J. COMBIN. MATH. COMBIN. COMPUT. 127b (2025) 7979--8000

Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Time of Resource Allocation Decision Model Based on Multi-Objective Optimization in School Information Technology Promotion

Huiping Mei[1,✉]

[1] *Zhejiang Fashion Institute of Technology, Ningbo, Zhejiang, 315211, China*

ABSTRACT

Aiming at the allocation of teaching resources for school affairs scheduling, a decision-making model for school affairs scheduling is designed based on a multi-objective optimization model. The "conflict detection and repair" module is added after the "initial population generation" operation in the traditional genetic algorithm, which decouples the scheduling model and meets the needs of scheduling decision-making. The designed method is compared with the standard genetic algorithm and stochastic two-point crossover genetic algorithm on the data set, and then the efficiency of resource allocation for school scheduling is improved by solving an example problem. The average faculty satisfaction with scheduling is 2.8, which is about 17% higher than the second place NPGA. Applying the algorithms to a college scheduling project, the feasible solutions of the algorithms in this paper satisfy all the various constraints, and the results of the three-stage style algorithm in the self-selected course scheduling mode yield better solutions than the baseline algorithm based on the course set in any of the arithmetic cases. This paper provides an informative solution path for the allocation of school scheduling resources, which can satisfy the course allocation needs of the three parties: teachers, students and schools.

Keywords: constraints, multi-objective optimization, genetic algorithm, scheduling problem

1. Introduction

With the continuous development of science and technology, the need for optimization of complex real-world problems becomes more and more urgent. Many real-world problems often involve multiple objectives and multiple tasks, and there are usually interactions and conflicts between these objectives and tasks [1-3]. Traditional single-objective optimization algorithms cannot effectively deal with such multi-objective multi-task optimization problems, so researchers have gradually focused their attention on multi-objective multi-task evolutionary optimization algorithms, which are a class

✉ Corresponding author.
E-mail address: 13777123664@163.com (H. Mei).

Received 01 January 2024; Revised 12 March 2024; Accepted 20 December 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-436

© 2025 The Author(s). Published by Combinatorial Press. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

of heuristic optimization methods, in order to find more flexible and adaptable solutions [4-6]. Multi-objective optimization problems are very common in practical applications, and typical examples include engineering design, financial portfolios, energy management, and so on. These problems usually involve multiple conflicting objectives, and the decision maker needs to find a set of nondominated solutions among these objectives, constituting a Pareto frontier [7-8]. However, traditional single-objective optimization algorithms cannot directly cope with multi-objective problems because they cannot handle the trade-offs and trade-offs among the objectives, and the complex shape of the Pareto frontiers and the huge search space pose a great challenge to the robustness and global search capability of the algorithms [9-10]. At the same time, multi-objective optimization is also a research area of great interest. Informatization is an inevitable trend in the development of information society. As an important base for the concentration of advanced productive forces and advanced culture in the whole society, it is necessary for higher education institutions to take the lead in realizing informatization, so as to promote and guide the informatization process of the whole society. Informatization of higher education institutions can greatly improve the teaching, research level and management efficiency of the whole school, and better integrate all kinds of school resources and social resources to serve the work of the school. In the process of school informatization, the correlation and mutual influence between different tasks exist from time to time [11-12]. Traditional single-task optimization can hardly take multiple tasks into account when solving such problems, therefore, it is necessary to start exploring how to combine the superiority of the resource allocation decision model to solve the multi-objective multi-task optimization problem, and strive to obtain a more comprehensive and efficient solution, and then promote the sustainable development of school affairs informatization [13-15].

With the continuous improvement of the government's macro-control and management system of school informatization, the market space and external resources of universities are severely limited, which makes the importance of the resource utilization problem more prominent. Literature [16] constructed a multi-objective multi-stage fuzzy optimization dynamic resource allocation model based on Markov Decision Process (MDP), and verified the validity and feasibility of the model through empirical analysis, which can derive the resource allocation strategy that prompts the optimization of the overall project benefits. Literature [17] analyzes the problems existing in the financial resource management of colleges and universities in the new period, and then uses the intelligent optimization algorithm to carry out multi-objective intelligent optimization of the financial resource management and allocation methods of colleges and universities, aiming to improve the scientific rationality of the financial budget of colleges and universities, and then to ensure the smooth implementation and development of the various tasks of colleges and universities. Literature [18] proposed an adaptive multi-objective teaching-based optimization (AMO-TLBO) algorithm for dynamic resource allocation strategy with the support of cloud computing, and its feasibility and effectiveness were verified through evaluation experiments, which can bridge the gap between frequently changing customer demands and service availability infrastructure. Literature [19] improves the teaching-based optimization algorithm and proposes a distributed manufacturing resource parallel optimization allocation method based on it, and the superior performance of the proposed method is verified through simulation experiments, which can be used to deal with trade-offs between multiple objectives and thus obtain a better manufacturing resource allocation scheme. Literature [20] constructs a multi-objective framework for multi-access edge computing resource allocation based on the Teaching-Learning-Based Optimization (TLBO) algorithm, which is confirmed by simulation experiments to be feasible and practical, and it achieves high throughput, low latency, high energy efficiency and user fairness.

In this paper, we first mathematically analyze the school scheduling problem and model the constraints involved, such as teaching plans, courses, teachers, classes, classrooms, time, and school districts. Then the traditional genetic algorithm is optimized, the gene and chromosome coding scheme is designed, the initial population generation method based on greedy algorithm is improved, and the conflict detection and repair mechanism is added, so that the repaired schedule corrects the errors while taking into account the rationality and other constraints. The fitness evaluation function is sufficiently quantized to impose constraints on the end conditions of the algorithm. The performance of coding, conflict detection and elimination, and local search operators are verified on the dataset, and then the overall performance is compared to analyze whether the improved genetic algorithm in this paper has a faster convergence speed or improves the diversity of individuals to a certain extent to achieve the purpose of enhancing the search space and jumping out of the local optimum. Finally, the reasonableness of the design model of the article is tested with the actual scheduling problem.

2. Analysis and modeling of school scheduling problems

2.1. *Overview of school scheduling issues*

The problem of scheduling is, in fact, a problem of scheduling and allocation of school instructional resources. In scheduling, the main factors involved are teaching plan, course, teacher, class, classroom, time, campus and other factors. The use of computers to solve the scheduling problem has become an important means of teaching in colleges and universities.

2.2. *Constraints on the school scheduling problem*

The main factors involved in school scheduling are the instructional program, curriculum, teachers, classes, classrooms, time, and school district. These factors are interrelated and at the same time constrained each other. For the teaching program, it is mainly formulated by schools and colleges according to the actual situation. Instructional program mainly consists of courses, teachers and classes, i.e., a particular course in a particular class is taught by a particular teacher, which can be taken as an instructional event. School districts, classrooms, and time can form spatio-temporal combinations. Then the scheduling problem is to rationally arrange a series of instructional events in multiple spatio-temporal combinations of school districts, classrooms and times, so that the resulting schedule is conflict-free and humane. The constraints are the main basis for avoiding conflicts and ensuring humanization. The constraint relationship between the factors in scheduling is shown in Figure 1.

