals in anticipation of better trait changes. The normal strategy uses P_c and P_m, which are in between the developmental strategy and the exploration strategy, and uses the GA selection replication, crossover, and mutation operations in the evolutionary operation, and this strategy has the functions of both. In addition, the parallel mechanism of the evolutionary strategy of GA uses the public population as a container to continuously receive outstanding individuals passed from other populations and further develop them. First, the definition of evolutionary potential is given:

Definition 1. The evolutionary potential of individual x_i exhibited by local search is defined by formula (5):

$$V(i_x) = \omega\left(1 - Z^{\lambda}\right)\left(f'(i_x) - f_{\max}\right), \tag{5}$$

where ω denotes the weight of evolutionary potential. Z denotes the temperature decay coefficient. λ denotes the number of new solution searches. $f'(i_x)$ denotes the optimal solution of chromosome x_i after local search. $f_{\max}$ denotes the population fitness maximum. The evolutionary potential of an individual is related to the number of searches for new solutions and the fitness of the optimal solution. The lower the number of searches, the higher the evolutionary potential.

The public population first performs crossover operations on the best individuals. After the crossover is finished all individuals are searched and evolutionary potential is calculated according to the local search strategy, and then the fitness is updated according to formula (6). Finally, the chromosomes with the population size number are selected according to the fitness into the next generation.

$$f(i_x) = f(i_x) + V(i_x). \tag{6}$$

The purpose of a public population is to enable the population to explore the solution space around individuals, to explore individuals with evolutionary potential, and to discover new solutions. Individuals with high evolutionary potential have a higher probability of being selected for the next generation. On the one hand, it can enrich the population diversity and maintain the search range of the population, and on the other hand, crossover with other good individuals can hopefully produce new and better individuals.

3.2.2. Probability of Adaptive Determination Strategy Transformation. If the population evolution process is in stagnation or caught in the local optimal solution, the evolutionary strategy should be changed according to the population evolution. The evolutionary algebra A_f, in which the optimal solution is maintained constant, is introduced to measure the evolutionary state of the population, and the probability of the population changing its strategy is calculated by the following formula:

$$U_{cb} = \frac{A - A_n}{A} - \frac{1}{1 + exp\left[\beta\left(2A_f/A_{\max} - 1\right)\right]}, \tag{7}$$

where A_n is the current evolutionary generation. A is the total evolutionary generation. β is the control parameter with a value of 6 in the text. A_f indicates the number of generations of population stagnation. $A_{\max}$ is the maximum number of generations of stagnation.

The nonlinear probability can more reasonably control the population to change the evolutionary strategy. U_{cb} grows slowly in the early stage of evolutionary stagnation, allowing some time for the population to escape from stagnation by relying on the existing evolutionary strategy. As the number of generations of stagnation increases, the algorithm determines that the population is in stagnation.

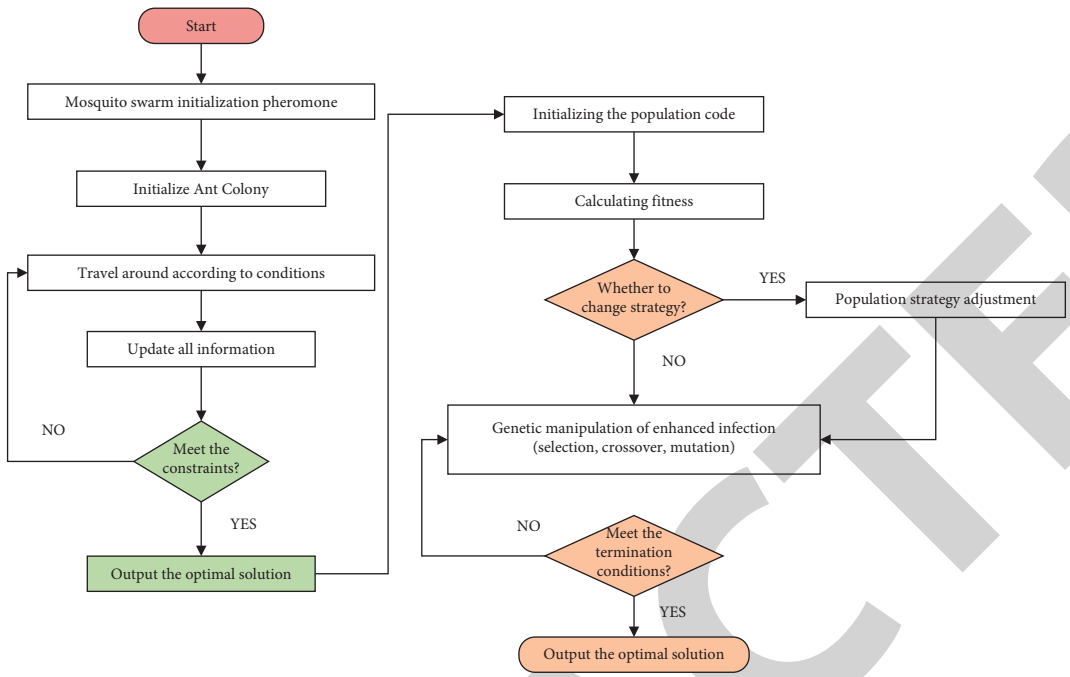

Figure 3: Improved genetic ant colony algorithm flow.

