

A Study on Optimizing Physical Education Curriculum Scheduling Using Data Flow Analysis Algorithm

Yan Shang*

* Faculty of Physical Education, Baotou Teachers' College, Inner Mongolia University of Science and Technology, Baotou, Inner Mongolia, 014030, China; shangyan2023@126.com

Abstract: In order to improve the speed and accuracy of large and complex information processing, the use of data flow analysis technology and to replace the traditional manual scheduling is an inevitable requirement for the quality improvement of higher education. Based on the basic concept of data flow analysis, this paper proposes the use of data flow analysis in different situations. Considering the actual demand of constraints, the scheduling problem is divided into hard and soft constraints, and a multi-objective optimization constraints mathematical model is constructed for physical education course scheduling. Using genetic algorithm, the sports course scheduling problem is optimized and solved. Test the performance of the algorithm in the actual operation and the use of the effect, the model in the completion of the iteration score of 97.058, the whole process takes no more than 2.5 s. Before and after the optimization of scheduling program students change classroom per capita time consumption of 10.04 min and 6.93 min, respectively, the passage of the time consumption reduced by 30.98%, the algorithm used in this paper effectively reduces the passage of time, the physical education course schedule The algorithm used in this paper effectively reduces the passage time and plays an optimizing role in the schedule of physical education courses.

Keywords: data flow analysis; constraints; genetic algorithm; optimization solution; physical education scheduling program

1. Introduction

Due to the influence of traditional education concepts and the aftermath of the exam-based education system, most of the current physical education teaching in Chinese schools is formal and only focuses on knowledge and skill transfer [1-2]. With the new curriculum reform, the depth of the concept of quality education and the planning and construction of modern cities, such as the development of diseases at an early age, schools are directly required to pay attention to the physical quality of the students [3-5], starting from the enhancement of physical exercise, the design of a reasonable physical education curriculum, to carry out the optimal arrangements for physical education, and to promote the love of physical education and sports among students in colleges and universities, as well as to form a good habit of lifelong physical education and sports [6-8]. Therefore, redesigning and optimizing the physical education curriculum has become an urgent task.

Physical education curriculum arrangement is mainly based on college physical education teaching, facing the influence of the exciting network, the arrangement of physical education curriculum must establish a correct, scientific and effective ideological guidance [9-10]. Its ability to adopt a scientific and effective design scheme for network development and implementation in accordance with the actual needs of college students and the actual development of colleges and universities, due to the rapid development of society, the actual teaching effect brought about by the design of the physical education curriculum can not be optimized [11-15]. And with the development of artificial intelligence, the application of intelligent algorithms plays an important role in the optimization of physical education curriculum arrangement [16].



As a branch of artificial intelligence, data flow analysis algorithm is a computational method to deal with dynamic data flow [17]. In the digital era, school data are everywhere, student user behavior data, to the massive data generated by teaching management and so on, which constitute a complex data network [18-19]. And this lays the foundation for the data flow analysis algorithm to optimize the curriculum arrangement of physical education teaching, which can help teachers find the direction of data flow, the law and the secrets hidden in it, so as to reasonably arrange the physical education teaching curriculum according to the law of data combined with the school's curriculum design [20-24].

Physical education is crucial for students' growth, but currently physical education faces many challenges, among which the problems of unreasonable curriculum arrangement, low resource input, and low treatment of teachers have seriously hindered the development of physical education. Literature [25] analyzed the planning and organizing guidelines affecting the arrangement of physical education courses in schools, and based on structured interviews, it was shown that the current arrangement of physical education courses in schools is unreasonable, and more reasonable and good management and decision-making should be introduced in the arrangement of physical education courses. Literature [26] identified various deficiencies pointed out in physical education and traditional professional development in physical education, such as inadequate staffing of teachers, resulting in a lack of confidence and competence of teachers and prompting a call for more effective methods to improve teacher competence and quality of physical education teaching. Literature [27] points out that students' academic performance essentially stems from the subject content of the school, but the positive impact of physical activity in improving students' academic performance has also been confirmed by its role in the development of students' brain networks and the formation of adaptive mental functions. Literature [28] assessed the resources of representative public elementary schools for their possible impact on physical education and emphasized that financial support helps schools to better develop physical education. Literature [29] examined the difficulties and expectations faced by physical education teachers in public schools, raising a questionnaire to teachers pointed out that physical education teachers face low salaries, unstable infrastructure and lack of equipment, which reflects the lack of emphasis on physical education in schools. Literature [30] aimed to find out the views and suggestions of physical education teachers on the problems in teaching practice, raising the analysis of interviews showed that the biggest problem is that schools do not pay attention to physical education classes and some parents and students are not interested in physical education classes.

In order to promote the good development of physical education, a reasonable curriculum arrangement is crucial, and the development and application of artificial intelligence promotes the rationalization of curriculum arrangement, and is not limited to physical education, but plays an optimizing role in the curriculum arrangement of other professional courses. Literature [31] emphasized the importance of school physical education curriculum scheduling to improve public health, and developed a weighted fitness function to assess health fitness scores, and used algorithms such as genetic algorithms and particle swarm optimization for use in the above function, aiming to optimize physical activity schedules, and the results of the study showed that the physical activity schedules played a role in reducing the burden of disease, and improving the cognition of mental health. Literature [32] pointed out the inadequacy of traditional methods for optimizing physical education schedules and proposed a deep learning approach for optimizing physical exercise schedules, and improved experiments revealed that the deep learning approach effectively optimized physical education schedules and helped to improve students' physical fitness. Literature [33] improved the genetic algorithm by using the method of co-evolution in order to solve the problem of course scheduling in universities, and improved simulation experiments verified that the improved genetic algorithm is better than the traditional genetic algorithm, and it is an efficient course scheduling method. Literature [34] emphasized that undergraduate course scheduling has a significant impact on students' academic performance and satisfaction, based on which an automated scheduling system incorporating models such as sequence-to-sequence was proposed, which can simplify the course scheduling process and improve academic planning. Literature [35] proposed a course scheduling method based on the Improved Binary Cuckoo Search (IBCS) algorithm and improved simulation experiments revealed that the IBC algorithm is capable of achieving global scheduling schemes with higher stability. Literature [36] examined the application of genetic algorithms in course scheduling optimization in educational institutions and emphasized that genetic algorithms, especially when combined with hybrid techniques, provide reliable solutions to complex course scheduling problems. [37] discussed the variational genetic algorithm and introduced it into a course scheduling system for students, which was shown to perform well in the course scheduling system by testing the speed and accuracy of the system, which provided a strong support for the course scheduling process.