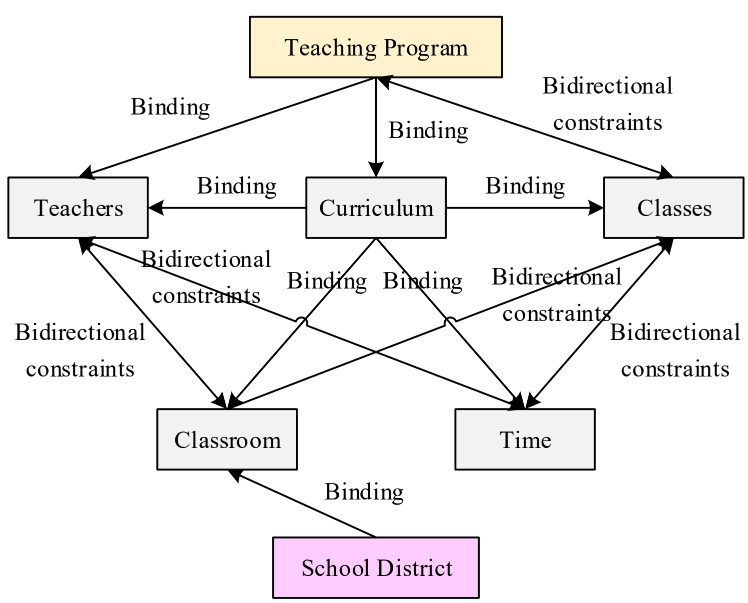

Fig. 1. Constraint relationship between various factors of course scheduling

2.3. Mathematical modeling of the school scheduling problem

It is clear from the previous introduction that scheduling is a process in which various scheduling factors are rationally combined and continuously optimized. In order to easily describe the scheduling problem, the actual scheduling problem needs to be transformed into a mathematical problem. The scheduling problem is modeled using mathematical language. The mathematical model is then processed and calculated to find the better result in the model, i.e., the better schedule corresponding to the scheduling problem.

2.3.1. Mathematical modeling of factors in the scheduling problem. In order to meet the actual needs of school scheduling, this paper proposes a way of scheduling all teaching weeks simultaneously. First, we establish corresponding mathematical models for each factor in scheduling, which are defined as follows:

Course set: $C=\{c_1,c_2,\ldots,c_m\}$, the total number of courses is M.

Classes collection: $G=\{g_1,g_2,\ldots,g_n\}$, the total number of classes is N. The number of people in each class is $\{gx_1,gx_2,\ldots,gx_n\}$.

Teacher set: $T=\{t_1,t_2,\ldots,t_p\}$, the total number of teachers is P.

Classroom set: $R=\{r_1,r_2,\ldots,r_k\}$, the total number of classrooms is K. The capacity of the classrooms is $\{x_1,x_2,\ldots,x_k\}$.

Time set: $S=\{s_1,s_2,\ldots,s_q\}$, the total number of time sets is Q.

The school district set: $W=\{w_1,w_2,\ldots,w_v\}$, the total number of school districts is V_0.

2.3.2. Mathematical description of constraints in scheduling problems. Based on the organization and analysis of the constraints in the previous section, some major hard and soft constraints were derived. Before proceeding with the scheduling, the constraints are first modeled in a mathematical language way [21]. The modeling of each constraint is shown below:

Hard constraints:

1) A class can only take one course at the same time. It can be expressed as:

$$\sum_{p=1}^{p}\sum_{k=1}^{K}\sum_{m=1}^{M} t_p s_q w_r r_k g_n c_m \le 1 \tag{1}$$

2) A faculty member can teach only one course at a time. It can be expressed as:

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{m=1}^{M} t_p s_q w_v r_k g_n c_m \le 1 \tag{2}$$

3) Only one course can be scheduled in a classroom at any one time. It can be expressed as:

$$\sum_{p=1}^{P}\sum_{n=1}^{N}\sum_{m=1}^{M} t_p s_q w_v r_k g_n c_m \le 1 \tag{3}$$

4) The number of seats in a classroom must be greater than or equal to the number of students attending the class. It can be expressed as:

$$\sum_{n=1}^{L} gx_n \le x_k \tag{4}$$

where L is the number of classes that are also taught in the classroom. gx_n indicates the number of students in the n rd class attending classes in that classroom.

5) Courses that require a specific classroom or time slot are scheduled in a specific classroom or time slot.

If a course requires a certain classroom or time slot, the corresponding classroom or time slot needs to be set in advance during scheduling, and will not be modified afterwards. If a course requires a certain type of classroom or time slot, it is sufficient to assign these courses to the corresponding classrooms or time slots during scheduling. Courses with specific requirements need to be prioritized during scheduling.

6) Courses and credit hours in the instructional program may not be added or subtracted at will.

The constraint is that in the process of scheduling classes, the teaching information such as courses, classes, teachers, and total hours should be strictly in accordance with the teaching plan, and the requirements of the teaching plan should not be arbitrarily increased, decreased, or modified.

7) Two consecutive morning or afternoon classes for the same teacher or student cannot be scheduled on different campuses.

This constraint means that for teachers and students who are scheduled for two consecutive morning or afternoon classes, the scheduler needs to determine whether the corresponding two classrooms are on the same campus. If they are not on the same campus, the scheduling is considered a failure. For two different campuses in close proximity to each other, they can be treated as the same campus when scheduling classes.

8) Classes are not scheduled for certain periods of time, such as every Thursday afternoon when the school does not schedule classes at all. According to the overall requirements of the school, classes cannot be scheduled during some specific time periods. In this paper, we can according to the school requirements, can not be arranged to teach the time period set up in advance, in the subsequent scheduling courses can not be arranged to these time periods.

Soft constraints:

1) For teachers with special requirements, they should try to fulfill them. For example, some teachers are far away from home and cannot be scheduled to teach in the morning 1-2 periods.

In this paper, in order to better describe and distinguish the importance of teachers' special requirements, all special requirements are divided into three levels: especially important, important and generally important. The weight values of the three grades are set to 0.9, 0.4 and 0.2 respectively.Finally, the average value of the corresponding weights of all teachers' special requirements is calculated. The calculation formula is:

$$m_1 = \frac{1}{w}\sum_{i=1}^{w}\alpha_i \tag{5}$$

where w is the number of special requirements for all teachers and α_i is the weight value corresponding to the i rd special requirement. If the i th special requirement is satisfied, α_i is the weight value corresponding to the rank. Conversely, α_i is 0.