Compared with the probability of linear control, U_{cb} is significantly improved to avoid the population wasting time on the existing evolutionary strategy. Finally, to maintain a balance between population exploration and algorithm convergence rate, U_{cb} shows a linear decrease with further increase in evolutionary generations when the number of stagnation generations is constant.

3.2.3. Evolutionary Strategy of Fuzzy Inference. This paper draws on the idea of fuzzy control and applies fuzzy inference to adjust the evolutionary strategy of the population according to the individual variability and evolutionary status of the population. Formula (8) is used to represent the evolutionary status of the population.

$$E_1 = \frac{f_{\max} - f_{avg}}{f_{\max}} \in [0, 1]. \tag{8}$$

The size of $f_{\max} - f_{avg}$ is often used to measure the degree of evolution of a population. The method is indeed valid. However, the shortcoming is that the fitness function is designed based on a specific problem and its value varies with the specific problem. Moreover, during the calculation of the algorithm, the maximum and minimum fitness of the population also change continuously, so it is difficult to determine the appropriate threshold to judge the evolutionary degree of the population. The smaller the value of E_1, the closer the average fitness of the population to the optimal fitness of the population, which means that the population is more evolved. On the contrary, it means that the population is less evolved. Formula (9) indicates the variability of the population individuals.

$$E_2 = \frac{1}{T} \sum_{x=1}^{T} \frac{f(i_x) - f_{\min}}{f_{\max} - f_{\min}} \in [0, 1]. \tag{9}$$

The smaller the value of E_2, the more dispersed the population adaptation, the greater the population variation, and the better the diversity, and vice versa, the poorer the population diversity.

In formulae (8) and (9), $f(i_x)$ denotes the fitness of individual i_x. $f_{\max}$ is the maximum value of population fitness. $f_{\min}$ is the minimum value of population fitness. f_{avg} is the average value of population fitness. T is the number of population chromosomes.

Crossover arithmetic is used to obtain superior individuals by recombinant development of existing genes. If all individuals in the population do not have a certain gene, the missing gene cannot be obtained by crossover arithmetic in any way. If the population only generates new genes without further exploration, the good individuals in the population will be destroyed continuously, and convergence will be delayed. When the population is in a stagnant state, the population convergence is higher if the population is more concentrated in terms of adaptation. This will lead to more similarity between chromosomes and poor population diversity. In this case, it is necessary to replace the evolutionary strategy that can generate new genes to expand the population diversity. If the population fitness is more dispersed, the population convergence is low, the similarity between chromosomes is low, and the genes of chromosomes are more abundant in the population. At this time, the main focus should be on exploring the solution space of the current individual and the need to replace the evolutionary strategy that can exploit the existing genes.

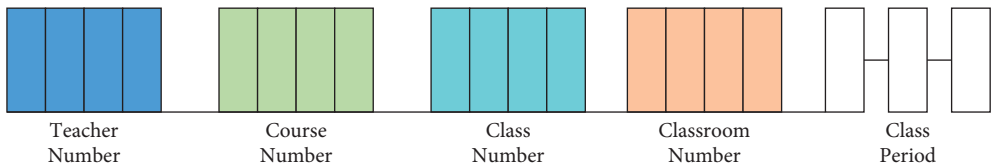

Figure 4: Example diagram of gene structure.

3.3. Multistrategy Ant Colony Algorithm

3.3.1. Initial Pheromone Nonuniform Distribution Strategy. In the solution process, one of the main bases for path selection by ants is the pheromone content of intergrid paths. In the original ant colony algorithm, the initial pheromone of each intergrid path is uniformly distributed and is a constant value. The value of the heuristic function of the ants will determine the magnitude of the state transfer probability, which is known by its calculation method. The difference in the heuristic function values of different grids is small, and without pheromone guidance, the blindness of the ant search is large, and the quality of the solution obtained from the search is low. Nonuniformization of the initial pheromone of the intergrid path is beneficial to accelerate the convergence speed of the algorithm. The heuristic function idea of A^* algorithm is used to differentially process the pheromones of different interraster paths. The corresponding pheromone concentrations are assigned according to their coordinates, and the calculation process is as follows:

$$\begin{aligned} \tau_{xy}(0) &= \tau_0 + \Delta\tau_{xy}, \\ \Delta\tau_{xy} &= c\left(\frac{1}{f(x)}\right)\cdot\left(\frac{1}{f(y)}\right), \\ f(i) &= \varepsilon\cdot a(i) + (1-\varepsilon)\cdot b(i), \end{aligned} \tag{10}$$

where τ_0 is the original initial pheromone value. $\Delta\tau_{xy}$ is the additional pheromone added to the (x, y) path. c is a constant value, taken according to the empirical value and the raster scale. ε is the weighted value. $a(i)$ is the spacing between the starting point and the current point. $b(i)$ is the spacing between the current point and the end point.

