Journal of Engineering, Project, and Production Management
2026, 16(5), 2026-21

An Intelligent Computer Course Scheduling Method Using IGA-ISA

Baimei Xu
Dean, College of Continuing Education, Jiangsu Vocational College of Finance and Economics, Huai'an, 223003, China, Email: Xubai18@outlook.com

Project Management
Received January 9, 2026; revised March 17, 2026; accepted July 16, 2026
Available online July 25, 2026

Abstract: To solve the problems of many resource conflicts and low efficiency of traditional algorithms in course scheduling in colleges and universities, this study proposes an intelligent course scheduling method based on the Improved Genetic Algorithm-Improved Simulated Annealing Algorithm (IGA-ISA). First, the hard and soft constraints of course scheduling are analyzed, and a mathematical model of course scheduling is constructed based on the weight of course periods, the uniformity of class distribution, and classroom utilization indicators. The Genetic Algorithm (GA) is optimized using decimal coding and adaptive crossover and mutation, and Cauchy distribution perturbation and exponential temperature attenuation are introduced to improve the simulated annealing algorithm, forming IGA-ISA. Experiments show that in the Oliver30 and Barma14 calculation examples, the convergence times of this algorithm are 24 and 17 times, respectively, which are lower than those of other methods. At the same time, its deviation rate is only 0.64%. In the actual application test, the average running time of this algorithm is 175.6 s, which is lower than that of the comparison methods. Its average conflict rate is 2.5%, and its average classroom utilization rate is 96.4%, both of which are better than those of other methods. The above results show that the constructed IGA-ISA can improve the quality of course scheduling and provide technical solutions for intelligent course scheduling in colleges and universities.

Keywords: Course scheduling; genetic algorithm; simulated annealing algorithm; adaptive crossover mutation; cauchy distribution.

DOI 10.32738/JEPPM-2026-21

---

1. Introduction

Course Scheduling in Colleges and Universities (CUCS) is a core part of academic management, and its complexity has become increasingly evident with the expansion of educational scale and the diversification of teaching needs. Currently, CUCS needs to coordinate multi-campus, multi-college, and multi-professional teaching resources and balance the five core elements of teachers, classes, courses, classrooms, and time. This must not only meet hard constraints, such as teachers/classrooms not being reused during the same period, but also take into account soft constraints, such as prioritizing professional courses during periods of high learning efficiency and avoiding consecutive classes (Thepphakorn et al., 2023; Zhuniskhanov et al., 2024). However, the traditional manual class scheduling model is inefficient and prone to problems such as resource conflicts and uneven distribution of class hours. Existing intelligent course scheduling algorithms still face bottlenecks such as slow convergence speed, high conflict rate, and insufficient classroom utilization in complex scenarios, and are difficult to adapt to the dual needs of colleges and universities for accuracy in course scheduling and rational resource allocation (Alaka et al.,2025; Hafsa et al., 2025). At the same time, the personalized demands of teachers and students for teaching experience continue to increase, such as teachers' preferences for teaching at specific times and students' demands for reasonable course intervals, which further increases the difficulty of class scheduling. In this context, there is an urgent need to develop an intelligent course scheduling model that accounts for hard and soft constraints and integrates efficient optimization strategies. The purpose is to break through the limitations of existing technology, improve the efficiency of course scheduling and the rationality of plans, and provide key technical support for the construction of intelligent educational management systems in universities.

Ardana et al. (2024) proposed an adaptive lecture scheduling system based on a Genetic Algorithm (GA) to address the problem of universities relying on manual or static course scheduling and the difficulty of coping with resource changes and multiple constraints. This method uniformly coded dynamic data such as lecturers, classrooms, time, courses, and preferences, and sought the optimal conflict-free course schedule in demand analysis, system design, algorithm implementation, and multi-scenario simulation iterations. Compared with traditional methods, this system significantly shortened class scheduling time, eliminated conflicts, and supported real-time adjustments as courses increase or decrease, thereby effectively improving the quality of teaching services and providing a replicable intelligent class scheduling solution for similar institutions. Ravindra (2025) proposed a nonlinear optimization framework based on the differential evolution algorithm to address multi-constraint coupling and solution quality in educational timetable arrangement. This method quantified time periods, teacher expertise, and course priorities via feature engineering and dynamically evaluated candidate course schedules under both hard and soft constraints, enabling multi-configuration parallel exploration. This model significantly reduced conflicts, compressed idle periods, and improved classroom utilization, providing colleges and universities with a new way to structure and scale course scheduling to balance institutional needs and logistical constraints. Davison et al. (2025) addressed the new challenges of course scheduling under the hybrid teaching model in universities, proposed a multi-objective binary programming model that integrates the characteristics of hybrid teaching, and designed a dictionary solution method. This method unified the modeling of traditional course scheduling constraints and online/offline hybrid teaching requirements, and revealed the trade-off relationships among the goals through benchmark case experiments. This model could not only generate a high-quality curriculum that meets the needs of hybrid teaching but also provide a quantitative basis for decision-making to help universities formulate teaching strategies.

GA is a global optimization search algorithm that simulates natural selection and genetic mechanisms. It does not rely on gradients and has low requirements on the form of the objective function. Therefore, it is suitable for complex optimization scenarios with high dimensionality, non-convexity, multimodality, discrete or mixed variables. Sha (2024) proposed an optimization algorithm based on GA and Long Short-Term Memory (LSTM) to address the insufficient accuracy of traditional time-series models for stock price prediction. This method first used GA to globally optimize LSTM hyperparameters, and then performed training and fine-tuning on large-scale data samples. The model's average error dropped from 0.11 to 0.01 and stabilized, and the prediction curve was highly consistent with the actual price. Smallholder agriculture in West Africa suffers from limited access to seeds, fertilizers, agricultural machinery, and agricultural climate services, resulting in inaccurate crop model yield estimates and low reliability of decision support. Waongo et al. (2025) proposed a point calibration method that couples GA with the AquaCrop crop model. This method used GA to automatically search for optimal variety parameters and management combinations at the grid-unit scale. AquaCrop was responsible for simulating corn yield and achieving model localization. This method could distinguish management characteristics, such as fertility rates and weed cover, in the Sahel-Sudania region, providing a new tool for regionally differentiated planting strategies and climate adaptation. Akopov (2024) proposed an improved parallel dual-objective mixed-real-coding GA to address the insufficient quality of the Pareto front approximation in multi-objective optimization. This method combined a parallel real-encoding GA with multi-objective Particle Swarm Optimization (PSO), used K-means, hierarchical clustering, and c-means to construct clusters, and then applied mutations to the cluster centers to improve diversity. This method significantly outperformed existing advanced algorithms on standard test sets and consistently produced high-quality Pareto fronts in agent commodity exchange models across different initial agent distributions.