2) Lectures are scheduled in the corresponding teaching week as much as possible according to the teaching plan or the teacher's preferred teaching week. In this paper, h_t is used to denote the number of all courses in the teaching week preferred by a teacher t for a certain course, and h_t' denotes the number of the course in the preferred teaching week by a teacher t after the actual scheduling. The ratio of the number of courses actually scheduled in the preferred instructional week for all teachers to the number of courses in all preferred instructional weeks in the original setup is then calculated. The formula for this is:

$$m_2 = \frac{\sum_{t=1}^{p} h_i}{\sum_{t=1}^{p} h_i} \tag{6}$$

3) Each course in the class should be scheduled at appropriate intervals during the week so that it is easy for the teacher to prepare for the class and for the students to prepare and review in time after the class.

According to the weight values in the table, the average weight value of all courses assigned in the class in a week can be obtained. Its calculation formula is:

$$wei_n = \frac{1}{sn} \sum_{s=1}^{xn} w_g \tag{7}$$

where sn is the total number of courses in class n and w_s is the weight value corresponding to the course s allocation in that class.

The merit of the class schedule can be expressed as the average of the weight values of the course assignments within all weeks for all classes, which is calculated by the following formula:

$$m_3 = \frac{1}{20^* N} \sum_{n=1}^{N} \sum_{z=1}^{20} wei_{n,z} \tag{8}$$

where N is the number of all classes.

4) The number of people arranged in the classroom is as close as possible to the classroom capacity to improve the classroom resource utilization rate.

In this paper, the ratio of the number of people actually attending classes to the classroom capacity is used to measure the resource utilization rate of the classroom, which is calculated by the following formula:

$$pee_k = \frac{\sum_{n=1}^{L} gx_n}{x_k} \tag{9}$$

The merit of a class schedule can be expressed as the average of the resource utilization of all classrooms in which instruction is scheduled, which is calculated using the following formula:

$$m_4 = \frac{1}{K} \sum_{k=1}^{K} per_k \tag{10}$$

where K is the number of classrooms in which all courses are scheduled.

5) Courses with high difficulty coefficients are scheduled in better time slots, and courses with low difficulty coefficients are scheduled in other time slots. In general, the priority order of time slots is: 1-2 periods in the morning → 1-2 periods in the afternoon → 3-4 periods in the morning → 3-4 periods in the afternoon → 1-2 or 1-3 periods in the evening.

During the scheduling process, the constraint can be measured by the average of the difference between the difficulty level of all courses and the weights corresponding to the scheduled time slots, which is given by the formula:

$$m_5 = \frac{1}{M} \sum_{m=1}^{M} \left(1 - \left|\delta_m - \delta_m\right|\right) \tag{11}$$

where M is the number of times all courses are taught, δ_m is the weight value corresponding to the difficulty factor of course m, and δ'_m is the weight value corresponding to the time period in which course m is scheduled.

6) Instruction for each course in the class is scheduled in the same classroom to the extent possible.

The ratio of the number of all courses to the number of all classrooms in which lectures are scheduled is used to measure this constraint. The formula is calculated as:

$$m_6 = \frac{\sum_{n=1}^{N} cg_n}{\sum_{n=1}^{N} rg_n} \tag{12}$$

where cg_n indicates the number of all courses in class g_n, and rg_n indicates the number of classrooms in which all courses in class g_n are taught.

7) For special courses to be arranged in special time slots or classrooms, for example, physical education classes try to be arranged in 3-4 periods in the morning or 3-4 periods in the afternoon and so on.

For physical education classes, we can set the weight values of the five time periods in a day as 0.1, 0.8, 0.3, 0.9, 0. The constraint can be measured by the average weight value of the scheduling of physical education classes for all classes. The formula for this is:

$$m_7 = \frac{1}{ZX}\sum_{z=1}^{ZX} \varepsilon_z \tag{13}$$

where ZX is the number of physical education classes for all classes and ε_z indicates the weight value corresponding to the z rd physical education class.

8) For cross-campus classes, try to avoid scheduling 1-2 periods in the morning and 1-2 or 1-3 periods in the evening.

Different weight values were set for each of the five time periods in a day, which were 0.1, 0.7, 0.9, 0.8, and 0.1, respectively.Then, the average weight value of all the courses with cross-campus classes was calculated with the formula:

$$m_g = \frac{1}{D}\sum_{m=1}^{D} \omega_m \tag{14}$$

where D is the number of times all cross-campus classes are taught. ω_m denotes the weight value corresponding to the time slot scheduled for the mth time of a cross-campus course.

3. Design of scheduling algorithm based on improved genetic algorithm

3.1. Algorithm general flow

According to the general steps of the genetic algorithm, a general flow chart is drawn as shown in Figure 2.

This process is designed according to the standard genetic algorithm process, and the “conflict detection and repair” module is added after the “generation of the first generation of the population” operation, which is a function specially designed to meet the requirements of the quality evaluation of the class schedule. This function can be used separately [22].

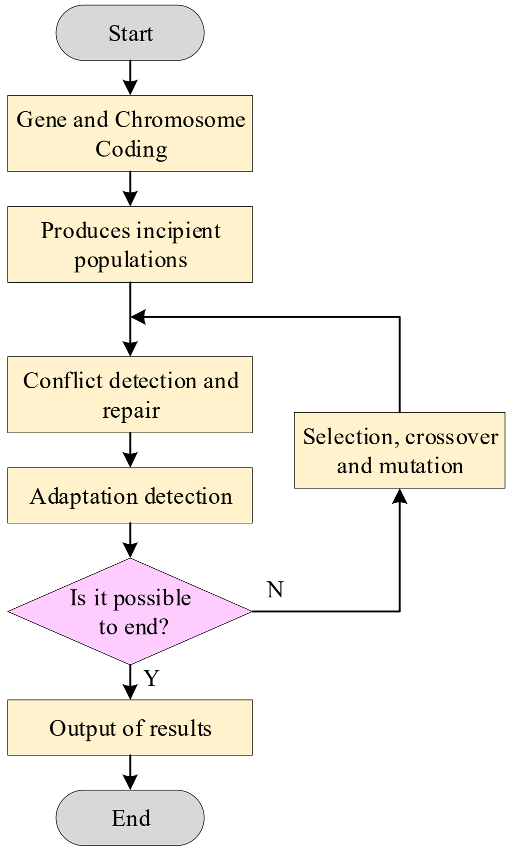

Fig. 2. General Flow Chart Of Algorithm

3.2. Coding and Primary Populations

3.2.1. Gene and chromosome coding. Genetic coding is where 5-tuples are encoded. This work is the basis of the entire algorithm, and a high-quality encoding scheme contributes greatly to the efficiency of the algorithm [23].

1) A gene is an instance of a 5-tuple, i.e., 1 grid in the schedule of 1 class.

2) A chromosome is a gene sequence consisting of 20 genes in the schedule of a class.

3) An individual is a combination of chromosomes of all classes involved in scheduling, a collection of class schedules, a solution to a scheduling problem, a scheduling scheme. What this algorithm finally gets, is a relatively optimal individual.

4) A population is a combination of many individuals, that is, many kinds of class schedules. Through the algorithm, the implementation of the selection of individuals in the population, genetic crossover and mutation, in order to obtain the optimal solution, is the essence of this algorithm.