Based on the shortest distance between two points, the closer the ant's path is to the starting and ending line, the better the path is. When the next grid to be selected is closer to the end point, the path is better. Therefore, ε should be taken as a small value so that the end point plays a dominant role. Besides, for the overall effect of the algorithm, if the gap between the nonuniform pheromone values is too large, it will cause the algorithm to converge too early and lead to local optimum problems, and if it is too small, the strategy will have limited effect. Therefore, the values of c and ε were experimentally compared several times, and the final values were determined as $c = 2\times4\times4 = 32$, $\varepsilon = 0.1$.

3.3.2. Directional Neighborhood Expansion Strategy. In the original raster map-based algorithm, ants can only select the next path node in four or eight neighboring raster grids around them. The search direction is limited, and the optimal path length found due to the step length limitation is also long. The idea of this extended neighborhood is introduced, and the neighborhood is extended directionally according to the relative positions of the starting and ending points of the map, as shown in Figure 2.

The solid line represents the feasible raster of the original ant colony algorithm. The dashed line represents the feasible grid added after the directed neighborhood expansion. Using the directed neighborhood expansion strategy not only enriches the search direction of ants and achieves the purpose of finding shorter paths through one search, but also reduces the computational effort of the algorithm and speeds up the operation of the algorithm. Due to the increase of the searchable grid range of ants in the improved algorithm, the search method of ants is reset. In the eight surrounding grids of the current grid, ants may not select an obstacle grid (hard constraint) as the next move path node in the extended neighborhood grid.

3.4. Genetic Ant Colony Hybrid Optimization Algorithm Process. The improved genetic ant colony algorithm flow is given in Figure 3. By using the ant colony algorithm, a better set of solutions is generated after one iteration, which is used as the initial population of the genetic algorithm. This is an effective way to reduce the number of times the genetic algorithm seeks to find the optimal course scheduling result quickly.

4. Course Scheduling Implementation

4.1. Gene Coding and Chromosome Construction

(1) The teacher number, course number, class number, classroom number, and class period form a tuple, which is the gene of the genetic algorithm. Each gene can be regarded as a classroom unit in the class schedule, and the gene structure is shown in Figure 4. For example, in gene code 1001_2001_3001_4001_0-1-0, it means that teacher 1001 is teaching class 3001 in classroom 4001 during lesson period 0-1-0. The course number is 2001. u_x = (day, division, section). The day indicates the day of the week; division indicates the hour (AM, PM); section indicates the class period.

In practice, the instructor needs to set up the schedule for each semester before scheduling; i.e., the relationship between courses, classes, teachers, and classrooms is determined.