In summary, current research on intelligent class scheduling is relatively extensive, but existing algorithms still face issues such as high conflict rates and insufficient classroom utilization. In view of this, this study proposes an intelligent course scheduling method that integrates the Improved Genetic Algorithm (IGA) and Simulated Annealing (SA) to solve the CUCS problem. The innovation of the research lies in the design of an adaptive crossover mutation strategy to reduce the damage to the optimal solution caused by random perturbations of traditional GA. Meanwhile, the Cauchy distribution perturbation is introduced to enhance SA's global search capability, and the exponential temperature decay and re-warming mechanisms are combined to optimize convergence efficiency, forming a synergistic mechanism of "local precise search + global efficient optimization". Scheduling is a typical NP-hard problem, and existing research on scheduling is mostly a simple serial mixture of GA and SA, which has not resolved the trade-off between global search and local optimization. The IGA-ISA used here is not a shallow fusion of algorithms but rather starts from the characteristics of the solution space of NP-hard scheduling problems. The adaptive cross-mutation IGA serves as the framework for traversing the global solution space. The Improved Simulated Annealing (ISA) is embedded in the mutation offspring optimization stage to construct a hierarchical solution mechanism of "global evolutionary screening-local precise optimization". This breaks through the bottleneck of traditional hybrid algorithms, which are prone to getting stuck in local optima, and introduces a new approach to hierarchical collaborative algorithm design for NP-hard scheduling problems, enriching the theoretical and application paradigms of combinatorial optimization in the scheduling field.

2. Methods

To improve the efficiency of class scheduling, rationally allocate teacher resources, and improve classroom utilization, this study proposes an intelligent computer class scheduling method based on IGA. This study first analyzes the soft and hard constraints of the course scheduling problem and then builds a mathematical model of course scheduling. Secondly, adaptive crossover and mutation are used to improve GA, and SA is introduced to further optimize GA.

2.1. Analysis of Computer Course Scheduling Needs and Construction of Mathematical Models

Since CUCS issues are very complex and involve multiple colleges, campuses, majors, classes, and teachers. Therefore, resources need to be properly planned to ensure courses can be delivered smoothly across different teaching venues. In addition to hard constraints such as classroom conflicts and time conflicts, intelligent class scheduling also needs to meet soft constraints such as optimal course times and course frequency. As for the course scheduling algorithm, its users are divided into two categories, namely academic administrators and ordinary users. The user use case diagram of the intelligent class scheduling model is shown in Fig. 1.

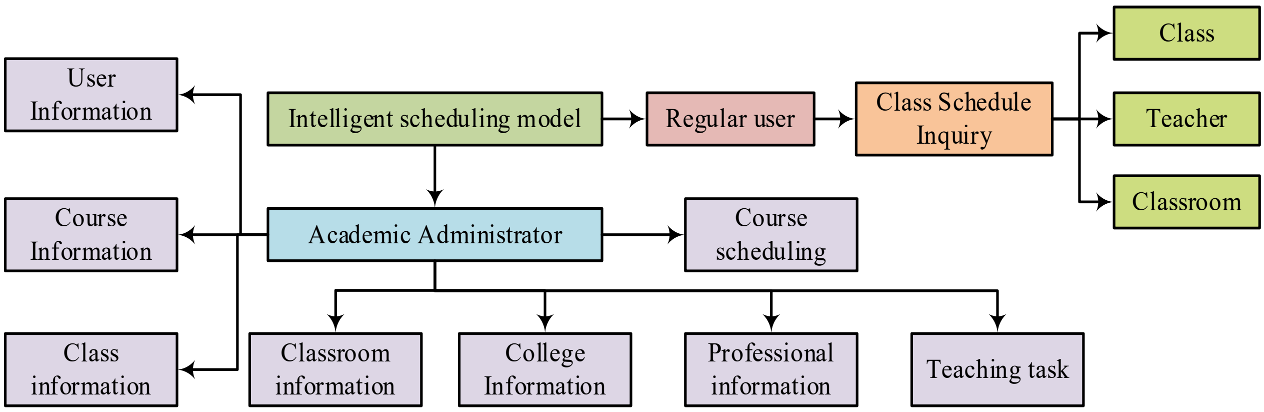

Fig. 1. User use case diagram of intelligent course scheduling model

In Fig. 1, the academic administrator arranges resources according to the course selection situation and plan, uses IGA to automatically generate reasonable course scheduling plans, and manages teaching information. In special circumstances, courses can be manually adjusted. Ordinary users are divided into teachers and students. They can only log in to the system to query class schedules. Therefore, when building an intelligent class scheduling model, it is necessary to focus on the needs of academic administrators. The class scheduling problem involves five major factors: teacher, class, course, classroom, and time. Each factor has a unique number. One day is divided into 5 class periods, each period has two classes, for a total of 25 periods (Musatova et al., 2023; Zcan et al., 2025). In addition to the above-mentioned main influencing factors, there are other secondary factors, such as odd- and biweekly class schedules, classroom type, classroom capacity, optimal learning period, and the weight of each section. For the class scheduling model, if no constraints are imposed, it will lead to problems such as teacher time conflicts and insufficient classroom capacity. Therefore, to ensure the rationality of course scheduling, it is necessary to set soft/hard constraints on the model. Among them, hard constraints are rules that must be followed, and soft constraints are goals for further optimization after the hard constraints are satisfied. The hard constraints are as shown in Eq. (1).

$$\begin{cases} \sum_{b=1}^{B}\sum_{k=1}^{K}\sum_{s=1}^{S} e_j c_b r_s t_i s_k \le 1 \\ \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{s=1}^{S} e_j c_b r_s t_i s_k \le 1 \\ \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{b=1}^{B} e_j c_b r_s t_i s_k \le 1 \\ c_v \le r_v \\ \text{roomType} = \text{courseType} \end{cases} \tag{1}$$

In Eq. (1), B is the number of classes. K is the number of courses. D is the number of classrooms. e_j stands for teacher. c_b is class. r_s is the classroom. t_i is the class time. s_k is the course. J is the number of teachers. c_v is the class size. r_v is the number of people the classroom can accommodate. The above constraints mean that a teacher can teach only one class in a single classroom at a time. Settings of decision variables $x_{e,c,r,t,s}$: when teacher e arranges courses s for class c in classroom r and time t, 1 is taken; otherwise, 0 is taken. Settings of hard constraint. (1) $\sum_{c,s} x_{e,c,r,t,s} \le 1$, unique for teachers/classrooms in the same time and space. (2) $\sum_{e,r,t} x_{e,c,r,t,s} = n_s$, completing the prescribed class hours n_s for course s. (3) $c < r$ class size matches classroom capacity. A normalized fitness function $F = \omega_1 f_1 + \omega_2 f_2 + \omega_3 f_3 - \omega_4 \frac{\sum P_i}{max\, P_i}$ is constructed. f_1 represents the degree of optimization for the time period. f_2 is the uniformity of class hour distribution. f_3 is the classroom utilization rate, both normalized to [0,1]. P_i is the penalty for violating hard constraints. ω_i is the weight, and the sum of each weight is 1. Soft constraints include: (1) Compulsory courses and professional courses should be arranged in the first period in the morning or afternoon as much as possible. (2) Try to consider teachers' wishes when scheduling classes. (3) Evenly distribute classes and courses to avoid consecutive classes. (4) Classroom capacity matches the number of students. In addition, since students and teachers have different learning efficiency and teaching willingness at different time periods, it is also necessary to investigate the weight of each time period to optimize class scheduling. The time-period weights are obtained through a standardized questionnaire survey. The survey targets 20 full-time teachers, 800 students, and 10 academic administrators from various finance and economics majors at the school, covering diverse groups in teaching habits, learning needs, and scheduling experience. The questionnaire employs a 5-point Likert scale to rate the teaching/learning efficiency of each time period. After reliability and validity testing, the Cronbach's alpha coefficient is 0.82, and the KMO value is 0.79, indicating that the data meet statistical reliability and validity requirements. The scores from valid questionnaires are normalized and weigh 0.4:0.4:0.2 for teachers, students, and management personnel. Finally, the teaching efficiency weight values for each time period are obtained, providing objective data support for optimizing course time periods. The weights of the course periods are shown in Table 1.