3.2.2. Initial population generation method based on greedy algorithm. Having determined the genotype coding, let's design the algorithm to generate the 1st individual and the primordial population.

As an NPC optimal solution problem, the obtaining of the 1st individual is very important, and if it is cleverly designed and the algorithm is appropriate, it will greatly reduce the efficiency of the subsequent search.

The algorithm for generating the initial population is described in detail below.

Step 1, for each class location with restricted scheduling time (e.g., special teaching places such as staircase classrooms), set up a scheduling probability matrix:

$$P_a = \begin{bmatrix} p_{11} & \cdots & p_{51} \\ \vdots & \ddots & \vdots \\ p_{14} & \cdots & p_{54} \end{bmatrix} \tag{15}$$

Step 2, a scheduling probability matrix is set for each teacher:

$$P_n = \begin{bmatrix} p_{11} & \cdots & p_{51} \\ \vdots & \ddots & \vdots \\ p_{14} & \cdots & p_{54} \end{bmatrix} \tag{16}$$

Step 3, Set the class sequence. Sort all the n classes involved in scheduling (this ordering can be randomized for $n < 20$, or manually intervened if n is large) to produce the 1st class sequence

$$C_1 = \{c_1, c_2, \cdots, c_n\} \tag{17}$$

Step 4, the scheduling chromosomes for each class in sequence C_1 are obtained in turn (expanded below). The above n chromosomes form the 1st individual. That is, the 1st solution to the problem.

Step 5, a different ordering of equation (17) yields C_2.

Step 6, repeat step 4 to get the 2nd individual.

Step 7, repeat steps 5 to 6 to get M individuals, forming the initial population. The algorithm ends.

Specifically, in step 4 above, how to give the scheduling plan for each class? We design a mechanism to determine the "suitable time slice" according to the following rules, in accordance with the order of classes (in fact, when the program is executed to step 4, this order has already been given to place each course on the suitable time slice).

First, the rule of correctness. The used time slice cannot be selected again, and a time slice conflict matrix is set for each class, each teacher, and each location:

$$ConflictMatrix_i = \begin{bmatrix} cm_{11} & \cdots & cm_{51} \\ \vdots & \ddots & \vdots \\ cm_{14} & \cdots & cm_{54} \end{bmatrix} \tag{18}$$

Initialize the value in each position of the above matrix to 1 to indicate "ready to schedule". Scheduling process, if the use of a time slice, it is necessary to the corresponding class, teacher, class location of the time slice of the conflict matrix of the corresponding position on the 0, said "occupied" (that is, it is very not recommended to row in this position).

Second, the principle of optimization. Each time you choose to place a course location, first according to the previous rule to determine the optional location, calculate the probability value of each location. This probability value is defined as "the probability of the location of the course × the probability of the instructor", which is determined separately. Then a number of grids are selected in order of highest to lowest probability and the corresponding courses are placed.

3.3. *Conflict detection and remediation*

In fact, the algorithm that generates the initial population, as designed, is conflict-free, and therefore, this section is designed to be a redundant check for the initial population. However, this detection and repair is needed for the offspring after the implementation of the genetic operation.

Conflict detection refers to the correctness test for a given individual, where the correctness constraints steps are:

Step 1, the individual to be tested is populated.

Step 2, verify the correctness constraints. The requirement that "a teacher can only teach at most one class at one location in one time slice" is implemented by searching each time slice from top to bottom, and under normal circumstances, a teacher will appear at most once. If a teacher appears more than 2 times, then check whether it is a "combined class", if it is a "combined class", then the test passes, otherwise an error is reported and the test continues to the next column.

Step 3, a test report is given. The report should indicate the location of errors, missing items, etc. End of inspection.

If a conflict is detected, there is an issue to be discussed regarding the choice of conflict repair method. Some research supports a "random variation" algorithm that randomly modifies the location of the conflict, for example, if two classes are scheduled in a classroom in time slot t_{12}, then one of the classes is randomly switched to another free time slot, and then tested again until there is no conflict.

3.4. Adaptation function design

The temporal fitness of an individual is defined as:

$$F_1 = k_1 F_a + k_2 F_n \tag{19}$$

where k_1 and k_2 are the weights. Obviously, the larger the value of F_1, the more friendly the program is to the location of the class and the teacher's schedule.

Here is an assessment of the degree of quality of the schedule (i.e., a chromosome) itself for a class.

The dispersion d of each course is defined as the ratio of the number of days the course is taught in a week m to the number of times the course is taught n:

$$d = \frac{m}{n} \tag{20}$$

That is, if the course meets only 1 time a week, then the dispersion is 1. If the course meets 2 times in a week, it needs to be scheduled on 2 days for the dispersion to reach 1.

Define the total individual dispersion as the sum of all course dispersions:

$$F_d = \sum_i d_i \tag{21}$$

Evaluation of self-study sessions.

Define self-study class adaptation F_z as the product of the values in each row of the table and the coefficients in the corresponding positions of the table, and then sum them.

Define the individual's class fitness as:

$$F_2 = k_3 F_d + k_4 F_z \tag{22}$$

where k_3 and k_4 are the weights. Obviously, the larger the value of F_2, the more reasonable the program's curriculum decentralization and self-study course arrangement.

Thirdly, the problem of arranging the time for moral education activities and teachers' collective preparation time. Considering the exactness of the algorithm, it is categorized separately.

Define this article as Penalty P, if the number of conflicts occurring in the chromosome for moral education activities is E_1 and the number of conflicts occurring in the lesson preparation arrangement is E_2, then:

$$P = k_5 E_1 + k_6 E_2 \tag{23}$$

where k_5 and k_6 are the weights. Obviously, the smaller the value of P, the more the program can meet the special requirements of these two aspects. Combining the above three considerations, the individual fitness function is defined as:

$$F = F_1 + F_2 - P \tag{24}$$

If the overall fitness of the population needs to be examined, for a population of size M, the population fitness can be defined as:

$$F_{group} = \frac{1}{M}\sum_i F_i \tag{25}$$

The distribution of population fitness can be examined by calculating the variance of individual fitness, taking full account of the effect of the value of k (since the effect of the value of k would put the value of individual fitness in the order of a few thousand):

$$\sigma_{group}^2 = \frac{1}{M}\sum_i \left(F_i - F_{group}\right)^2 \tag{26}$$

3.5. Selection, crossover and mutation

After the initial generation of the population is produced, genetic operations are performed to obtain the 2nd and 3rd ...... n st generation populations from which the optimal individuals are selected.

The genetic operations (also called genetic operators) are selection, crossover and mutation.