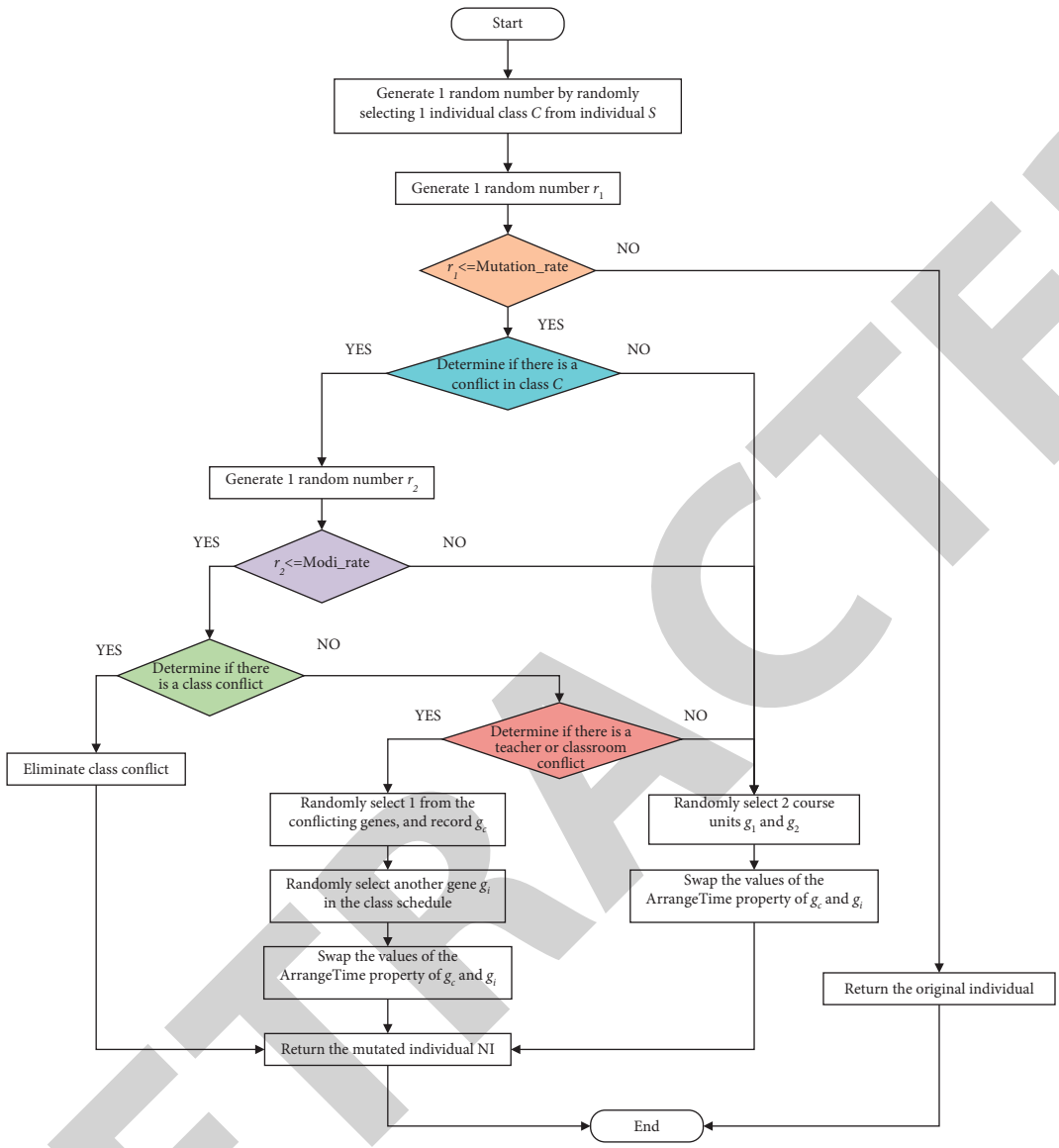

Figure 5: Flowchart of mutation operation.

(2) Construction of chromosomes on the basis of gene coding. The class schedule units (genes) of all classes form 1 chromosome (individual). The CourseInfo object represents the course information of each course in the schedule, including the course, class, instructor, classroom, campus, and weekly class hours. Each CourseInfo corresponds to one or more genes. This is determined by the number of weekly sessions under the CourseInfo. Each individual has the same number of genes.

4.2. Adaptation Function. In this paper, the fitness function is considered in terms of the priority and uniformity of the course sessions, and punishment degree is added to reduce the violation of constraints.

In summary, the fitness function in this paper is given by

$$f = (\omega_1 \cdot f_1 + \omega_2 \cdot f_2) \cdot \beta, \tag{11}$$

where ω_1 and ω_2 are the weights, and $\omega_1 + \omega_2 = 1$. $\beta = \alpha_1^{\text{hard\_}vios} \cdot \alpha_2^{\text{soft\_}vios}$ is punishment degree. α_1 and α_2 are the penalty factor, $\alpha_1 < 1$, $\alpha_2 < 1$, and $\alpha_1 < \alpha_2$.

4.3. Design of Crossover and Mutation Operations in Scheduling System. The crossover strategy in this paper is as follows: individuals Individual_1 and Individual_2 are selected by the roulette algorithm based on individual adaptation. 1 grade is randomly selected, and the set of gene objects is collated. Exchange the gene objects of the classes involved in these 2 individuals.

The idea of gene editing is introduced in the mutation operation in this paper.

Step 1: 1 class is randomly selected and all gene objects related to the class are obtained from the individuals to be mutated.