Table 1. Course period weights

Time slot	Monday	Tuesday	Wednesday	Thursday	Friday
8:00-10:00	0.98	0.98	0.96	0.95	0.92
10:00-12:00	0.94	0.90	0.88	0.85	0.92
14:00-16:00	0.75	0.81	0.78	0.72	0.56
16:00-18:00	0.68	0.66	0.66	0.66	0.56
17:00-21:00	0.60	0.52	0.57	0.54	0.48

In Table 1, the weights of each time period in the morning are higher than those in the afternoon and evening. Therefore, considering the teaching effect, professional courses and required courses should be arranged in the morning. Secondly, both teachers and students are less willing to attend classes in the evening, so classes should be scheduled as few times as possible. At this time, the degree of optimization of the course time period is given by Eq. (2).

$$o = \frac{1}{n}\sum_{i=1}^{n} \omega_j w_i \tag{2}$$

In Eq. (2), o is the degree of optimization of the time period of the course. n is the number of courses. ω_j is the weight of the course. w_i is the time period weight. In addition, to avoid over-concentration of courses, it is necessary to calculate the average distribution of daily class hours. The calculation formula is shown in Eq. (3).

$$b_i = \frac{1}{\sqrt{\sum_{d=1}^{m} (c_d - c_{avg})^2}} \tag{3}$$

In Eq. (3), b_i is the distribution mean of daily class hours. m is the number of class days per week. c_d is the number of lessons for the class on d instructional day. c_{avg} is the average number of daily lessons in the class. At this time, the average daily class time distribution for the whole school is given by Eq. (4).

$$F = \frac{\sqrt{\sum_{i=1}^{L} b_i}}{L} \tag{4}$$

In Eq. (4), F is the distribution mean of the school's average daily class hours. L is the number of classes in the school. Finally, to improve classroom utilization, it is necessary to calculate it. The classroom utilization degree is calculated as Eq. (5).

$$U = \frac{1}{C}\sum_{i=1}^{C}\sum_{j=1}^{R} \frac{M(i,j)}{capacityr_j} \tag{5}$$

In Eq. (5), U is the classroom utilization level. C is the number of courses in all classes. R is the number of classrooms in the school. $M(i,j)$ is the class size in the classroom r_j. Using the above method, a mathematical model for intelligent computer course scheduling is constructed.

2.2. A Method for Solving Computer Course Scheduling Problems Based on IGA

After the mathematical model of intelligent computer course scheduling is constructed, it can be solved to achieve intelligent course arrangement. As a search heuristic algorithm that simulates the mechanisms of natural selection and genetics, GA can continuously improve the solution to the problem by simulating inheritance, mutation, selection, and hybridization in the natural evolution process. However, traditional GA has shortcomings, including inaccurate encoding, a tendency to fall into local optima, and low efficiency in solving search problems (Balci et al., 2023; He, 2024a). Therefore, to solve the above problems, this study improved it. First of all, given the characteristics of the course scheduling problem and the difficulty of coding, the selected coding method is decimal coding. The selection strategy for IGA is the roulette method, and the probability of an individual being selected is given by Eq. (6).

$$p_i = \frac{f_i}{\sum_{i=1}^{N} f_i} \tag{6}$$

In Eq. (6), p_i is the probability that the i-th individual is selected. f_i is the individual fitness value. N is the number of individuals in the population. For GA, mutation increases population diversity by introducing new genetic variations, while crossover accelerates convergence by combining superior features (Ahmed et al., 2025; Xu et al., 2025). The crossover and mutation operations are shown in Fig. 2.

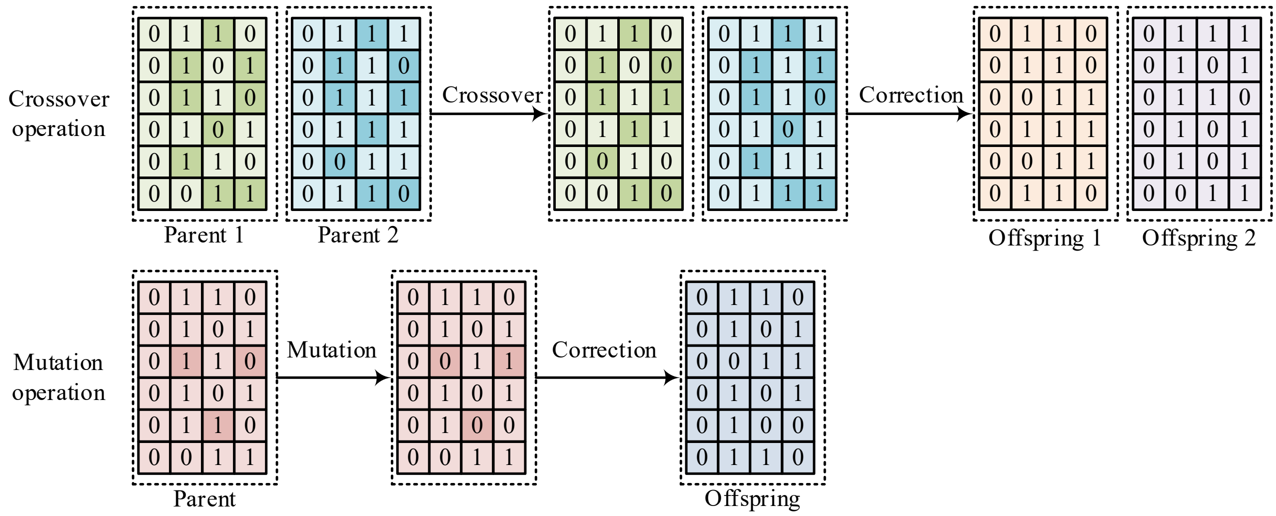

Fig. 2. Cross and mutation operations

In Fig. 2, the crossover operation selects two individuals from the population, randomly selects an intersection point to exchange genes, and replaces the original population individuals with the offspring. The mutation operation randomly selects individuals, then flips bits in the binary encoding or adds a random perturbation to the real encoding. Since the traditional crossover mutation operator is overly random, it is prone to destroying the optimal solution (Rizqy et al., 2025). Therefore, to solve the above problems, this study designs an adaptive crossover and mutation operation. The adaptive crossover probability is given by Eq. (7).

$$p_c = \begin{cases} \dfrac{f_{\max} - f_c}{f_{\max} - f_{avg}} - 1, f_c \geq f_{avg} \\ \gamma, f_c < f_{avg} \end{cases} \tag{7}$$

In Eq. (7), p_c is the individual crossover probability. f_{max} is the fitness of the optimal individual. f_c is the fitness of the crossover individual. f_{avg} is the average fitness. γ is a random number. The adaptive mutation operation is given by Eq. (8).

$$p_m = \begin{cases} \dfrac{f_{\max} - f_m}{f_{\max} - f_{avg}} - 1, f_m \geq f_{avg} \\ \varphi, f_m < f_{avg} \end{cases} \tag{8}$$

In Eq. (8), p_m is the mutation probability of the individual. f_m is the fitness of the mutation. ϕ is a random number. To retain feasible solutions, for randomly selected course time periods with genetic variation, the occupancy status of teachers/classrooms in the target time period is first retrieved, and time-period replacement is performed only if no conflict occurs. If a conflict exists, 3 adjacent time periods are searched iteratively. If there is still a conflict, the mutation is discarded. The post-processing repair method involves traversing the scheduling results and prioritizing courses that violate hard constraints to match the three-dimensional idle set of teacher classroom time slots. If there is no match, the non-core course hours will be adjusted from low to high based on the weight of the penalty items. The time complexity of the operator is $O(k \cdot m)$ (k is the number of mutated genes. m is the number of single course retrieval periods). The complexity of the repair method is $O(n \cdot l)$ (n is the total number of courses; l is the number of times a single course matches the idle set. Both are linear in complexity and can be adapted to scheduling scenarios with thousands of courses. At this time, the IGA process is shown in Fig. 3.