This paper analyzes the basic concepts of data flow analysis, refines the key operational steps to realize the basic operations of data flow, and proposes different data flow analysis solution methods

according to different processing situations. The scheduling problem is regarded as a multi-objective optimization problem, and the objective function, constraints, decision variables and other factors are set to construct a constrained mathematical model of the scheduling problem, which is optimized and solved by using genetic algorithm. Based on the genetic algorithm, design the data flow, coding scheme, fitness function and other aspects of the scheduling problem. Apply simulation test to examine the model's physical education course scheduling effect, and the experiment shows that the model demand sensitivity is better, and the scheduling quality is improved.

2. Data flow analysis techniques

2.1. Basic Concepts in Data Flow Analysis

Data flow analysis is a set of algorithms that collect information about the moment of program execution through static analysis [38]. It can collect the semantic information of a program from the program code, and mine and derive the target program information through algebraic methods. Using the results of data flow analysis, problems in compiler optimization, program verification, debugging, testing, parallelism, vectorization and parallel programming environments can be solved.

2.1.1. Basic blocks and control flow

The so-called basic block is a sequence of statements in a program that is executed in a certain order, with one and only one entry and one exit. The entrance is the first statement; the exit is the last statement. For a basic block, in the execution, the control flow can only enter from its entrance, and must exit from the exit, there can be no interruptions or branches (except for the exit statement). Simply put, the basic block is a single entry and exit area, that is, as long as the first statement in the basic block is executed, then each of the remaining statements will be executed in order.

2.1.2. Aggregation operations

Let's first define the two types of sets that are of primary interest for data flow analysis: the *in* and the *out* sets. These two sets represent the set of data flow information at the inlet and outlet of a basic block, respectively. The data flow analysis is based on the difference between the information contained in these two sets before and after the execution of the basic block. These two sets are interrelated and not completely independent.

The aggregation operation is an operator in the functional relationship between the in set and the out set. Assuming that the data flow information flow direction is the same as the control flow direction, the aggregation operation is “ \bigcap ” (intersection) under the premise of full path data flow analysis. The *in* set $in[i]$ of the basic block i and the *out* set of the antecedent basic blocks j_1, j_2, \dots, j_m of i have the following relationship:

$$in[i] = \bigcap_{i=1}^m out[j_i] \quad (1)$$

This formula denotes the intersection “ \bigcap ” operation of the set of i properties in the set of *in* derived from the set of *out* of its precursor basic blocks j_1, j_2, \dots, j_m .

2.1.3. Data flow equations

When performing data flow analysis, it is not only necessary to know how the data flow information is propagated and used between basic blocks, but also how it is propagated and used within basic blocks.

For a data flow analysis problem where the direction of data flow information flow is consistent with the direction of control flow for a certain basic block i , $in[i]$ denotes the data flow information at the inlet of i , and $out[i]$ denotes the data flow information of the problem at the outlet of i . During the execution of the program, new data flow information and cancellation of part of the data flow information may be generated within i , which are denoted as $gen[i]$ and $kill[i]$, respectively. Then these sets are related to each other as follows:

$$out[i] = gen[i] \cup (in[i] - kill[i]) \quad (2)$$

Data flow equations and aggregation operations both are basic operations used to compute data flow problems. By traversing the control flow graph, these two operations give the in , out set for each statement, basic block.

2.2. Methodology for the use of data flow analysis

2.2.1. Syntactic Guidance Solving Methods for Structured Programs

The syntax guidance method of structured programs is a relatively simple, direct and easy-to-understand data flow analysis method that deals with different situations according to the structural characteristics of branching, looping and jumping statements in the syntax. Figure 1 shows three typical structures in the program structure, which includes three common control structures: sequential statement, branching statement, and looping statement.

To facilitate the explanation, the information in the figure is abstracted, where the big circle is the basic block, the small circle is the dummy block without statements, and the arrow indicates the control flow diagram. And, it is assumed that this data flow analysis method is full path problem and forward flow problem. Each of these three regions is analyzed in the following, giving their syntactic guidance definitions for $in[R]$, $out[R]$, $gen[R]$, and $kill[R]$.

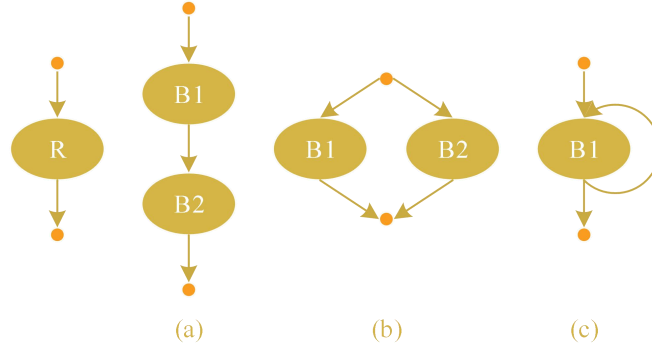


Figure 1. Three typical structures in the program structure

First analyze the case in Figure 1(a), where (a) is a sequentially executed statement and the region R consists of the basic blocks $B1$ and $B2$ in series. For this case, the initial set of data flow information in will pass sequentially through $B1$, $B2$, whose properties are successively changed by $B1$, $B2$, and finally reach the end point. So it can be obtained:

$$gen[R] = gen[B2] \cap (gen[B1] - kill[B2]) \quad (3)$$

$$kill[R] = kill[B2] \cup (kill[B1] - gen[B2]) \quad (4)$$

For the branching structure shown in Fig. 1(b), the nature of the data flow in any branch reaches the end point of the region, so there:

$$gen[R] = gen[B1] \cap gen[B2] \quad (5)$$

$$kill[R] = kill[B1] \cup kill[B2] \quad (6)$$

Finally, consider the region of loops as shown in Figure (c). Intuitively, loops do not affect gen and $kill$. So there is:

$$gen[R] = gen[B1] \quad (7)$$

$$kill[R] = kill[B1] \quad (8)$$

Now that the *gen* and *kill* sets of the three control structures shown in Fig. 1 have been obtained, the *in* and *out* sets are computed below.