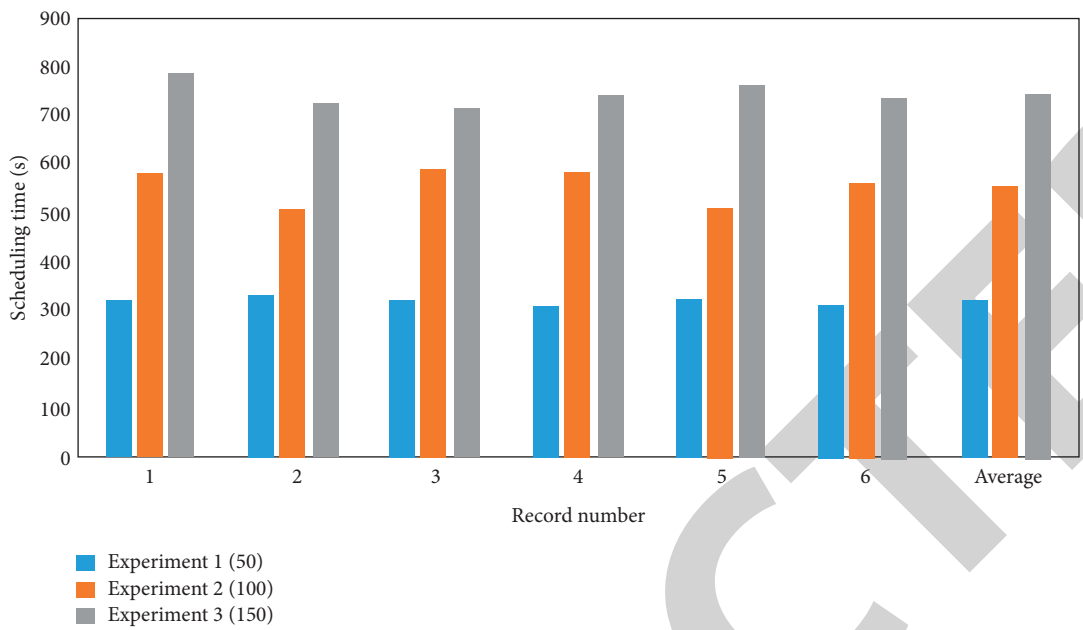

Figure 6: Effect of population size on scheduling time.

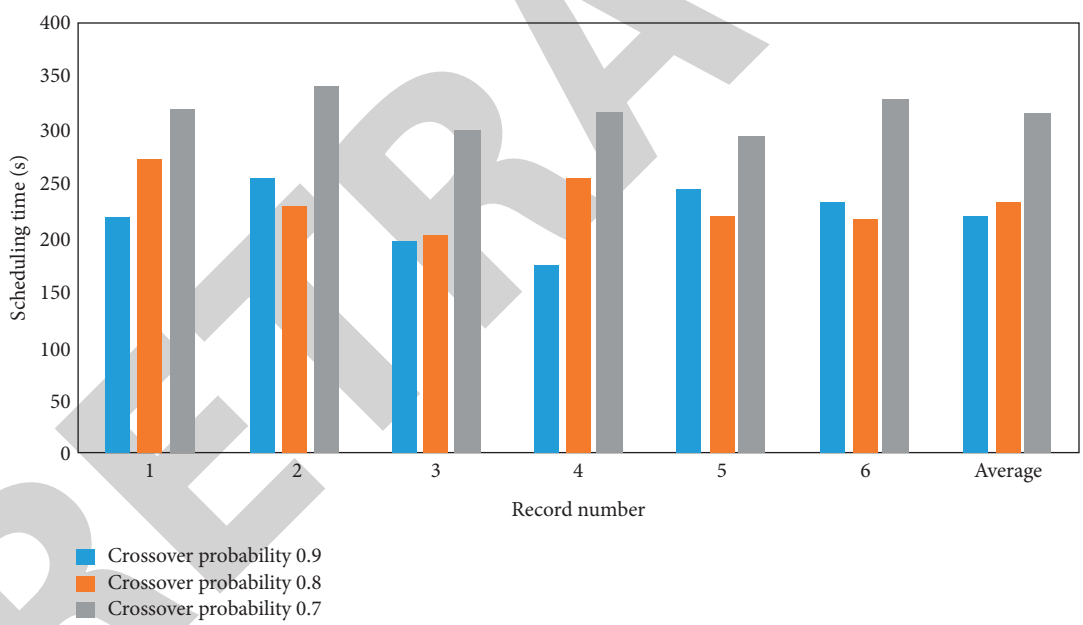

Figure 7: Effect of crossover probability on scheduling time.

Step 2: determine whether the set of gene objects has conflicts. If the constraint is violated, step 3 is executed; otherwise step 5 is executed directly.

Step 3: identify if the course schedule has a conflict (2 or more courses are scheduled for the same class period). If there is, only 1 class is reserved for that period, and the rest of the classes are scheduled for the free period. That is, select $(i-1)$ gene objects from the i conflicting gene objects and set their arrangeTime property value to the unscheduled free period, ignoring whether it will cause other conflicts. If there is no class conflict, then step 4 is executed; otherwise, the current mutation operation is exited.

Step 4: identify if the teachers and classrooms schedule has a conflict. If yes, swap the conflicting course schedule with another schedule. After step 4, save the mutated individuals and exit the mutation operation.

Step 5 : randomly select 2 courses and swap them.