By introducing probabilities, individuals with low fitness also have a certain probability of being selected into the next generation of evolution. The probability of individual j being selected is:

$$p_j = F_j \Big/ \sum_i F_i \tag{27}$$

As:

$$\sum_i p_i = 1 \tag{28}$$

Then the algorithm for selecting individual j according to the above probability is: divide interval $(0,1]$ into M subsections in the order of $p_1, p_2, \cdots, p_M$ (M is the population size), let the 1st subsection be $(0, p_1]$, then the jth subsection is:

$$\left(\sum_{i=1}^{j-1} p_i, \sum_{i=1}^{j} p_i\right] \tag{29}$$

Generate 1 random number ξ in the interval $(0,1]$ if:

$$\sum_{i=1}^{j-1} p_i < \xi \le \sum_{i=1}^{j} p_i \tag{30}$$

Then select individual j. Repeat this to select M offspring. This method of selection is called roulette selection.

Since the principle of probability can only be demonstrated in multiple experiments, the above selection may be biased when M is small, resulting in the highly adapted individuals not being selected instead.

The following is an improvement on the roulette algorithm.

Still considering a population with initial and offspring sizes of M, first find the expected number of survivors of individuals in the initial generation:

$$n_i = Mp_i \tag{31}$$

If $n_i \ge 1$, take the integer part $[n_i]$ of n_i, then for each i for which an integer part exists, take $[n_i]$ times into the child. If made:

$$N = \sum_i [n_i] \tag{32}$$

Then the first N individuals of the offspring are given.

Then take the fractional part $\left(n-\left[n_i\right]\right)$ of each n_i and arrange this fractional part in descending order and take out the first $\left(M-N\right)$ individuals.

At this point, M children are obtained.

The parameter that controls whether the crossover operation occurs is the crossover rate p_c and, again, using the random number $\xi \in \left(0,1\right]$, if the

$$\xi < p_c \tag{33}$$

Crossover occurs randomly at a given location, otherwise it does not occur.

3.6. Closing conditions

The end conditions of the algorithm are as follows:

1) Reaching the target fitness. According to the design for the fitness function in this algorithm, the higher the value, the more optimized the solution is represented. Then, if there is:

$$F > F_s \tag{34}$$

where F_s denotes the end value of fitness. That is, if the fitness test value of the current population or individual reaches a pre-set higher level, it represents that a quite reasonable program has been obtained and the algorithm can be ended.

2) Reaching the number of iterations. As an NPC problem, it is not possible to let it execute indefinitely if the goal has not been reached, so it is necessary to set an iteration number Gen of about 20 to 50. In particular, if the number of iterations is set to 1, it means that the optimal solution arises right in the initial population.

3) The rate of convergence of adaptation is below the threshold. Depending on the choice of each parameter in the algorithm, the speed of convergence of the fitness function may vary considerably. If the convergence speed in the evolutionary process is below a certain threshold, i.e:

$$\left|F'-F\right| < f_{th} \tag{35}$$

At this point the point of continuing the iteration is lost and the algorithm can end.

4. Genetic algorithm-based scheduling simulation

The above is used in dataset experiments to verify that the scheduling problem model, related algorithms and related operations in the genetic algorithm designed in this paper are of good practicability and validity, and through comparative experiments with standard genetic algorithms, or genetic algorithms in the literature to verify whether the improved genetic algorithms in this paper have a faster convergence speed, or improve the diversity of the individuals to a certain extent to achieve the enhance the search space and jump out of the local optimum.

4.1. Performance analysis of the improved genetic algorithm

The coding scheme in more literature is a fixed coding length for each teaching task code. In this regard, the variable length coding scheme designed in this paper can still have good adaptability and scalability for courses with different weekly hours compared to it, for example, different weekly hours only need to change the number of bits in the genes representing the teaching time. In terms of classroom numbering, this paper can achieve that a course can be taught in different classrooms at different times, while the traditional algorithm can only use one classroom. In terms of teaching week, this paper can meet the teaching reality, indicating the beginning and ending weeks of the course. In the experimental data, there are a total of 979 lessons in the course, of which 251 lessons are in the course with weekly credit hours of 2, 263 lessons in the course with weekly credit hours of 4, and 465 lessons in the course with weekly credit hours of 6. In this paper, a variable length coding scheme is used, where the total length of the decimal code for each chromosome is a numeric representation of 17586 bits of decimal.

In the scheduling algorithm designed in this paper, conflict detection and conflict elimination are both performed after initializing the population and genetic operations to ensure that the class schedule can satisfy the basic constraints and hard constraints. Table 1 shows the average value of scheduling conflict detection before and after scheduling conflict elimination after initializing the population for 30 times on the scheduling data.

From the table, it can be seen that in the scheduling algorithm after the initialization of the population is bound to occur scheduling conflicts, which “the same time classroom and the course and its required classroom capacity and type of conflict number” is even as high as 4481.62. In this regard, this paper designed a number of conflict elimination algorithms to eliminate the conflict that occurs.

Table 1. Lecture conflict

Conflict description	Before the conflict is eliminated	Post-conflict elimination
The same time classroom and course and the number of classroom capacity and type conflict	4481.62	0
The number of teachers and courses in the same time	78.17	0
The number of classes and courses in the same time	136.25	0
The number of courses and the number of course conflicts	2.05	0
Time conflict between teachers and regular meetings	84.35	0

4.2. Simulation results and analysis

4.2.1. Local search operator performance analysis. The local search operator designed in this paper aims to speed up the convergence speed with the enhancement of local search ability. At the same time, this paper adopts the optimal individual retention strategy, if the initial population can have a higher fitness function value after the local algorithm, then it can be considered that it can have a better effect in the later genetic. For the local search operator performance analysis experiment is only after initialization of the population, conflict detection, conflict elimination and local search operator operation, not genetic operator (selection, crossover and mutation) operation, this is because at this time only want to consider the effect of the use of the local search operator, for which the initialization of 1,000 chromosomes for the experiment. The fitness values of the initialized populations using the local search operator and without it, the number of chromosomes in each population is 500.Figure 3 shows the comparison of the fitness of the initialized chromosomes. Since there is a certain amount of randomness in the initialized populations, the use of the local search operator does not guarantee that the fitness of each chromosome is higher than that of the initialization without the use of the local search operator.

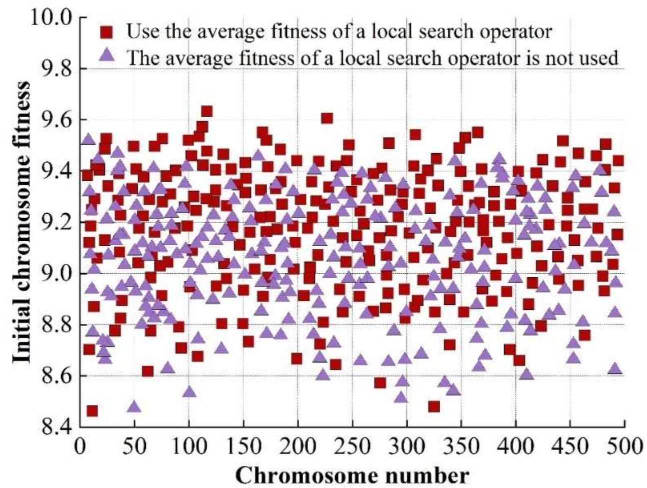

Fig. 3. Initial chromosomal fitness comparison

Table 2 shows the average fitness of initialized chromosomes. From the analysis of the average fitness value of chromosomes in the table, the fitness value of chromosomes in the initialized population using the local search operator is slightly higher than the fitness value of chromosomes in the initialized population without using the local search operator, which shows that the local search operator helps to increase the fitness value of chromosomes.