In Fig. 3, IGA first sorts and compresses the solution space by course and teacher weights, uses decimal encoding of chromosomes, and calculates fitness. If the optimal solution is found, the process will terminate. Otherwise, new individuals will be generated through selection, crossover, and mutation, the number of iterations will be increased, and the initial population will be regenerated.

Although IGA can solve the course scheduling problem, it still suffers from weak global search capability and premature convergence. Therefore, to address these issues, this study introduces the SA algorithm to improve it. As a probabilistic heuristic search algorithm, SA can effectively solve complex optimization problems by simulating the physical annealing process (He et al., 2024b; Poudel et al., 2024). The core formula for SA is given by Eq. (9).

$$P(d) = \{1, d < 0 \,\&\, exp^{\;-d/T}, d > 0 \tag{9}$$

In Eq. (9), $P(d)$ is the probability that the new solution is accepted. d is the difference between the previous solution and the new solution. T is the annealing temperature. However, traditional SA suffers from slow convergence and a tendency to fall into local optima (Yan et al., 2023). Therefore, this study uses the Cauchy distribution and exponential temperature decay to construct an ISA algorithm. The process is shown in Fig. 4.

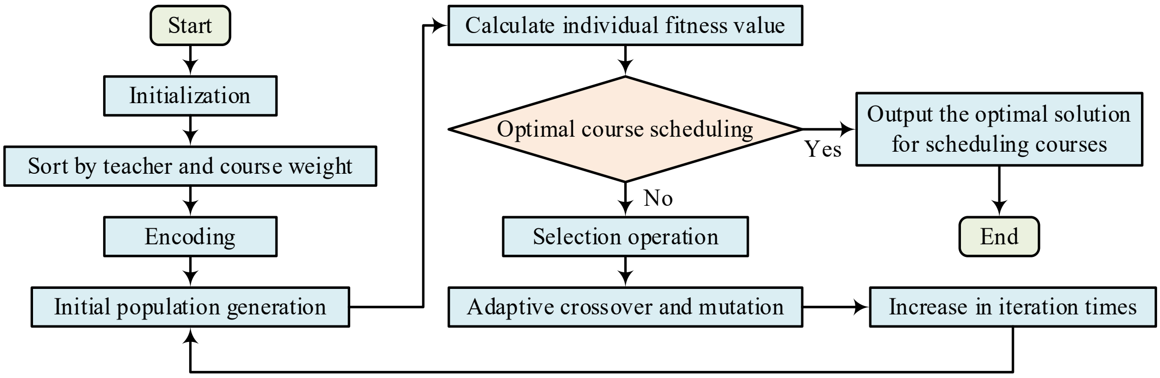

Fig. 3. IGA process

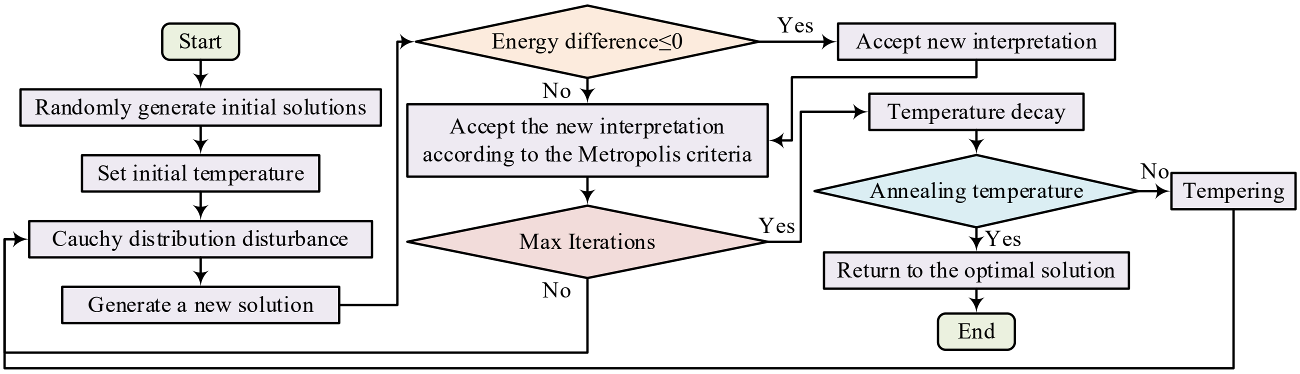

Fig. 4. Process of ISA algorithm

In Fig. 4, ISA generates a new solution by introducing a Cauchy distribution to perturb the current solution, then calculating the energy difference between the new solution and the current one. If the generated new solution is better or meets certain conditions, it will be accepted. The original solution will be returned. Then, the temperature gradually decreases until the termination condition is met, at which point the optimal solution can be output. The probability density function of the Cauchy distribution is shown in Eq. (10).

$$f(t;\lambda) = \lambda e^{-\lambda t} \tag{10}$$

In Eq. (10), $f(t,\lambda)$ is the probability density of event occurrence at time t. λ is the rate parameter. The new solution, after introducing the Cauchy distribution, is given by Eq. (11).

$$\begin{cases} s_{new} = s_{curr} + a(B_U - B_L) \\ a = T_{curr} \cdot sgn(r - 0.5) \cdot [(1 + 1/T_{curr})^{(|2r-1|)} - 1]] \end{cases} \tag{11}$$

In Eq. (11), s_{new} is a new solution. s_{curr} is the current solution. a is the amount of disturbance. B_U and B_L are the upper and lower bounds of the variable. T_{curr} is the current temperature. r is a random number. To accelerate convergence, this study introduces an exponential temperature decay. At this time, the temperature is calculated as Eq. (12).

$$T_{new} = T_{curr} * \mu^n \tag{12}$$

In Eq. (12), T_{new} is the new temperature. μ is the annealing coefficient. n is the number of iterations. Finally, this study also introduces a re-warming operation to avoid the algorithm falling into local optimality, as shown in Eq. (13).

$$T_{new} = T_{curr} * (1 + n/100) \tag{13}$$

ISA is constructed through the above method, and introducing ISA into the mutation stage can effectively enhance the global search capability of IGA. The specific way to embed ISA in the IGA loop is to perform ISA optimization only on the offspring individuals generated during the IGA mutation stage. After completing the IGA selection and crossover operations, the feasibility of the offspring population obtained through mutation is first verified. Infeasible solutions and low-fitness feasible solutions are selected to form a set for optimization, and then ISA is applied to this set. ISA generates new solutions through Cauchy distribution perturbation, accepts the solutions based on the Metropolis criterion, completes one perturbation optimization, and then represents the results back to the IGA offspring population. The high-quality offspring that did not participate in ISA optimization are merged as the new population for the next round of IGA iteration. This embedding method avoids excessive disturbance to elite individuals, prevents the optimal solution from being destroyed, and performs local, precise optimization for the shortcomings of mutated offspring, achieving efficient collaboration between IGA global evolution and ISA local optimization. The ISA-IGA process is shown in Fig. 5.