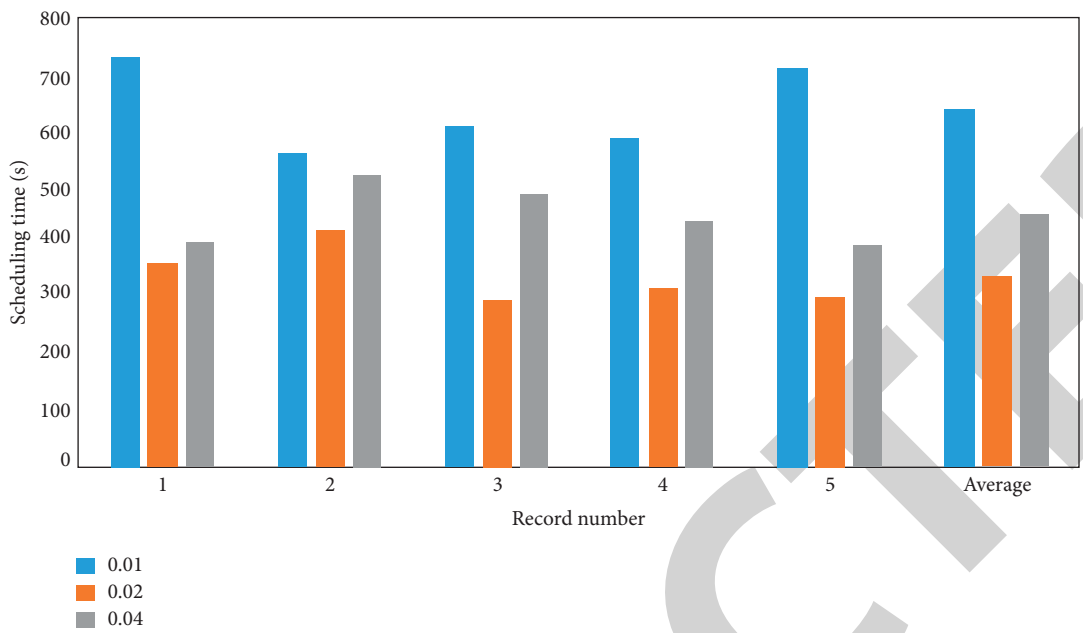

Figure 8: Effect of mutation probability on scheduling.

The overall flowchart of each mutation operation is shown in Figure 5, where Mutation_rate is the mutation probability and Modi_rate is the genetic modification probability.

4.4. Termination Conditions. When the class size is large, the individuals obtained from 1,000 iterations of the population may still have a certain gap with the optimal solution, so the maximum number of genetic iterations should be automatically adjusted according to the class size. The termination conditions of evolution are given:

(1) The population is iterated to $125 \times$ classNum times, where classNum denotes the number of classes in the school.

(2) The best individual in the population is conflict-free, and the ratio of the average fitness of the individuals in the population to the fitness of the best individual is greater than 0.9.

(3) The best individual in the population is maintained continuously for $5 \times$ classNum generations.

5. Result Analysis and Discussion

5.1. Parameter Setting. The university scheduling algorithm based on improved genetic ant colony hybrid optimization was performed in a microcomputer environment with 2 GB RAM, Core 2 Duo2.2 GHz CPU, Windows 7, Visual C++2008. The proposed algorithm was tested using data from the School of Mathematics and Information Systems of a university.

Time slots: Classes are held 5 days a week, with 5 classes scheduled each day, for a total of 25 slots.

Classrooms: 56 (including all scheduled classrooms and computer rooms).

Practical training room: 25.

Teachers (including external teachers): 180.

Classes: 63 (maximum number of students per class is 50).

Students: 3620.

According to the principle of genetic algorithm to search for optimal solutions, the following factors were considered experimentally.

First, the effect of population size on scheduling: the main reference is the time required to find the optimal solution. How large the population size is moderate, so that it is more cost-effective for scheduling. Therefore, it is important to find a moderate population size.

Second, the impact of crossover probability on scheduling: the size of the crossover probability affects the convergence of the search for the optimal solution; the larger the value the faster the convergence, but too large will converge too early for a better global search. Therefore, it is important to find a moderate value of crossover probability.

Third, the impact of variance probability of scheduling: too large variance probability will cause the genetic algorithm to be unstable. Too small variation probability will in turn make the global search difficult. Therefore, it is important to find a moderate value of variation probability.

Fourth is the number of generations of genetic evolution at convergence.

5.2. Experimental Comparison. The effect of population size on scheduling is given in Figure 6. The population size was tentatively set at 50, 100, and 150, and three groups were tested, with six results recorded for each set of experiments.

Table 1: Comparison of metric performance of several algorithms.

Algorithm	Conflict rate (%)	Success rate (%)	Run time (s)	Scalable soft constraint
Proposed algorithms	0	100	36	YES
Greedy algorithm	44	57	94	NO
Genetic algorithm	29	72	164	NO
Ant colony algorithm	18	85	127	NO
Genetic ant colony algorithm	13	91	113	NO
Annealing algorithm	52	49	234	NO

Table 2: Comparison of algorithm adaptability.