Table 2. The average fitness of the initialization chromosome

	Use local search operators	No local search operator is used
Average fitness for each chromosome initialization	9.1854	9.0962

4.2.2. Scheduling results from different perspectives. The algorithm designed in this paper is compared with standard genetic algorithm as well as stochastic two-point crossover genetic algorithm. The results are shown in Figure 4. Among them, the standard genetic algorithm, the selection operation uses the roulette method, and the crossover and mutation operations use the stochastic two-point crossover method using the optimal individual retention method; the stochastic two-point crossover uses the traditional selection, crossover, and mutation strategies; the coding scheme, initialization population, scheduling conflict elimination, fitness function design, and genetic algorithm related parameter settings in the rest of the genetic algorithm related operations of the standard genetic algorithm and the stochastic two-point crossover etc. are consistent with this paper. The relevant values are the results of the average of the scheduling experimental data for a number of experiments.

From the observation and analysis of the figure, it can be seen that the standard genetic algorithm reaches convergence in about 65 generations; the stochastic two-point crossover method enters convergence in about 82 generations, although the convergence speed is slower, but the value of the fitness function is much higher than that of the standard genetic algorithm; the improved genetic algorithm designed in this paper converges in about 37 generations, and has the characteristics of fast convergence speed and strong global search ability, which is due to the local search operator in this paper. Optimized for the scheduling of specific problems in the part of the solution target for local search. For the performance of the algorithm, there are two indicators: global search capability and local search capability. Global search capability refers to the ability to find the optimal solution in the whole world, and local search capability refers to the better solution algorithm based on the existing solution from a local analysis of a specific problem. In the actual problem, the result value in all the search space set will often have many extreme points, it is easy to fall into the local extreme points, that is, the local optimal situation described above, the local search algorithm designed in this paper is to rely on the solution space to search.

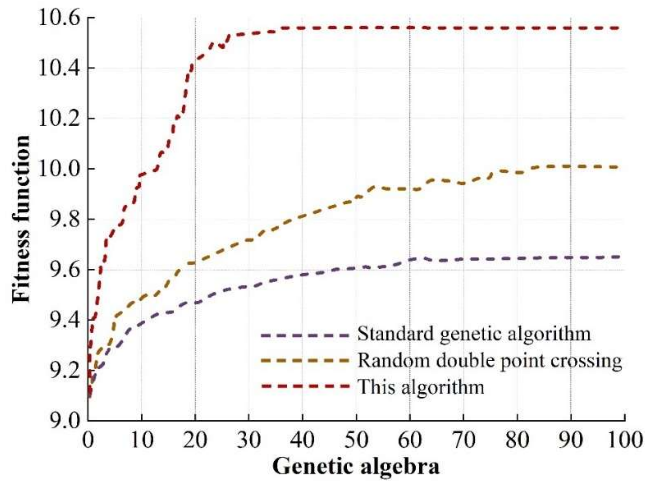

Fig. 4. The fitness function of three genetic algorithms

The experimental results of the objective function considered from the perspective of school resources are shown in Table 3. The fitness function value obtained by the algorithm in this paper is 0.76984, which is much higher than the standard genetic algorithm's 0.58625, and also higher than the randomized two-point crossover method's 0.64618.

Table 3. The fitness function value of the school resource perspective

	Standard genetic algorithm	Random double point crossing	This algorithm
Classroom seating rate	0.58625	0.64618	0.76984

The experimental results of the objective function from the teacher's point of view are shown in Table 4. This paper's algorithm is still optimal in the remaining items, except for the item of “facilitating counselors to listen to lectures”, which is lower than the other two methods. Since college counselors usually take classes that are not of the same major or even the same grade, the objective function of "convenient for counselors to listen to classes" may be contrary to the objective functions such as "classroom seat utilization", "staggered matching of course nature", and "physical education class scheduling".

Table 4. The fitness function of the teacher

	Standard genetic algorithm	Random double point crossing	This algorithm
Empty out at least 1 day	0.60402	0.67047	0.75096
Six sessions a day	0.885	0.91397	0.94835
Special teacher floor arrangement	0.97584	0.96637	0.97914
Facilitate the instructor	0.86443	0.8432	0.8161

The experimental results of the objective function considered from the students' point of view are shown in Table 5.

The fitness function values of this paper's genetic algorithm are all better than the standard genetic algorithm and the improved genetic algorithm, which is precisely due to the improvement of the genetic operator and the design of local search in this paper's algorithm.

Table 5. The fitness function of the conception of the birth Angle

	Standard genetic algorithm	Random double point crossing	This algorithm
The courses are reasonable during the day	0.76699	0.83065	0.84662
For the class, the daily course is arranged evenly	0.72991	0.72359	0.74598
In class, no more than 30 hours per week	0.8886	0.89054	0.98538
Reasonable planning class per week	0.66524	0.82493	0.86975
The course is staggered	0.76763	0.73843	0.77118
Gym time	0.57259	0.6006	0.66312
Classroom change distance	0.81967	0.87022	0.89916
Each layer of teaching leaves an empty classroom	0.69631	0.66752	0.70105

4.2.3. Scheduling Satisfaction Results. In this section, the designed algorithms are compared with the current mainstream algorithms in a side-by-side overall comparison to examine their overall performance.

The comparison data between the test results of this paper's algorithm and five other algorithms (Multi-Objective Evolutionary Algorithm, VEGA, HLGA, NPGA, and NSGA) are given, and various algorithms are implemented in the programming environment to examine the level of teacher satisfaction of several methods, using the average of the results of 60 scheduling, and the specific test results are shown in Table 6.

Comparing the test results, it can be found that in this test problem, the algorithm of this paper has obtained better results in the comparison of user satisfaction level compared to the methods of VEGA, HLGA, NPGA, and NSGA. The average teacher satisfaction is 2.8, which is about 17% higher than the second place NPGA.

Table 6. Teacher satisfaction comparison results

	This method	Multiobjective evolutionary algorithm	VEGA	HLGA	NPGA	NSGA
Aver	2.8	2.1	2.2	1.9	2.4	2.3
Max	3	2.4	3	2.7	2.6	2.8
Min	2.1	1.8	0.8	1.1	2.1	1.2

Figure 5 shows the Pareto optimal solutions generated by the six methods. As can be seen from the figure, in the region where the coordinates from 9100 to 9400 overlap, the Pareto optimal solution generated by this paper's model basically dominates the Pareto optimal solution generated by the multi-objective evolutionary algorithm, indicating that the optimal solution obtained by searching has a certain advantage over the optimal solution obtained by searching the multi-objective evolutionary algorithm.