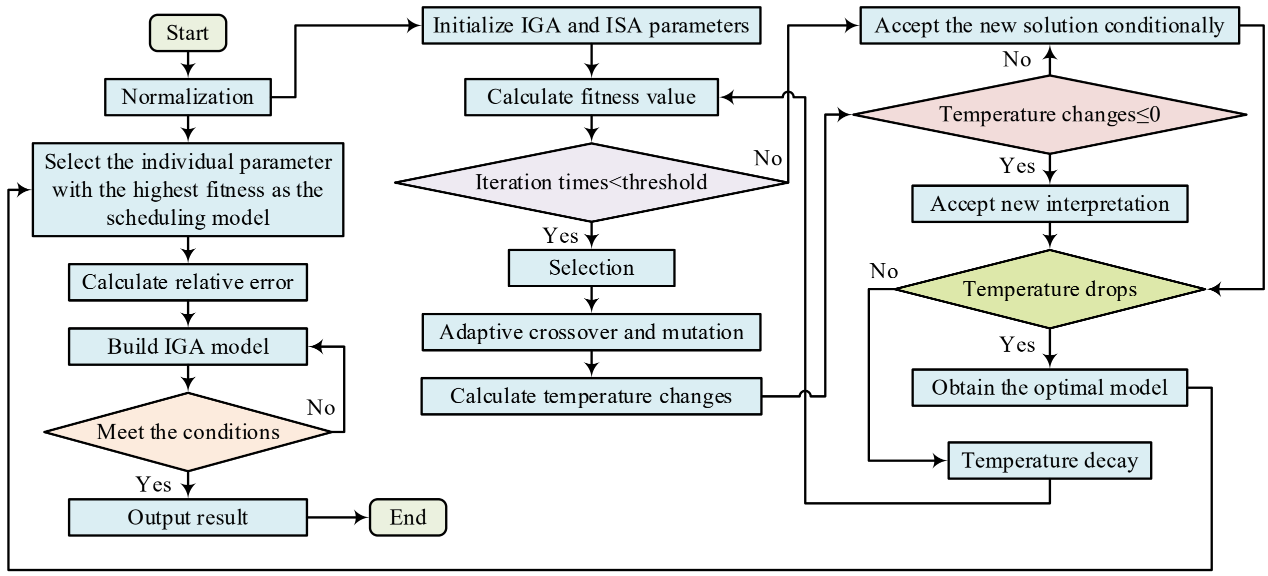

Fig. 5. Process of ISA-IGA

In Fig. 5, the genetic population parameters and SA parameters are first initialized, and the fitness function is calculated. If the maximum number of iterations is not reached, selection, crossover, and mutation operations will be performed to calculate the fitness of the new solution. If the new solution is better or meets the conditions, it will be accepted. It will be decided according to the Metropolis criterion. The next step is to lower the annealing temperature and repeat the above operation until the termination condition is met, at which point the optimal solution can be output.

3. Results

3.1. ISA-IGA Performance Test Results

To evaluate the performance of ISA-IGA, this study uses the classic traveling salesman problem as a benchmark to compare it with Reinforcement Learning-IGA (RL-IGA) and Combination Strategy-IGA (CS-IGA). The experimental calculation examples are Oliver30 and Barma14. The iteration curves of different methods are shown in Fig. 6.

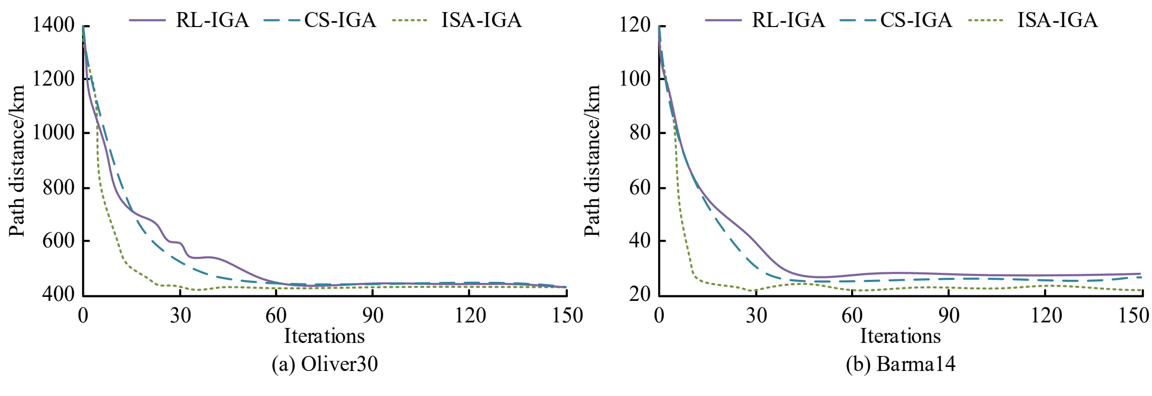

Fig. 6. Iteration curves of different methods

In Fig. 6(a), in the Oliver30 calculation example, RL-IGA and CS-IGA converge after 52 and 43 iterations, while ISA-IGA converges after 24 iterations. The path-based differences between the three are small. In Fig. 6(b), in the Barma14 calculation example, RL-IGA and CS-IGA converge after 43 and 35 iterations. ISA-IGA converges after only 17 iterations, and its path length is significantly smaller than that of other methods. This shows that ISA-IGA has better convergence performance. Table 2 shows the solution results of different methods.

In Table 2, compared with other algorithms, the optimal solutions obtained by ISA-IGA for the Oliver30 and Barma14 examples are closer to the known optimal values of 423.76 and 31.05. The deviation rates of ISA-IGA for the Oliver30 and Barma14 calculation examples are 1.85% and 0.64%, and the average errors are 0.03% and 0.11%, which are lower than those of other methods. This shows that ISA-IGA has a strong problem-solving ability. To verify the algorithm's effectiveness in standard scheduling scenarios, experiments are conducted using the recognized ITC2007 scheduling benchmark set. It contains multiple instances, such as comp01-comp13, that differ significantly in size. The search space ranges from 10^{36} to 101171, covering configurations such as 24-99 teachers, 5-20 classrooms, and 30-131 courses. The datasets UTP-01 and UTP-02 are selected, and the hard violation count, soft penalty value, and objective cost are used as standard indicators for testing. The results are shown in Table 3.

Table 2. Results of solutions using different methods

Item	Example	RL-IGA	CS-IGA	ISA-IGA
Known optimal solution	Oliver30	423.74	423.74	423.74
	Barma14	30.89	30.89	30.89
Optimal solution	Oliver30	431.56	428.74	423.76
	Barma14	33.71	32.48	31.05
Worst solution	Oliver30	467.75	458.92	436.74
	Barma14	38.92	36.51	33.73
Average solution	Oliver30	434.15	432.81	426.44
	Barma14	34.51	33.13	31.42
Deviation rate/%	Oliver30	3.43	2.86	1.85
	Barma14	2.77	1.98	0.64
Mean value error/%	Oliver30	0.96	0.71	0.03
	Barma14	0.88	0.69	0.11

Table 3. Course scheduling benchmark test results

Benchmark dataset	Algorithm	Hard violation count	Soft penalty value	Objective cost
UTP-01	RL-IGA	3.2	89.5	102.7
	CS-IGA	2.5	76.3	88.8
	ISA-IGA	0.0	42.1	42.1
UTP-02	RL-IGA	4.1	95.2	113.5
	CS-IGA	3.0	81.7	94.7
	ISA-IGA	0.0	45.8	45.8

According to Table 1, compared to other algorithms, ISA-IGA achieves a hard violation count of 0 on the ITC2007 benchmark set and reduces the soft penalty value and objective cost by more than 40%. This result verifies the algorithm's excellent performance on the standard scheduling benchmark.