For the case where the basic blocks in Fig. 1(a) are executed sequentially, it is clear that there are:

$$in[B1] = in[R] \quad (9)$$

$$out[B1] = in[B2] \quad (10)$$

$$out[B2] = out[R] \quad (11)$$

For the branching structure in Fig. (b), it is assumed that the control flow will flow through any branch:

$$in[B1] = in[B2] = in[R] \quad (12)$$

$$out[R] = out[B1] \cap out[B2] \quad (13)$$

Similarly, for the loop structure in Fig. (c), it can be obtained by analyzing and calculating:

$$in[B1] = in[R] \cap gen[B1] \quad (14)$$

$$out[R] = out[B1] \quad (15)$$

At this point, we have obtained the syntactic guidance definitions of $in[R], out[R], gen[R]$ and $kill[R]$ for all three control structures R . As long as there is a Gen, Kill for each basic statement, it is possible to decompose a structured program step-by-step according to this rule and perform a data flow analysis, and ultimately find the set of *out* corresponding to each basic block by using the set of *in*'s of the basic block at the beginning of the program.

2.2.2. Iterative Solution Algorithm for the Full Path Forward Flow Data Flow Problem

Instead of computing the set of *in*, *out* through syntactic constraints, the iterative solution algorithm is an algorithm that realizes the data flow analysis by iteratively computing the set of *in* and *out* for each node on a control flow graph, and so it is called the iterative solution algorithm [39].

For each node N in the control flow graph, $in[N]$, $out[N]$, $kill[N]$, and $gen[N]$ can be defined as in the previous method. If the set of *gen*, *kill* for each control-flow graph node has been computed, a set of equations can be established: (P is the control-flow graph antecedent node of N and it is a full-path forward flow problem)

$$in[N] = \bigcap out[P] \quad (16)$$

$$out[N] = gen[N] \cap (in[N] - kill[N]) \quad (17)$$

Since the set of $out[N]$ for any node N in the control-flow graph does not decrease, and since if *out* does not change, then neither does *in* for the next iteration, and if *in* stays the same, then neither does *out* for the next iteration, and all subsequent iterations do not change any more, the algorithm will always stop eventually.

3. Physical education curriculum organization issues and related technologies

3.1. Scheduling problems

Scheduling is the process of rationalizing classes, courses, teachers, classrooms and time to avoid conflicts while trying to meet the demand for the purpose of making teaching and learning activities can be carried out smoothly. Therefore its solution space is a 5-dimensional vector. As for the constraints, we need to consider the actual demand, such as: whether the class time is reasonable, the priority of class time, etc. Thus, it can be seen that scheduling is a multi-objective, multi-constraint, non-linear combinatorial optimization problem.

In order to schedule a good class schedule, some constraints need to be formulated in class scheduling, which are divided into: hard constraints and soft constraints. Among them, hard constraints must be satisfied and soft constraints are satisfied as much as possible. Hard constraints are usually to avoid conflicts, and soft constraints are usually to optimize the class schedule scheme. Common constraints are listed below.

The hard constraints are as follows:

- (1) A class may have only one lesson in session at any one time
- (2) A teacher can only have one class in session at any one time
- (3) A classroom can only have one class at a time
- (4) The number of classrooms multiplied by the number of time slots must result in a total number of time slots that is greater than the number of combinations (classes, lessons, teachers).

The soft constraints are as follows:

- (1) Try not to schedule consecutive classes in the same class.
- (2) Minimize the number of occupied classrooms and maximize classroom utilization.
- (3) Minimize evening scheduling
- (4) It is better to schedule the classes during the time when the participants' minds are active.

Under the premise of meeting the hard constraints, a good scheduling program should try to meet the soft constraints, in order to arrange a scientific and reasonable, to meet the needs of most teachers and students.

3.1.1. Mathematical model of the scheduling problem

It has been shown above that scheduling is a multi-objective optimization problem, here let the objective function be t , the constraints be m , and the decision variables be n . The mathematical model of the scheduling problem is as follows:

$$\max y = f(f_1(x), f_2(x), \dots, f_t(x)) \quad (18)$$

$$\text{s.t. } e(x) = (e_1(x), e_2(x), \dots, e_m(x)) \quad (19)$$

Where: $x = \{x_1, x_2, \dots, x_n\} \in X, y = \{y_1, y_2, \dots, y_t\} \in Y$

Eq. (18) is the objective function of the scheduling problem and Eq. (19) is the constraints of the scheduling problem. Where x is the decision vector, y is the objective vector, X is the decision space formed by x , and Y is the objective space formed by y . The multi-objective optimization of the scheduling problem is to find the maximum value of y .

The essence of the scheduling problem is to talk about rationalizing classrooms and schedules to classrooms and classes and avoiding conflicts between these five factors at the same time. Here, the class is denoted by C , the classroom is denoted by R , the course is denoted by L , the teacher is denoted by T , and the schedule is denoted by S . Then five sets are used to represent the value space of these five dimensions: class set $C = \{C_1, C_2, C_3 \dots C_i\}$, classroom set $R = \{R_1, R_2, R_3 \dots R_j\}$, course set $L = \{L_1, L_2, L_3 \dots L_k\}$, teacher set $T = \{T_1, T_2, T_3 \dots T_m\}$, schedule set $S = \{S_1, S_2, S_3 \dots S_n\}$.

3.1.2. Mathematical modeling of hard constraints for scheduling problems

Based on the above definition, the hard constraints for scheduling classes can be expressed using the following:

- (1) A class can have only one class in session at any one time

$$\sum_{j=1}^J \sum_{k=1}^K \sum_{m=1}^M xC_i R_j L_k T_m S_n \leq 1 \quad (20)$$

The expression: $i = 1, 2, \dots, I; n = 1, 2, \dots, N; xC_i R_j L_k T_m S_n = \begin{cases} 1, & \text{Satisfaction of conditions} \\ 0, & \text{Failure to meet conditions} \end{cases}$

(2) A teacher can schedule only one class at a time

$$\sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K xC_i R_j L_k T_m S_n \leq 1 \quad (21)$$

In the expression: $m = 1, 2, \dots, M; n = 1, 2, \dots, N; xC_i R_j L_k T_m S_n = \begin{cases} 1, & \text{Satisfaction of conditions} \\ 0, & \text{Failure to meet conditions} \end{cases}$

(3) A classroom can only have one lesson scheduled at the same time

$$\sum_{i=1}^I \sum_{k=1}^K \sum_{m=1}^M xC_i R_j L_k T_m S_n \leq 1 \quad (22)$$

In the expression: $j = 1, 2, \dots, J; n = 1, 2, \dots, N; xC_i R_j L_k T_m S_n = \begin{cases} 1, & \text{Satisfaction of conditions} \\ 0, & \text{Failure to meet conditions} \end{cases}$

(4) The total number of time periods obtained by multiplying the number of classrooms by the number of time periods must be greater than the number of combinations of (class, course, teacher)

$$\sum_{j=1}^J \sum_{n=1}^N 1 \geq \sum_{i=1}^I \sum_{k=1}^K \sum_{t=1}^T 1 \quad (23)$$

In summary, equations (20), (21), and (22) are hard constraints that must be observed in the scheduling problem, and

Eq. (23) is a necessary condition for scheduling to have a solution. The hard constraint $e(x)$ of the scheduling problem is thus obtained, and the conflict has to be eliminated by making $e(x) = 0$ on the actual scheduling.