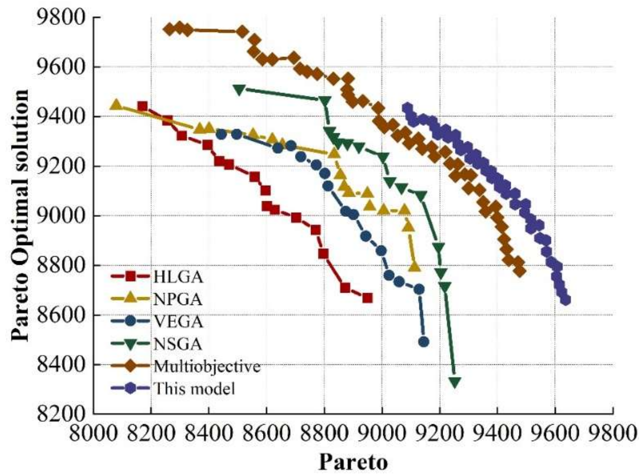

Fig. 5. Performance comparison of various multi-objective algorithms

5. Example application of school scheduling

5.1. Data acquisition

In this paper, the algorithm designed in the previous paper is combined with the actual course information of journalism and communication majors in a university to verify the arithmetic examples. Set the number of teaching weeks w of each semester as 18 weeks, the number of class days d per week as 5 days, and each day is divided into three time periods, i.e., three lectures, with dp=3, and each time period is 4 class hours.

Other relevant input data are course information table and student information table. Among them, the course information table contains <course, total hours, weekly hours, instructor> and so on, and the student information table includes <student, number of courses selected, course selection preference table> and so on. The number of students is set to a fixed value of 250. In the example, the number of courses in the set of courses to be selected has 4 levels, up to 50 courses, and the total number of hours of all courses is 1915, and the average number of hours is 38.3. For the reason of space, the data in the course information table and the student information table of all the examples will not be listed in the article, and a brief summary of the relevant data and information is shown in Table 7.

Table 7. Data statistics for scheduling

Course number	Total period	Mean hour	Mean week	Teacher number	Student population
50	1915	38.3	4.79	30	300
40	1537	38.43	4.8	25	300
30	1155	38.5	4.81	20	300
20	761	38.05	4.76	15	300

The final results obtained and provided include the weekly and section scheduling of the course and the final course selection schedule of each student as well as the course selection satisfaction. The scheduling results are denoted by (course,beginning week,end week,session1,session2), and the final course selection results of student j, i.e., based on the composition of the student's course matrix, and its average course selection satisfaction, which is a single value in each case, are also computed by the method described in the text.

5.2. Comparative analysis

In order to objectively evaluate the degree of satisfaction of this paper's algorithm on each constraint and its effect on improving students' satisfaction with course selection, another course set-based scheduling pattern alignment baseline model is used for model construction, algorithm design, and data quizzing of the varsity scheduling problem in a university. In order to facilitate the comparison of the two models and algorithms, the comparison model used has the same conditions and factors as the study in this paper, except that it does not take into account the students' course selection process.

The scale of the algorithm is 20-11, i.e., the number of courses in the set of courses to be selected m=20, the number of prescribed courses to be selected N=11, and the data in the course information table are shown in Table 8. The comparison of the results of the specific scheduling table obtained by solving is shown in Table 9.

Table 8. Course information sheet

Courses	Total period	Week class	Instructor
1	34	4	12
2	36	4	7
3	36	8	3
4	36	4	13
5	24	4	16
6	24	4	1
7	36	8	10
8	48	8	10
9	48	8	4
10	16	4	7
11	24	8	3
12	48	4	5
13	48	4	11
14	36	8	15
15	36	4	1
16	36	4	5
17	36	4	8
18	40	8	4
19	36	4	11
20	48	4	2

Table 9. Contrast table

Courses	Self-selected course				Curriculum based			
	Initial cycle	Termination cycle	Time 1	Time 2	Initial cycle	Termination cycle	Time 3	Time 4
1	1	12	5	0	1	12	8	0
2	1	16	4	0	1	8	15	0
3	1	16	0	0	1	12	11	0
4	9	16	3	0	9	16	2	0
5	1	16	2	9	1	6	0	12
6	1	8	4	0	9	16	15	0
7	9	16	6	0	9	8	5	0
8	9	12	11	0	1	12	3	0
9	1	8	7	12	9	8	1	10
10	9	16	13	0	9	16	4	0
11	1	12	12	0	9	9	12	0
12	9	16	1	0	1	12	10	0

13	1	12	10	0	1	15	6	0
14	1	8	3	0	1	11	5	0
15	1	16	5	0	1	8	13	0
16	1	12	14	0	1	12	14	0
17	9	16	2	9	9	12	4	9
18	1	12	4	0	1	8	0	0
19	9	16	10	0	9	16	8	0
20	1	8	9	0	1	8	14	0

All the arithmetic examples are measured by comparing the algorithms to get the better solution, in which the average course selection satisfaction results measured by the scheduling algorithms under the two modes are shown in Table 10. Where the example 20-11 indicates that the total number of courses to be selected, m, is 20 and the number of required selections, N, is 11; Mode 1 represents the scheduling mode based on self-selected courses under the semester schedule and Mode 2 represents the scheduling mode based on course sets under the semester schedule.

The data in Mode 1 is the average student selection satisfaction, i.e., the objective function value/number of students, for self-selected course scheduling, and the data in Mode 2 represents the average student selection satisfaction under the course set-based scheduling mode. Since students' course selection is not considered in the comparison algorithm, the calculation of relevant course selection and course selection satisfaction needs to be carried out according to the scheduling results and the students' course selection preference table. After comparing the results, it can be seen that all the algorithms set in this paper have feasible solutions, i.e., they all satisfy all the constraints, and the results of the three-phase algorithm in the self-selected course scheduling mode can obtain a better solution than the course-set-based scheduling algorithm in any of the algorithms.

When the number of courses to be selected is certain, the greater the number of prescribed courses to be selected, the more obvious the scheduling effect of mode 1; and when the number of prescribed courses to be selected is certain, the greater the number of courses to be selected, the scheduling effect of mode 1 will also be more prominent than that of mode 2, but the advantage of the comparison is fading. There are two reasons for this result: mode 1 integrates the course selection process into the scheduling process, fully considers the students' course selection preferences before scheduling operation, considers possible course conflicts in advance, and enables students to select preferred courses to a greater extent after the formation of the class schedule, and thus the more the number of prescribed courses is, the smaller the impact of scheduling results on the satisfaction of course selection is; When the total number of courses to be selected is larger, the result of students' independent selection is more dispersed, the conflict between courses increases, in order to obtain a feasible solution without generating overlapping course schedules, the scheduling process will be adjusted by the backtracking method for students' selection of courses, which sacrifices the satisfaction with course selection to a certain extent, so the comparative advantage tends to diminish.