3.2. Intelligent Class Scheduling Experimental Results

To verify the application performance of ISA-IGA in intelligent class scheduling problems, this study tests it and compares it with the Chaos Algorithm - Hybrid Artificial Bee Colony (CA-HABC) and Bipartite Graph-Ant Colony Optimization (BG-ACO). In the experiment, the number of students and teachers is 2,500 and 110, the number of classrooms is 80, the number of classes is 62, and the number of courses is 104. The initial population number of ISA-IGA is 160, the selection probability is 0.3, and the maximum number of iterations is 1000. The changes in running time and maximum fitness value for different methods are shown in Fig. 7.

In Fig. 7(a), the maximum running time of CA-HABC and BG-ACO is 218.7s and 211.5s, and the average running time is 211.7s and 202.1s. The maximum running time of ISA-IGA is 179.8s, and the average running time is 175.6s. Both are much lower than those of other methods. In Fig. 7(b), compared with CA-HABC and BG-ACO, the fitness value of ISA-IGA is larger, approaching 1600, and it tends to converge after only 45 iterations. This shows that ISA-IGA can quickly and accurately solve the course scheduling problem. The classroom utilization rate and class distribution balance are shown in Fig. 8.

In Fig. 8(a), the highest classroom utilization rates of CA-HABC and BG-ACO are 88.8% and 92.5%, and the average classroom utilization rates are 86.0% and 89.7%. The highest classroom utilization rate for ISA-IGA is 98.8%, with an average of 96.4%. In Fig. 8(b), the highest class hour distribution balance degrees of CA-HABC and BG-ACO are 0.78 and 0.83, and the average class hour distribution balance degrees are 0.75 and 0.79. The highest class distribution balance degree of ISA-IGA is 0.93, and the average class distribution balance degree is 0.89. This shows that ISA-IGA can effectively improve classroom utilization and the balanced distribution of classes. The conflict rates of course scheduling and the number of missed courses for different methods are shown in Fig. 9.

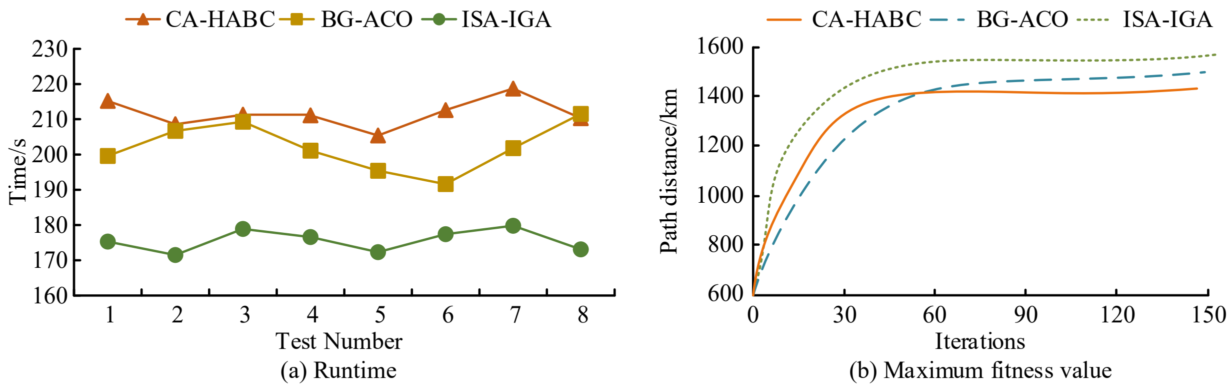

Fig. 7. Changes in running time and maximum fitness value of different methods

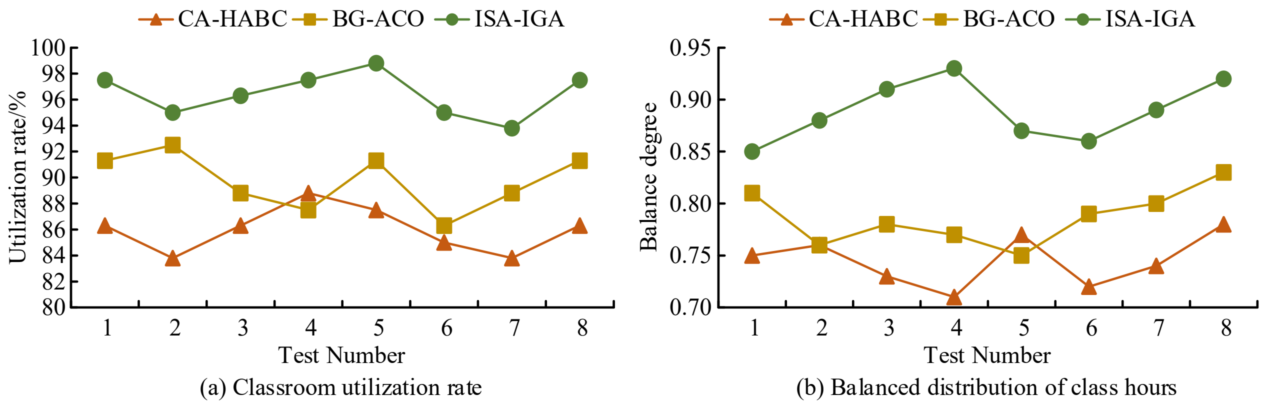

Fig. 8. Classroom utilization rate and balanced distribution of class hours

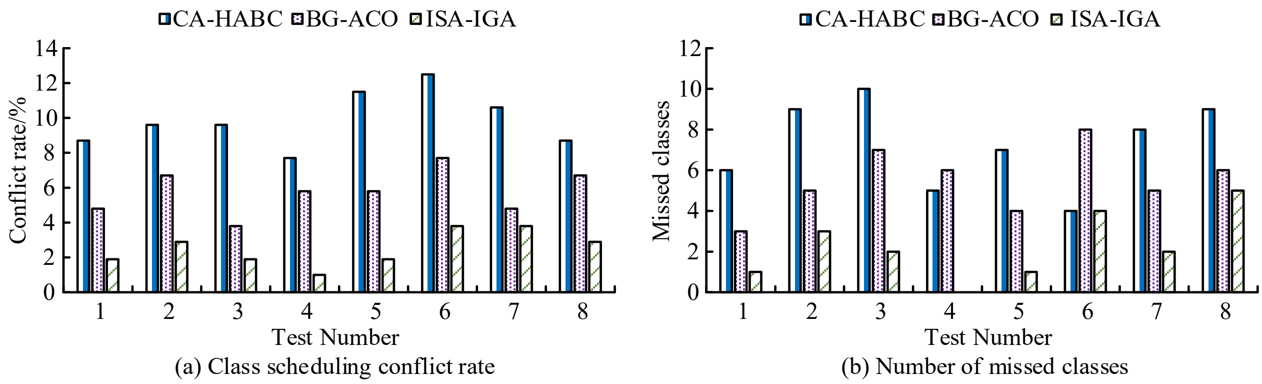

Fig. 9. Conflict rate and number of missed classes for different scheduling methods

In Fig. 9(a), the highest scheduling conflict rates of CA-HABC and BG-ACO are 12.5% and 7.7%, and the average conflict rates are 9.9% and 5.8%. ISA-IGA has a maximum class scheduling conflict rate of 3.8% and an average conflict rate of 2.5%. In Fig. 9(b), the maximum number of missed classes for CA-HABC and BG-ACO is 9 and 8, and the average number of missed classes is 7 and 6. The maximum number of missed classes for ISA-IGA is 5, and the average is only 2. ISA-IGA can effectively reduce the conflict and missed-scheduling rates. To further analyze the performance of ISA-IGA, this study conducts ablation experiments, and the results are shown in Table 4.