3.1.3. Mathematical modeling of soft constraints for the scheduling problem

The soft constraints of the scheduling problem can also be formulated as a mathematical model:

(1) The same course in the same class should not be scheduled consecutively as far as possible. Taking into account the acceptance of knowledge and the human memory curve, the interval between the same class should be neither too long nor too short, in order to improve the learning efficiency of the trainees. Therefore, the following model of trainees' acceptance ability is designed:

$$f_1 = G = \sum \Delta d(T_m) \delta \quad (24)$$

Where δ is the interval coefficient and $\Delta d(T_m)$ is the difference in the number of sections of the same class for the same course. The larger the value of G the better the acceptance of the participants.

(2) Minimize the number of occupied classrooms and maximize the usage of occupied classrooms. Scheduling should try to use the least number of classrooms and maximize the use of classrooms, the released classrooms can be used for other arrangements to improve the efficiency of classroom use. Therefore the following classroom utilization model is designed:

$$f_2 = U = N(\text{Classroom usage}) / N(\text{Number of classrooms occupied}) \quad (25)$$

The larger the U value the higher the classroom utilization.

(3) Minimize evening scheduling. Scheduling should minimize evening classes, which are usually self-study time and tutorial Q&A time. Therefore the self-study model is designed as follows:

$$f_3 = Q = S(\text{Number of hours of daytime classes}) / S(\text{Total number of hours}) \quad (26)$$

The larger the value of Q , the more the sessions will be concentrated during the day.

(4) Session Optimality Expectation

The session optimization expectation reflects the efficiency of the learners for the current time of the lesson, and it is better to arrange the lesson in the time period when the learners' minds are active. The active thinking time period is concentrated in the morning, when the state of the students is more concentrated, learning efficiency, while the afternoon will be rested, the brain is more likely to be sleepy, the students state is not good, thinking is easy to be scattered, the learning efficiency will be discounted. So the section optimization degree is defined as:

$$f_4 = M = \sum E_i / 100n \quad (27)$$

n is the number of classes, E is the optimization degree of each class, and $i = 1, 2, 3 \dots C, C$ is the number of courses per class.

The larger the value of M , the more reasonable the class scheduling.

This yields the objective function of the class scheduling problem: $\max y = f(f_1, f_2, f_3, f_4)$. The optimization of the scheduling problem can be quantified through functionalization by searching the solution space for the optimal solution that maximizes the objective function.

3.2. Genetic algorithms

For the iterative problem of solving the data flow problem of physical education courses, an optimal solution is proposed to rearrange the physical education courses to get an optimal solution to solve the physical education course scheduling problem.

3.2.1. Basic Ideas of Genetic Algorithms

Genetic algorithms start with the establishment of a population of the set of possible potential solutions to the problem, which consists of a certain number of individuals that have been genetically encoded [40]. Each individual is an entity that expresses a trait of genetic information on a chromosome. Chromosomes, as the main carriers of genetic material, determine the external expression of individual traits.

Genetic algorithms use natural evolutionary models such as selection, crossover, mutation, migration, localization and neighborhood. At the beginning of the computation, a certain number of individuals (parent 1, parent 2, parent 3, parent 4, ... parent N) are randomly initialized to form an initial population, and the fitness function value is calculated for each individual, and the first generation (the initial generation) is generated. If the termination condition of the genetic algorithm is not reached in the first generation, the calculation of generating the next generation starts. In order to produce the next generation, individuals are selected according to the fitness value and the parent generation requires genetic recombination (crossover) and produces offspring, all of which mutate with a certain probability. Then the fitness values of the offspring are recalculated again and the offspring are inserted into the population replacing the parent generation to form a new generation (child 1, child 2, child 3, child 4, ...). This process is performed cyclically until the termination condition is satisfied.

The basic genetic algorithm can be represented by equation (28):

$$SGA = (C, E, P_o, M, \phi, \zeta, \psi, T) \quad (28)$$

In Eq. (28), C denotes the coding method of the individual, which is the basic element in the genetic algorithm, and a combination formed by the connection of genes and chromosomes is a kind of solution of the course orchestration. E denotes the fitness evaluation function of an individual, by which the value of the function distinguishes whether the individual meets the requirements or not. P_o denotes the initial population, which is formed randomly by the genetic algorithm, and the subsequent operation process of the genetic algorithm will be centered around the initial population. M denotes the population size, which indicates the number of individuals in the population. ϕ denotes the selection operator, filtering the calculation by the size of the fitness value, usually the larger the fitness value of the individual is more likely to be selected. ζ denotes the crossover operator, where the offspring individuals are generated by random genetic exchange of the parental individuals, and there

are various ways of crossover, including single-point crossover and uniform crossover. ψ denotes the variation operator, where the offspring individuals randomly change their genes to generate random genes different from those of their parents. T denotes the termination condition of the algorithm, when performing genetic iteration, the number of genetic algorithm generations is usually set in advance, when the number of iteration generations reaches the number of genetic algorithm generations, the operation terminates.