Table 10. The average selection of the two classes was compared

Numerical example	Mode 1	Mode 2	Increase amplitude
20-3	63	48	31.25%
20-5	79	53	49.06%
20-7	69	47	46.81%
20-9	75	50	50.00%
20-11	70	42	66.67%
40-3	65	49	32.65%
40-5	78	44	77.27%

40-7	62	50	24.00%
40-9	77	59	30.51%
40-11	75	58	29.31%
30-3	70	56	25.00%
30-5	65	55	18.18%
30-7	66	48	37.50%
30-9	74	51	45.10%
30-11	64	42	52.38%
50-3	65	51	27.45%
50-5	75	49	53.06%
50-7	71	41	73.17%
50-9	79	51	54.90%
50-11	67	56	19.64%

6. Conclusion

In this paper, for the problem of unreasonable allocation of teaching resources in school affairs scheduling, the school affairs scheduling decision-making model is designed based on the improved genetic algorithm. Experiments such as coding, conflict detection and local search operator are carried out on the dataset, and it is found that scheduling conflicts are inevitably generated after initializing the population, while the conflict elimination algorithm designed in this paper is able to eliminate scheduling conflicts, and the local search operator of this paper's algorithm helps to increase the fitness value of chromosomes. In the overall performance analysis, in the region where the coordinates from 9100 to 9400 overlap, the Pareto optimal solution generated by this paper's algorithm outperforms the rest of the genetic algorithm combinations, and has a certain advantage over the optimal solution obtained by the multi-objective evolutionary algorithm search. Practical application of the model is able to generate class schedules that satisfy schools, teachers and students.

References

[1] Xie, Y., Wang, H., Wang, K., Ge, X., & Ying, X. (2024). The Application and Potential of Multi-Objective Optimization Algorithms in Decision-Making for LID Facilities Layout. Water Resources Management, 1-15.

[2] Bai, J. (2024). Mathematical Modeling and Multi-Objective Optimization Methods for Big Data in Higher Education. Frontiers in Educational Research, 7(7).

[3] Cheng, L., Wang, Y., Hu, B., & Maia, D. (2022). Allocation of English Distance Teaching Resources based on Deep Reinforcement Learning and Multi - Objective Optimization. Mathematical Problems in Engineering, 2022(1), 3729263.

[4] Bastian, N. D., Ekin, T., Kang, H., Griffin, P. M., Fulton, L. V., & Grannan, B. C. (2017). Stochastic multi-objective auto-optimization for resource allocation decision-making in fixed-input health systems. Health care management science, 20, 246-264.

[5] Lotfi, R., Pilehforooshha, P., & Karimi, M. (2023). A multi-objective optimization model for school location-allocation coupling demographic changes. Journal of spatial science, 68(2), 225-244.

[6] Tanantong, T., Pannakkong, W., & Chemkomnerd, N. (2022). Resource management framework using simulation modeling and multi-objective optimization: a case study of a front-end department of a public hospital in Thailand. BMC Medical Informatics and Decision Making, 22(1), 10.

[7] Tang, Z., & Sun, J. (2018). Multi objective optimization of railway emergency rescue resource allocation and decision. International Journal of System Assurance Engineering and Management, 9(3), 696-702.

[8] Feng, Y. Y., Wu, I. C., & Chen, T. L. (2017). Stochastic resource allocation in emergency departments with a multi-objective simulation optimization algorithm. Health care management science, 20, 55-75.

[9] Cho, J. H., Wang, Y., Chen, R., Chan, K. S., & Swami, A. (2017). A survey on modeling and optimizing multi-objective systems. IEEE Communications Surveys & Tutorials, 19(3), 1867-1901.

[10] Dang, Q. L., Xu, W., & Yuan, Y. F. (2022). A dynamic resource allocation strategy with reinforcement learning for multimodal multi-objective optimization. Machine Intelligence Research, 19(2), 138-152.

[11] Kumar, S., Tejani, G. G., Pholdee, N., Bureerat, S., & Jangir, P. (2022). Multi-objective teaching-learning-based optimization for structure optimization. Smart Science, 10(1), 56-67.

[12] Sha, Y., Wang, H., Wang, D., & Ghobaei-Arani, M. (2024). A multi-objective QoS-aware IoT service placement mechanism using Teaching Learning-Based Optimization in the fog computing environment. Neural Computing and Applications, 36(7), 3415-3432.

[13] Liu, C., Ma, Y., Yin, H., & Yu, L. (2021). Human resource allocation for multiple scientific research projects via improved pigeon-inspired optimization algorithm. Science China Technological Sciences, 64(1), 139-147.

[14] Zheng, J. H., Kou, Y. N., Jing, Z. X., & Wu, Q. H. (2019). Towards many-objective optimization: objective analysis, multi-objective optimization and decision-making. IEEE Access, 7, 93742-93751.

[15] Li, Q. (2024). Empowering financial management in educational institutions: A multi-objective decision-making system using intelligent fuzzy logic algorithm and digital marketing. Computer-Aided Design & Applications, 21, 198-210.

[16] Zhang, Y. T., Yang, J. Y., & Wu, Y. (2023). Dynamic Resource Allocation Strategy of Multi-objective Fuzzy Optimization Based on Markov Decision Process. IEEE Access.

[17] Guo, S. (2022). Research on Multi - Objective Intelligent Optimization of Financial Resource Management and Allocation Methods in Colleges and Universities. Mathematical Problems in Engineering, 2022(1), 1906031.

[18] Moazeni, A., Khorsand, R., & Ramezanpour, M. (2023). Dynamic resource allocation using an adaptive multi-objective teaching-learning based optimization algorithm in cloud. IEEE Access, 11, 23407-23419.

[19] Zhang, W., Zhang, S., Guo, S., Yang, Y., & Chen, Y. (2017). Concurrent optimal allocation of distributed manufacturing resources using extended teaching-learning-based optimization. International Journal of Production Research, 55(3), 718-735.

[20] Kassem, A. A., Kumar, S., Naik, A. V., Bhagavanthu, M., & Reddy, R. A. (2024, March). Resource Allocation in Multi-Access Edge Computing using Teaching-Learning Based Optimization: A Multi-Objective Approach. In 2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT) (pp. 1-5). IEEE.

[21] Ali Roozbeh Nia,Mohammad Hemmati Far & Seyed Taghi Akhavan Niaki. (2015). A hybrid genetic and imperialist competitive algorithm for green vendor managed inventory of multi-item multi-constraint EOQ model under shortage. Applied Soft Computing Journal353-364.

[22] Hongbing Ding,Guangchen Zhang,Shiwei Wang,Yu Zhang,Yan Yang & Chuang Wen. (2025). Multi-objective optimization of supersonic separator for gas removal and carbon capture using three-field two-phase flow model and non-dominated sorting Genetic Algorithm-II (NSGA-II). Separation and Purification Technology(PB),130363-130363.

[23] Kai Wang,Hong Zhao,Yanan Ding,Tao Li,Lujian Hou & Fengjuan Sun. (2015). Optimization of air pollutant monitoring stations with constraints using genetic algorithm. Journal of High Speed Networks(2),141-153.