Scheduling unit	Genetic algorithm	Ant colony algorithm	Proposed algorithm
66	178.9	197.6	210.4
124	200.6	219.8	227.5
239	229.5	247.5	269.7
528	268.6	283.9	319.6
849	297.4	317.8	353.8

Obviously, in agreement with the common knowledge and prediction of this problem in this paper, the smaller the population size, the shorter the time to find a suitable solution. However, the population size is too small to find the optimal solution within a limited number of genetic generations. From the experiment in Figure 6, it can be seen that population size 100 is a more moderate result.

The effect of crossover probability on scheduling is given in Figure 7. In this paper, experiments were conducted for crossover probabilities of 0.7, 0.8, and 0.9.

From the above experimental data in Figure 7, it can see that the larger the crossover probability is, the less time it takes to find a joint solution, which is actually the faster the search process converges. However, too large will converge too early to perform a better global search. Therefore, in this paper, the crossover probability is taken as 0.8, which is a more moderate result.

The effect of mutation probability on scheduling is given in Figure 8. If the mutation probability is too high, the algorithm will be unstable. According to genetic knowledge, too much variation does not produce good individuals but makes each chromosome unstable and produces a lot of conflicting results. Mutation probability is too small, too few new individuals are generated, and global search is difficult to carry out. In this paper, the probability of variation is 0.01, 0.02, and 0.04. From the experimental results in Figure 8, the difference between the experimental results of different mutation probability can be seen. The mutation probability of this system is 0.02, which is a moderate result.

Based on the original scheduling data, experiments were conducted for different penalty factors parameters. It can be obtained that when the probability of variation is 0.02, the penalty factor α_1 is 0.88, and the penalty factor α_2 is 0.94, and the fitness of the final result is the highest, reaching 0.83729.

5.3. Other Performance Analyses. Experimental simulations were conducted to compare with the greedy algorithm, genetic algorithm, ant colony algorithm, genetic ant colony algorithm, annealing algorithm, and the proposed algorithm to derive the performance differences of each algorithm in terms of conflict rate, success rate, running time, and scalable soft constraints for the metrics. The algorithm in this paper performs 20 simulation experiments with certain course weights, and the average performance values are shown in Table 1. It can be seen that the performance of the scheduling algorithm in this paper is better.

Different scheduling units such as 66, 124, 239, 528, and 849 were set and the adaptation values were compared under the above different scheduling units to obtain the results in Table 2.

The results in Table 2 show that the fitness values of the three algorithms increase simultaneously with the increase in the number of scheduling units. After the number of scheduling units reaches a certain level, the fitness of the three algorithms tends to stabilize. Keeping the number of scheduling units constant, the fitness value of the hybrid algorithm is higher than that of the single algorithm, which indicates that the hybrid algorithm is feasible and effective in the application.

6. Conclusion

The structural design and functional perfection of the scheduling system are increasingly demanding in view of the trend of diversity in university curriculum. Administrators should study the theory and practice of scheduling system design from the practical operational level. More intelligent, effective, and feasible algorithms are continuously explored to solve the problems encountered in the process of system development and actual operation. This paper presents a college scheduling algorithm based on improved genetic ant colony hybrid optimization. The algorithm proposes an enhanced infection genetic algorithm, a directed neighborhood expansion strategy, and a penalty mechanism and innovatively introduces the idea of gene editing in the genetic operator of the genetic algorithm. All these enhance the

scheduling performance. This paper is based on the actual needs of universities to solve the scheduling problem, and the innovation points are mainly in the following four aspects:

(1) The enhanced infection genetic algorithm is proposed, which removes the unnecessary replication operation in the traditional genetic algorithm and improves the convergence speed, convergence accuracy, and operation speed of the algorithm.

(2) The ant movement rules are redefined using a directed neighborhood expansion strategy in the ant colony optimization algorithm to further shorten the path and improve the search efficiency.

(3) The genetic operator in the genetic algorithm innovatively introduces the idea of gene editing to automatically locate a conflict in the class schedule with a certain probability and eliminate the conflict, which greatly improves the evolutionary effect of the genetic algorithm.

(4) For the initial population generation strategy of the genetic algorithm proposed in this paper, a penalty degree design is introduced into the fitness function, which also improves the evolutionary effect of the genetic algorithm to a certain extent.

Experimental results show that the algorithm proposed in this paper outperforms other algorithms in terms of scheduling conflict rate, success rate, running time, and fitness value. Therefore, the general framework and ideas of the improved genetic ant colony hybrid optimization algorithm for solving the scheduling problem proposed lead to an effective solution to the colleges and university scheduling problem. The future work is to improve the college scheduling algorithm to further improve the efficiency of scheduling.

Data Availability

The labeled data set used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by the Hunan University of Information Technology and Hunan First Normal University.

References

[1] W. D. Pahlawanti, E. Harapan, and D. Wardiah, "The influence of school principal supervision and school committee participation on the quality of junior high school education," *International Journal of Progressive Sciences and Technologies*, vol. 23, no. 1, pp. 324–333, 2020.