3.2.2. Adaptation function

Genetic algorithms in the evolutionary process are based on the fitness value of the individual as the basis for the superiority and inferiority. The good or bad setting of the fitness function directly affects the convergence speed of the genetic algorithm and the degree of superiority or inferiority of the solution. The design idea of the fitness function is to weight and sum the types of conflicts existing in each chromosome, where the weight value w_i represents the importance degree of the i th rule, and p_i is the penalty coefficient of the i th rule. If a chromosome violates a rule i , its penalty value is $w_i * p_i$. The conflicts present in a chromosome are weighted and summed and added to 1 and then their inverse is found as shown in equation (29) below:

$$f = \frac{1}{1 + \sum_{i=1}^k w_i * p_i} \quad (29)$$

The fitness function, also known as the object function, is used by genetic algorithms to relate to the specific problem to be solved through the fitness function. The fitness value represents the degree of adaptation of each individual in the population to its environment, and there is a correspondence with the objective value of the optimization problem. In essence, it is the process of abstracting a specific problem into an optimization mathematical model. For the same problem, different fitness functions can be designed. In the actual use of genetic algorithms, the design of an appropriate fitness function is a very critical issue. A good fitness function can truly reflect the optimization situation and find the real optimal solution of the problem, while a poor quality fitness function may make the optimized solution unusable. When designing the fitness function, it is usually necessary to consider its rationality, consistency, computational amount, generality, etc.

3.2.3. Initial populations

Random generation is used to generate the initial population, but the population will also produce some individuals that do not satisfy the hard constraints of the scheduling model. In this paper, the method is to screen the individuals in the initial population one by one, discard the individuals that cannot meet the hard constraints, and regenerate new individuals. The operation is repeated so that the individuals in the initial population satisfy the hard constraints as much as possible.

3.2.4. Selection operations

The frequently used method for selection operation is the roulette selection algorithm, which selects individuals with high fitness values from the previous generation population and puts them into the pairing pool, eliminates the individuals with low fitness values and retains the individuals with high fitness values to form a new population, so as to carry out the next step of crossover and mutation operation. In this paper, the roulette selection algorithm is improved on the basis of using the roulette selection algorithm, and the probability of selecting an individual to enter the next generation population is:

$$p_i = f_i / \sum_{i=1}^M f_i \quad (30)$$

where f_i is the individual fitness value and M is the population size. When each time after generating N random numbers, and then select an individual from them, not generate a random number to select an individual, so that the cycle of M times will produce M individuals, which can ensure that each time the highest fitness value of the individual can be selected, to improve the next

generation of the population of individuals more optimal.

3.2.5. Crossover and mutation operations

Crossover operation is the use of crossover operator to get a new generation of individuals, the two individuals within the population according to the crossover probability randomly exchanging some of their genes, generating a new combination of genes, constituting a new individual. Through crossover operation, the search ability of the algorithm can be improved, and it is the main method to generate new individuals, usually there are single-point crossover, multi-point crossover and other methods. Mutation operation can make the local search range increase, improve the algorithm local search ability, based on the mutation probability, some genomes in the individual are replaced to form a new individual, it is the auxiliary method of generating new individuals, and maintains the diversity of the population. The value of the crossover probability p_c determines the spatial search area of the population, but if the value of p_c is taken too large, it will destroy the good characteristics of individuals. If the value is taken too small, it can also cause premature convergence of the optimal solution. The value of the variation probability p_m maintains the diversity of the population, but too large a value of p_m will turn the algorithm into a random search algorithm, and too small a value will avoid the loss of some individual gene. In this paper, we introduce an improved adaptive regulation idea to make the crossover probability and mutation probability update with the iteration situation and the value of individual fitness function for self-adjustment. Based on the variation of p_c , when p_c increases, p_m decreases, and when p_c decreases, p_m increases, which allows the crossover and mutation operations to jointly ensure the global search capability of the genetic algorithm.

The crossover probability is:

$$p_c = \begin{cases} k_1 - \frac{(k_1 - k_2)(f' - f_{avg})}{f_{max} - f_{avg}}, & f' \geq f_{avg} \\ k_1, & f' < f_{avg} \end{cases} \quad (31)$$

The probability of mutation is:

$$p_m = \begin{cases} k_3 - \frac{(k_3 - k_4)(f - f_{avg})}{f_{max} - f_{avg}}, & f \geq f_{avg} \\ k_3, & f < f_{avg} \end{cases} \quad (32)$$

where f_{max} is the fitness value of the largest individual in the population, f_{avg} is the average fitness value of the individuals in the population, f' is the fitness value of the larger of the two individuals to be crossed, f is the fitness value of the variant individual, $0 < k_1, k_2, k_3, k_4 \leq 1$, all are constants.

For the crossover operation, this paper adopts a hybrid hybridization operator approach, which utilizes the crossover probability p_c to select two parent individuals in the population $x = (x_1, x_2, \dots, x_m) \in [L, U], y = (y_1, y_2, \dots, y_n) \in [L, U]$, two hybrid points i_1 and i_2 are subjected to two-point convex hybridization, and two offspring individuals are obtained, respectively:

$$x' = (x_1, \dots, x_{i_1-1}, x_{i_1}', y_{i_1+1}, \dots, y_{i_2-1}, x_{i_2}', x_{i_2+1}, \dots, x_m) \quad (33)$$

$$y' = (y_1, \dots, y_{i_1-1}, y_{i_1}', x_{i_1+1}, \dots, x_{i_2-1}, y_{i_2}', y_{i_2+1}, \dots, y_m) \quad (34)$$

Among them:

$$x_{i_1}' = ax_{i_1} + (1-a)y_{i_1}, x_{i_2}' = ax_{i_2} + (1-b)y_{i_2} \quad (35)$$

$$y_{i_1}' = ay_{i_1} + (1-a)x_{i_1}, y_{i_2}' = ay_{i_2} + (1-b)x_{i_2} \quad (36)$$

where $[L, U] = \{(x_1, x_2, \dots, x_m) \mid l_i \leq x_i \leq u_i, i = 1, 2, \dots, m\}$ denotes the space of feasible solutions, a and b are random numbers in $[0, 1]$, and the operation is repeated to obtain the set of offspring. The method of offspring individuals obtained by crossover operation can improve the evolutionary speed in the pre-evolutionary stage compared to the traditional genetic algorithm [41]. The mutation operation, according to the mutation probability, randomly selects two columns in the scheduling scheme and exchanges them to determine whether they satisfy the constraints, and if there is a conflict, randomly selects and exchanges them again until there is no conflict, producing a new individual.

3.3. Design of scheduling problem based on genetic algorithm

3.3.1. Data sheet on scheduling issues

The scheduling process begins with the teaching plan, i.e., the academic department associated with the scheduling process collects information about the start of classes for the next semester in each teaching unit before the end of each semester. In this paper, we use a data flow analysis algorithm to collect and initially solve and analyze the data related to physical education teaching courses.