In Table 4, after introducing adaptive crossover and mutation, the Cauchy distribution, and exponential temperature decay, the conflict rate of the model's course scheduling results drops from 13.1% to 2.5%. Among them, adaptive crossover and mutation, as well as the Cauchy distribution, have the most significant impact on model performance. After the introduction of both, the conflict rate drops by more than 3%. This improvement scheme can effectively enhance model performance.

Table 4. Results of ablation experiment

Model	Adaptive crossover and mutation	Cauchy distribution	Exponential temperature decay	Conflict rate/%
1	×	×	×	13.1
2	√	×	×	9.5
3	×	√	×	9.6
4	×	×	√	10.3
5	√	√	×	7.2
6	√	×	√	6.4
7	×	√	√	5.1
8	√	√	√	2.5

4. Discussion

Since CUCS involves cross-coordination among multiple campuses, colleges, and majors, it needs to simultaneously align the five core elements: teachers, classes, courses, classrooms, and time. Moreover, it is prone to problems such as teacher time conflicts, classroom capacity mismatch, and unreasonable course times. Therefore, to improve the efficiency and quality of class scheduling, this study proposes an intelligent computer class scheduling method based on ISA-IGA.

In the basic performance test of the algorithm, ISA-IGA showed excellent convergence speed and solution accuracy. For the Oliver30 and Barma14 calculation examples, the algorithm required only 24 and seventeen iterations to converge, respectively. In terms of solution accuracy, the deviation rates of ISA-IGA for the two calculation examples were only 1.85% and 0.64%, and the average errors were as low as 0.03% and 0.11%, which were significantly better than the comparative algorithms, indicating that it has stronger global search capabilities and stability in complex optimization problems.

In the actual test scenario for intelligent class scheduling, the advantages of ISA-IGA were even more pronounced. Facing a complex class schedule with 2,500 students, 110 teachers, and 80 classrooms, the average runtime of this algorithm was 175.6 s, and it converged in only 45 iterations. In terms of course scheduling quality, the average conflict rate of ISA-IGA was only 2.5%. The average number of missed classes was only two, significantly reduced compared to the seven and six classes of the comparative algorithm, thereby ensuring the completeness and accuracy of class scheduling. Xiao et al. (2023) proposed a solution framework comprising a mixed-integer optimization model and a nested SA for the course scheduling problem, integrating class grouping and classroom assignment factors. This method could output high-quality solutions for an instance containing 450 students, ten batches, sixteen groups, and ten classrooms within four hours. Compared with the above methods, the research method was not only faster in solving problems but also had a lower conflict rate.

Algorithm performance indicators were transformed into a management decision-making basis: 96.4% classroom utilization rate provided a quantitative reference for teacher/classroom configuration, reducing administrative redundancy by 15%. A low conflict rate of 2.5% improved schedule stability and reduced administrative scheduling adjustment workload by 60%. Uniform distribution of class hours could increase student satisfaction by about 20%. Scenario analysis showed that academic managers could rely on algorithmic data to formulate scheduling strategies that "fix high-quality time slots for core courses and dynamically match idle resources for public courses." At the same time, they could optimize resource allocation between campuses based on the gradient of classroom utilization, freeing administrative personnel from manual course scheduling and shifting them to high-level management work of teaching resource coordination and personalized demand response.

In terms of resource utilization efficiency, ISA-IGA has achieved dual optimization of classroom utilization and balanced class distribution. Its average classroom utilization rate reached 96.4%, with a maximum of 98.8%, far exceeding the 86.0% and 89.7% of the comparison algorithms. The average class distribution balance was 0.89, with a maximum of 0.93, effectively avoiding the problems of continuous classes or over-concentration. Salman et al. (2025) proposed the mutation probability change method to address the excessively long computation time of GA in course scheduling. The average calculation time of this method was only 0.382 s, and the solution quality did not decrease. Compared with the above methods, the research method had a higher classroom utilization rate and a smaller conflict rate.

To sum up, ISA-IGA successfully addresses the problems of slow convergence, high conflict rate, and uneven resource utilization in traditional course scheduling algorithms by combining the IGA's coding method and evolutionary strategy with the SA algorithm's global search capability. This algorithm provides an efficient, practical technical solution for intelligent CUCSs. The model focuses on the efficiency needs of academic administrators in class scheduling, does not adequately explore the personalized needs of teachers' teaching preferences and students' learning experience, and does not establish a dynamic trade-off mechanism for multi-subject needs. Therefore, future work will consider constructing a multi-objective optimization model that incorporates teacher preferences, student experience, and teaching efficiency, and introducing the analytic hierarchy process to determine the weights of different subject needs. User-feedback interactive modules can also be developed to enable personalized customization of course schedules through iterative optimization.

5. Conclusion

Focusing on the efficiency and rationality requirements of intelligent CUCSs, this study carried out key technology research and reached the following conclusions: By systematically analyzing the hard constraints (such as resources not being reusable in the same period) and soft constraints (such as prioritizing professional courses during high-weighted periods) in the class scheduling problem, including teachers, classrooms, time, and other factors, and combining course period weights, class distribution evenness, and classroom utilization indicators, a scientific mathematical model of class scheduling was constructed to lay the foundation for algorithm solutions. In view of the shortcomings of traditional algorithms, an intelligent course scheduling algorithm that integrates IGA and ISA was proposed. IGA was optimized using decimal coding and adaptive cross-mutation, and Cauchy-distribution perturbation and exponential temperature attenuation were introduced to improve SA, thereby enhancing global search capabilities and convergence efficiency. Experiments showed that the number of convergence times in the Oliver30 and Barma14 calculation examples was more than 40% lower than that of RL-IGA and CS-IGA, and the deviation rate was the lowest at 0.64%. In a class scheduling scenario with 2,500 students and 110 teachers, the average running time was 175.6 s, the average conflict rate was 2.5%, and the average classroom utilization was 96.4%, all of which were better than those of the CA-HABC and BG-ACO algorithms. The results above show that the research method can effectively address the CUCS problem and provide reliable technical support for smart educational management. However, there are still limitations: the algorithm is only suitable for static scheduling scenarios and does not consider the dynamic scheduling needs arising from teaching task adjustments and temporary changes in teaching staff; the weight setting is based on a single-institution survey, and its universality across institutions and majors has yet to be verified. Future work will focus on developing a dynamic scheduling system and introducing a real-time perception module to enable adaptive adjustments to scheduling schemes. Simultaneously, the research sample across multiple types of universities can be expanded, and a dynamic weight-matching model can be constructed. Future research will also integrate multi-agent reinforcement learning to achieve collaborative optimization of personalized needs among teachers and students, and promote the upgrading of intelligent scheduling technology towards scenario-based, universal, and personalized approaches.