[2] X. Xing, M. Huerta, and T. Garza, "College and career preparation activities and their influence on post-high school education and work attainment," *Journal of Career and Technical Education*, vol. 34, no. 1, p. 8, 2019.

[3] L. Williams, M. Martinasek, K. Carone, and S. Sanders, "High school students' perceptions of traditional and online health and physical education courses," *Journal of School Health*, vol. 90, no. 3, pp. 234–244, 2020.

[4] H. L. Huang, G. J. Hwang, and C. Y. Chang, "Learning to be a writer: a spherical video-based virtual reality approach to supporting descriptive article writing in high school Chinese courses," *British Journal of Educational Technology*, vol. 51, no. 4, pp. 1386–1405, 2020.

[5] K. P. Kremer, "Predictors of college success outcomes in emerging adults: the role of high school dual enrollment courses," *Emerging Adulthood*, vol. 10, no. 1, pp. 188–196, 2022.

[6] Y. V. Ermanto and Y. F. Riti, "Comparison of welch-powell and recursive largest first algorithm implementation in course scheduling," *Journal of Management Science (JMAS)*, vol. 5, no. 1, pp. 05–12, 2022.

[7] A. K. Nugroho, I. Permadi, and A. R. Yasifa, "Optimizing course scheduling faculty of engineering unsoed using genetic algorithms," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 7, no. 2, pp. 91–98, 2022.

[8] R. Parks, H. Ajjan, A. Gaebel, and A. Taylor, "Room scheduling: a dependent variable to reduce the spread of COVID-19," *College & University*, vol. 96, no. 2, pp. 39–41, 2021.

[9] X. Wang, Y. Chen, and X. Wang, "Design and application of experiment teaching arrangement system," *Technology Wind*, no. 6, pp. 69–71, 2022.

[10] Z. Zaeniah and S. Salman, "Designing class schedule information system by using taboo-search method," *Pilar Nusa Mandiri: Journal of Computing and Information System*, vol. 16, no. 2, pp. 241–248, 2020.

[11] R. Baker, B. Evans, Q. Li, and B. Cung, "Does inducing students to schedule lecture watching in online classes improve their academic performance? An experimental analysis of a time management intervention," *Research in Higher Education*, vol. 60, no. 4, pp. 521–552, 2019.

[12] M. Tomić and D. Urošević, "A heuristic approach in solving the optimal seating chart problem," in *Proceedings of the International Conference on Mathematical Optimization Theory and Operations Research*, pp. 271–283, Springer, Irkutsk, Russia, September 2021.

[13] N. G. A. H. Saptarini, P. I. Ciptayani, and I. B. I. Purnama, "A custom-based crossover technique in genetic algorithm for course scheduling problem," *TEM Journal*, vol. 9, no. 1, pp. 386–392, 2020.

[14] P. Wang, X. Xu, and C. Liu, "An improved adaptive genetic algorithm and its application in intelligent course scheduling system," in *Proceedings of the 2019 6th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 121–125, IEEE, Shanghai, China, December 2019.

[15] X. Chen, X. G. Yue, R. Y. M. Li, A. Zhumadillayeva, and R. Liu, "Design and application of an improved genetic algorithm to a class scheduling system," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 01, p. 44, 2021.

[16] S. A. A. Edalatpanah, "A direct model for triangular neutrosophic linear programming," *International journal of neutrosophic science*, vol. 1, no. 1, pp. 19–28, 2020.

[17] E. Kurtuluş, A. R. Yıldız, S. M. Sait, and S. Bureerat, "A novel hybrid Harris hawks-simulated annealing algorithm and RBF-

based metamodel for design optimization of highway guardrails," *Materials Testing*, vol. 62, no. 3, pp. 251–260, 2020.

[18] E. Olivares Benitez, M. Beatríz Bernábe-Loranca, S. O. Caballero-Morales, and R. G. Macias, "Multi-objective design of balanced sales territories with taboo search: a practical case," *International Journal of Supply and Operations Management*, vol. 8, no. 2, pp. 176–193, 2021.

[19] H. Rashidi and M. Hassanpour, "A deep-belief network approach for course scheduling," *Journal of Applied Research on Industrial Engineering*, vol. 7, no. 3, pp. 221–237, 2020.

[20] D. Shi, W. Fan, Y. Xiao, T. Lin, and C. Xing, "Intelligent scheduling of discrete automated production line via deep reinforcement learning," *International Journal of Production Research*, vol. 58, no. 11, pp. 3362–3380, 2020.

[21] M. A. M. Shaheen, H. M. Hasanien, and A. Alkuhayli, "A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution," *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 621–630, 2021.

[22] J. Kazemi Kordestani, M. R. Meybodi, and A. M. Rahmani, "A note on the exclusion operator in multi-swarm PSO algorithms for dynamic environments," *Connection Science*, vol. 32, no. 3, pp. 239–263, 2020.