This information mainly includes the starting class (in this case, teaching class), class size, course subject, weekly credit hours, total credit hours, course nature, instructor, classroom type, and prerequisite course markers. Once the Academic Affairs Department summarizes the information on the opening of classes, it issues teaching assignments to each teaching unit and makes uniform arrangements based on the school's teaching resources.

3.3.2. Design process for scheduling questions

Figure 2 shows the genetic algorithm flow for the scheduling problem, the specific work of each part of the scheduling algorithm is as follows:

(1) Determine the data related to scheduling, which is the preliminary preparation for solving the scheduling problem.

(2) Gene coding design: according to the serial number of the teaching task in the teaching schedule, the classroom where it is located, the course time and the starting week are combined together for hybrid coding.

(3) Population initialization: different teaching tasks were brought with their own unique gene coding, called a gene. Several genes of teaching tasks form a data set, called a chromosome, a chromosome is the solution of the scheduling problem. Due to the characteristics of the genetic algorithm needs to initialize multiple chromosomes, combined together is the population, the more chromosomes in the population, that is, the larger the size of the population, the solution to the scheduling problem will be more diverse.

(4) Calculation of fitness function value: design the fitness function according to the soft constraints and calculate the fitness of each chromosome in the population.

(5) Selection: according to the calculated fitness function value, select chromosomes in the population to re-form a population according to the selection strategy.

(6) Crossover: Based on the selection operation, each chromosome is allowed to crossover according to the crossover strategy, i.e., to crossover the chromosomes in the population two by two at the specified loci, so that two different new chromosomes will be generated.

(7) Mutation: one or more genes, one or more loci in a chromosome are recoded in the population according to the mutation strategy, and the chromosome is mutated to a new chromosome.

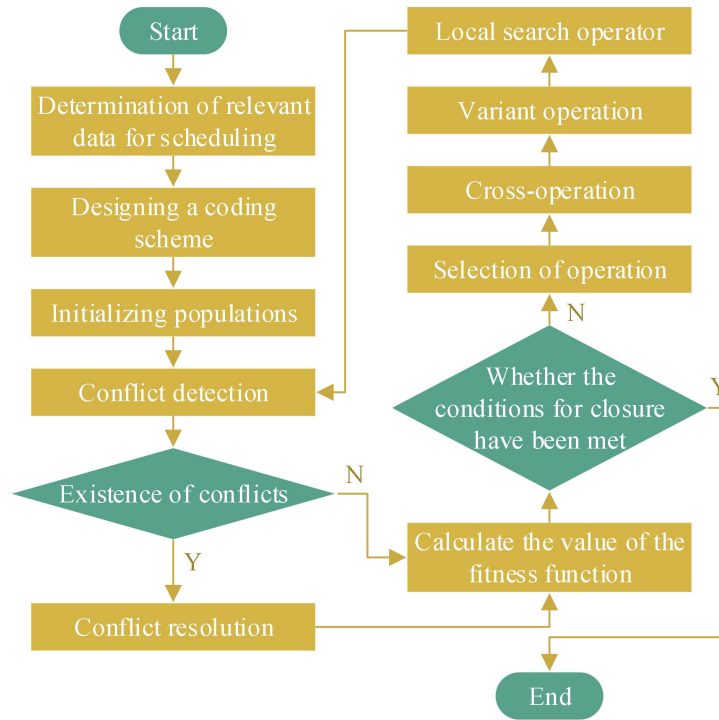


Figure 2. The genetic algorithm process for course scheduling problems

3.3.3. Coding scheme design

The first consideration of genetic algorithm is how to abstract the actual problem for encoding, in the class scheduling problem is to encode the teaching task. Commonly used coding methods include binary coding, Gray coding, etc., but for the class scheduling problem, the coding method is intuitive and easy to understand. For the class scheduling problem, this paper adopts the decimal variable length coding scheme. Although in the actual teaching, there will be a class taught by more than one teacher, a teacher at the same time on more than one course and so on, but for an administrative class or teaching class, the need to arrange the course, the teacher is scheduled before the start of the class has been set, which is reflected in the teaching schedule. Therefore, courses, teachers, and classes can be set as the same variable and correspond to classrooms and teaching hours.

Such coding ensures the uniqueness of each teaching task, and also shows the classroom used for the course, the teaching time, and the start and end weeks. And this variable length coding is more flexible, for example, the teaching time and the number of teaching weeks can be expressed in 2 decimal digits, and for the number of teaching tasks and the number of classrooms is less than one hundred or more than one thousand scheduling, it is only necessary to change the above three teaching tasks and classrooms into two or four or more digits. For all teaching tasks the length of the chromosome can be recorded as:

Total chromosome length:

$$\text{Total chromosome length} = \left(\begin{array}{l} \text{Instructional Task Serial} \\ \text{Number Code Length} + \\ \text{Classroom Serial Number \&} \\ \text{Instructional Time Code Length} + \\ \text{Instructional Week Code Length} \end{array} \right) \quad (37)$$

$$\begin{aligned} \text{Instructional task serial number code length} = \\ \text{number of instructional task serial number digits} \times n \end{aligned} \quad (38)$$

Classroom serial number and instructional time code length = $x_i \times (\text{number of classroom serial number digits} + \text{number of instructional time digits}) \times n_i$

$$i = 2, 4, 6, \dots, x_i = \frac{i}{2} \quad (39)$$

$$\text{Instructional week code length} = (\text{number of starting week digits} + \text{number of ending week digits}) \times n \quad (40)$$

In the above equation, n is the total number of courses and n_i is the number of courses with different number of weekly hours.

3.3.4. Initializing the population design

There is a certain amount of randomness and non-randomness in the generation of loci in each gene in a chromosome. The randomness of classroom generation is reflected in the fact that classroom numbers can be generated randomly, and the non-randomness is reflected in the fact that the value of the classroom serial number generated must be less than or equal to the number of classrooms. The randomness of the teaching time is reflected in the number of classes of the course for each weekly teaching day can be randomly generated, and the non-randomness is reflected in the fact that for the course with weekly credit hours of 6 (3 classes per week), the course schedule needs to have the number of days in between, and then it can only be Mondays, Wednesdays, and Fridays, and for the course with weekly credit hours of 4 (2 classes per week), the first time of scheduling is randomly generated from Mondays to Wednesdays, while the second class is just the number of days of the first class plus 2, while a course with 2 weekly credit hours (1 class per week) will not require an interval in instructional time, so Monday through Friday is randomly generated. The randomness of the starting week and the ending week is reflected in the starting week can be randomly generated in the first week of the teaching week to the last week, but in order to meet the objective rules of teaching, such as the ending week must be less than the last week of the semester two to three weeks, this is because the last two to three weeks will usually be used as an examination week without scheduling any courses, which non-randomly embodied in the end of the week not only according to the starting week and the total number of hours of study and the ratio of weekly hours and minus week, but also two to three weeks less than the last week of the semester.