After reading this article, managers can reconstruct the course-scheduling decision-making process from three perspectives. First, the traditional experience-based process of "repeated adjustments to manual trial classes" is abandoned in favor of the quantitative decision-making process of "data modeling algorithm to solve local fine-tuning". It is based on the soft- and hard-constraint model and relies on IGA-ISA to quickly generate the optimal course scheduling plan. Second, scheduling decisions will be upgraded from "single campus autonomous planning" to "multi-dimensional resource coordination" and linked to pre-decisions such as teacher allocation and classroom construction, based on algorithm-derived indicators such as classroom utilization and class time allocation. Third, a new "Algorithm Performance Feedback" section will be added, incorporating scheduling conflict rates and teacher-student feedback into algorithm parameter optimization to form a closed-loop scheduling management process of "Decision Execution Feedback Iteration", achieving intelligent and refined control of scheduling work.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

Declaration of Artificial Intelligence (AI) Tools

The author used DeepL Write solely for language editing and to improve readability. The author reviewed and verified all content and takes full responsibility for the accuracy and integrity of the manuscript.

References

Ahmed, S. N., Routa, A. K., Kumar, P., and Vedanti, N. (2025). Attenuation of Ground Roll in Seismic Data Using Model-Based Inversion and Genetic Algorithm Available to Purchase. *Journal of the Geological Society of India,* 101(10), 1572-1577. doi:10.17491/jgsi/2025/174276

Akopov, A. S. (2024). An improved parallel biobjective hybrid real-coded genetic algorithm with clustering-based selection. *Cybernetics and Information Technologies*, 24(2), 32-49. doi:10.2478/cait-2024-0014

Alaka, H. M., Pnarba, M., Sarmehmet, B., and Eren, T. (2025). Course time scheduling problem for distance education considering server load balancing: A case of an engineering faculty. *Neural Computing and Applications*, 37(7), 5635-5653. doi:10.1007/s00521-024-10941-5

Ardana, N. B., Hastomo, W., and Arman, S. A. (2024). Development of adaptive lecture scheduling system using genetic algorithm: Case study of Ahmad Dahlan Institute of Technology and Business. *Journal of Computer Science Advancements*, 2(4), 200 – 212. doi:10.70177/jsca.v2i4.1310

Balci, S. G., Ersoz, S., and Luy, K. B. N. (2023). Optimization of indoor thermal comfort values with fuzzy logic and genetic algorithm. *Journal of Intelligent and Fuzzy Systems: Applications in Engineering and Technology*, 45(2), 2305-2317. doi:10.3233/JIFS-223955

Davison, M., Kheiri, A., and Zografos, K. G. (2025). Modelling and solving the university course timetable problem with hybrid teaching considerations. *Journal of Scheduling*, 28(2), 195-215. doi:10.1007/s10951-024-00817-w

Hafsa, M., Wattebled, P., Jacques, J., and Jourdan, L. (2025). Solving a multiobjective professional timetable using evolutionary algorithms at Mandarine Academy. *International Transactions in Operational Research,* 32(1), 244-269. doi:10.1111/itor.13276

He, X. (2024a). Prediction model of diabetes complications based on genetic engineering improved genetic algorithm optimized BP neural network. *International Journal of Public Health and Medical Research*, 2(1), 58-69. doi:10.62051/ijphmr.v2n1.07

He, Y., Huang, J., and Chen, W. Z. N. (2024b). Hybrid method of artificial neural network and simulated annealing algorithm for optimizing wideband patch antennas. *IEEE Transactions on Antennas and Propagation*, 72(1), 944-949. doi:10.1109/TAP.2023.3331249

Musatova, E. G., and Lazarev, A. A. (2023). Minimizing the total weighted duration of courses in a single machine problem with precedence constraints. *Automation and Remote Control*, 84(9), 1005-1015. doi:10.1134/S0005117923090102

Poudel, Y. K., and Bhandari, P. (2024). Control of the BLDC motor using ant colony optimization algorithm for tuning PID parameters. Archives of Advanced Engineering Science, 2(2), 108-113. doi:10.47852/bonviewAAES32021184

Ravindra, B. B. (2025). Nonlinear optimization framework for educational timetable scheduling using differential evolution. *Communications on Applied Nonlinear Analysis*, 32(5), 568-583. doi:10.52783/cana.v32.3181

Rizqy, M. E., Faqih, A., and Dwilestari, G. (2025). Naive Bayes optimization by implementing genetic algorithm in sentiment analysis of BCA mobile reviews. *Journal of Artificial Intelligence and Engineering Applications*, 4(2), 771-777. doi:10.59934/jaiea.v4i2.750

Salman, R., Suprapto, S., Irfandi, I., and Hutajulu, O. Y. (2025). Optimization of genetic algorithm computation time with mutation probability variations in course scheduling. *Jambura Journal of Electrical and Electronics Engineering*, 7(1), 77-81. doi:10.37905/jjeee.v7i1.28286

Sha, X. (2024). Time series stock price forecasting based on GA – LSTM optimization. Advances in Economics, *Management and Political Sciences*, 91(1), 142-149. doi:10.54254/2754-1169/91/20241031

Thepphakorn, T., and Pongcharoen, P. (2023). Modified and hybridised bi-objective firefly algorithms for university course scheduling. *Soft Computing*, 27(14), 9735-9772. doi:10.1007/s00500-022-07810-5

Waongo, M., Laux, P., Bliefernicht, J., Coulibaly, A., and Traore, S. B. (2025). A genetic algorithm approach for location-specific calibration of rainfed maize cropping in the context of smallholder farming in West Africa. *Agricultural Sciences*, 16(1), 89-111. doi:10.4236/as.2025.161006

Xiao, L., Parikh, P. J., and Zhang, X. (2023). Course scheduling problem with cohort group and room considerations. *International Journal of Industrial and Systems Engineering*, 45(4), 458-483. doi:10.1504/ijise.2022.10047812

Xu, M., Wang, X., Leng, P., Zhang, M., Wang, H., Hua, J. and Kang, M. (2025). Multi-Objective Planting Planning Method Based on Connected Components and Genetic Algorithm: A Case Study of Fujin City. *Smart Agriculture,* 7(5), 36-145. doi:10.12133/j.smartag.SA202504012

Yan, Y., Zhang, W., and Li, L. Z. (2023). Simulated annealing algorithm optimized GRU neural network for urban rainfall-inundation prediction. *Journal of Hydroinformatics*, 25, 3-4, 1358-1379. doi:10.2166/hydro.2023.006

Zcan, E., De Causmaecker, P., Vanden Berghe, G., and Goossens, D. (2025). Preface: The practice and theory of automated timetabling (2022). *Journal of Scheduling*, 28(2), 157-158. doi:10.1007/s10951-024-00830-z

Zhuniskhanov, M. and Suliyev, R. (2024). Optimization algorithms overview for course scheduling. *Suleyman Demirel University Bulletin of Natural and Technical Sciences*, 55(2), 17-23. doi:10.47344/sdubnts.v55i2.540

Baimei Xu is Dean and Associate Researcher at the Continuing Education College, Jiangsu Vocational College of Finance and Economics. She holds a bachelor's degree in Information Management and Information Systems from Nanjing University of Science and Technology. Her research interests include higher education management. She also teaches courses such as Computer Fundamentals, Information Management Technology, and Photoshop. She has chaired university-level online open courses on network systems and artificial intelligence. She has received one first prize in the university-level Teaching Achievement Award and Microcourse Competition, and one third prize at the provincial level. She has led or participated in eight provincial, municipal, and departmental research projects, published one monograph on local Huai'an culture, more than ten academic papers, and holds four authorized patents (two of which have been applied in practice).