3.3.5. Adaptation function design

In this paper, based on improving the basic genetic algorithm problem, a multi-objective optimization scheduling problem is proposed and a suitable fitness function is designed according to the optimization objectives. In the scheduling problem of this paper, multiple optimization objectives with different weights need to be satisfied, and the coordination curve method is used to design the fitness function on the basis of the Pareto concept. In this regard, each chromosome fitness function in the population is designed as follows:

$$f_{Fitness} = \sum_{i=1}^{13} \mu_i \times f_i(s) \quad (41)$$

where μ_1 is the weight of the function from the perspective of school resources, μ_2 to μ_6 are the weights of the function from the perspective of the teacher in the soft constraints, μ_7 to μ_{13} are the weights of the function from the perspective of the student in the soft constraints, and the value of μ_i can be defined according to the level of importance of each objective in the scheduling.

3.3.6. Cross-variant operator selection

For the specific value of the crossover probability there is no definite value, there has been more literature that the crossover probability should be set between 0.6 and 1. If the probability is 1, there is a possibility that the better individual will disappear, so the crossover probability is less than 1, which allows the better individual to have the possibility of survival.

A mutation operation is a numerical change in certain gene loci in a chromosome. A mutation operation is essentially a numerical change in the chromosome itself at certain loci with a small probability of mutation, a mutual change of 0s and 1s in the case of binary coding, or a re-generation of a new real value in the case of real number coding. The value of the probability of mutation in more literature that the probability of mutation should be set between 0.005 and 0.05, the mutation operation

can increase the ability of chromosome exploration, then the diversity of the population is stronger.

3.4. Testing of computer energy consumption under algorithm operation

3.4.1. Energy consumption and operating speed test results

In order to test the performance of this paper's data-flow oriented structured iterative solving algorithm against mainstream algorithms, data streams of different sizes are now selected as variables, and there are four means of testing them: the data-flow oriented structured iterative solving algorithm of the current study, the data-flow oriented automatic classification analysis algorithm, the data-flow oriented decision-tree analysis algorithm, and the data-flow oriented unstructured linguistic analysis algorithm, and their algorithmic energy consumption is shown in Fig. 3. .

As the size of the data stream increases, the amount of computer energy consumption under the operation of the four algorithms also grows, and there is a large difference in the growth rate. Among them, the algorithm studied in this paper has an analysis energy consumption of 2.545% at 100byte, with a span of 13.348% energy consumption at 1000byte, while the remaining three algorithms run under the amount of computer energy consumption spanning 18.671%, 30.421% and 22.323%. The comparison shows that the energy consumption of this algorithm is lower. The main reason is that in this paper, the data streams are structured and divided, and the normalization operation is performed on the given data streams to count them into a unified jurisdiction, which reduces the number of iterations and makes the algorithms consume less energy.

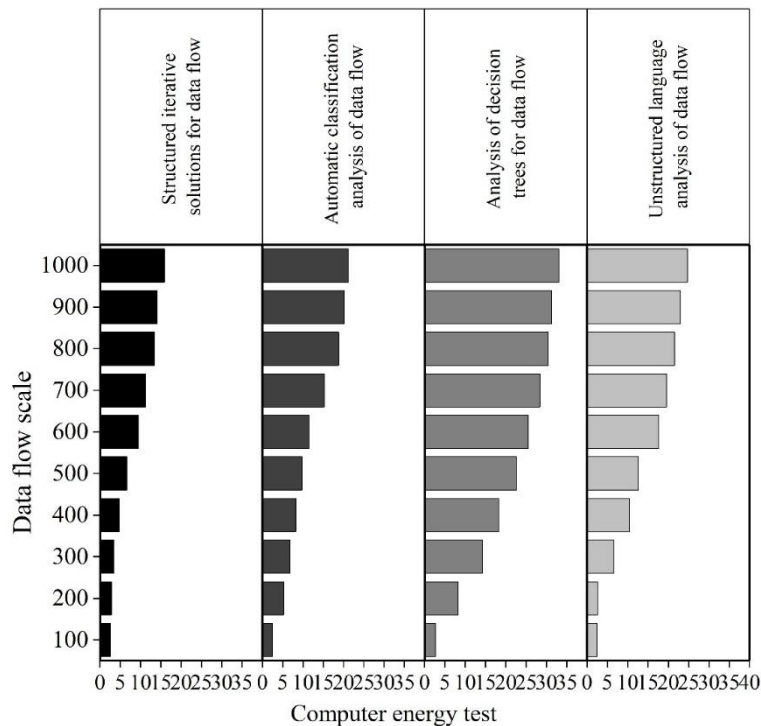


Figure 3. Algorithm energy consumption

In order to ensure the accuracy of the computer running speed test results under the operation of the algorithms, the same selection of different sizes of data streams as a variable, running four algorithms, and their running speed test results are shown in Figure 4.

With the increase in the size of the data stream, the computer running speed under the operation of the four algorithms is constantly slowing down, but there are differences in the range of change. When running the algorithm of this study, the computer running speed decreases from the original 259Hz to 215Hz, while under the operation of the remaining three algorithms, the computer running speed decreases from the beginning of 249Hz, 223Hz, 234Hz to 154Hz, 130Hz, 113Hz, respectively. It can be seen that the algorithms of this study do not slow down the computer running speed to a greater extent. The main reason for this is that the reduction in the number of iterations reduces the load on the system and ensures the computer running speed.

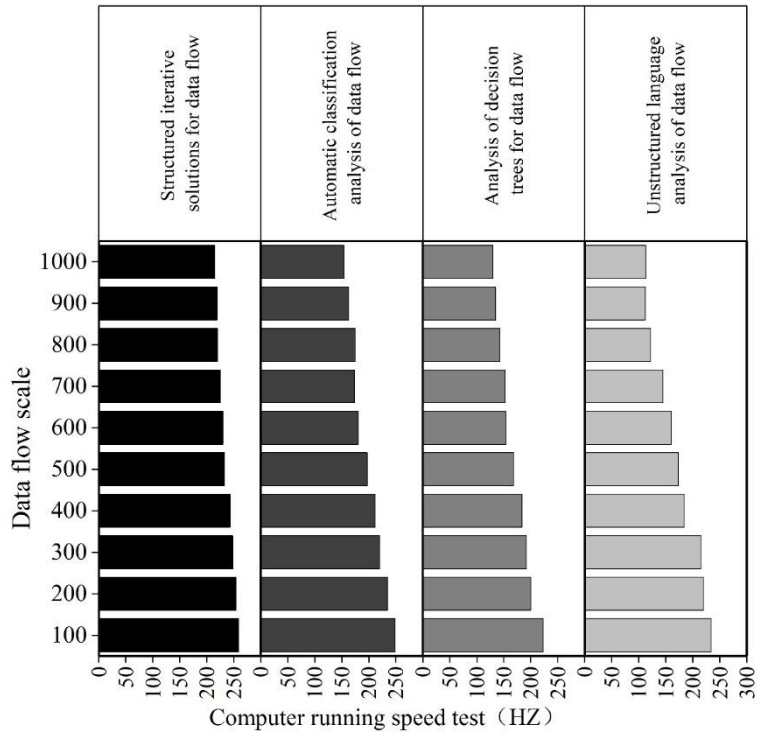


Figure 4. Running speed

3.4.2. Optimal Class Schedule Scoring and Number of Genetic Iterations

When conducting experiments, this system sets the number of crossovers to 18, the number of mutations to 2, the maximum number of population evolutionary generations to 10000, and the convergence condition is that the historical optimal class schedule scores remain unchanged for 2000 consecutive generations. After conducting the experiments, the current generation of optimal class schedule scores change with the number of genetic generations and the convergence speed change with the number of genetic generations are shown in Fig. 5.

The experimental results show that the use of genetic algorithm to find the optimal class schedule in this study does have a significant improvement in the rating compared to the deterministic solution obtained using the mainstream algorithm, and the rating is 97.058 when the number of evolutionary iterations of the genetic algorithm reaches 4500. In addition the score is close to perfect indicating that the final output class schedule basically fully satisfies the user-defined constraints based on certain satisfaction of soft and hard constraints. Meanwhile, the running time of the algorithm used in this system is extremely short, completing the entire iteration process in less than 2.5 s. This is because only a fixed number of times the new solution generation process is carried out, where a new solution is generated if it is successful, and vice versa the old solution of the previous generation is used, instead of generating a new solution again when the new solution fails to meet the requirements until it does so.

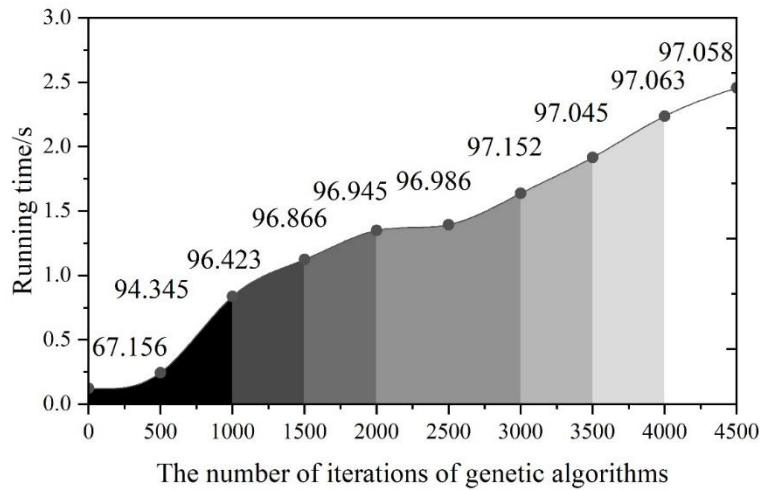


Figure 5. The optimal class score and convergence rate

3.4.3. Physical education course scheduling and selection performance

The performance statistics of this paper's algorithm for analyzing the scheduling and course selection data of different sports courses are shown in Figure 6. The check completeness rate and check accuracy rate of this paper's algorithm for analyzing the scheduling data and course selection data of 10 kinds of physical education courses in experimental universities are 96.925% and 97.074%, respectively, which are higher than 95%, and the analysis performance is excellent.

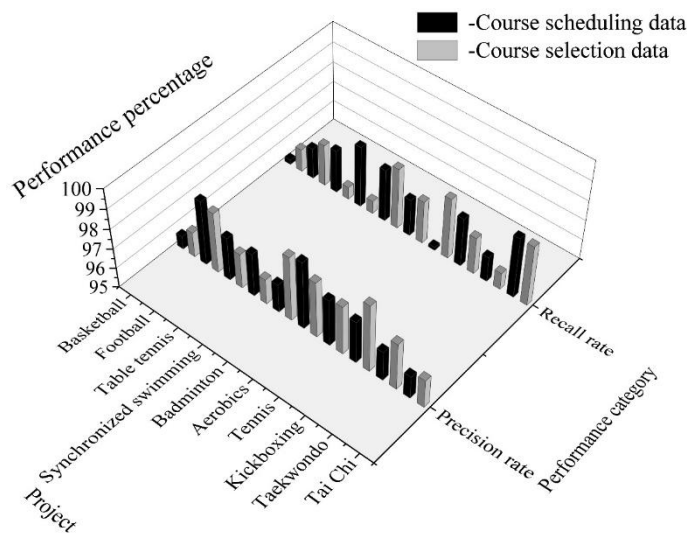


Figure 6. Data analysis performance

4. Analysis of the effect of physical education curriculum arrangement

4.1. Model Validation

Figure 7 shows the validation of per capita time consumed in changing classrooms. 21 physical education classrooms in the three academic buildings of the gymnasium were selected for the empirical study. Record the time of students entering and leaving the classroom in each physical education room during the physical education class on Friday of the 8th week, and the difference between the two times is the time consumed by each student on the way to change the classroom. Comparing the measured per capita time with the calculated per capita time according to the model genetic algorithm, it can be seen that the difference between the measured results and the model calculation results is approximate, the difference is not more than 0.8min and the trend is basically the same, which indicates that the constructed genetic algorithm has a high degree of credibility in the control of classroom change trip time.

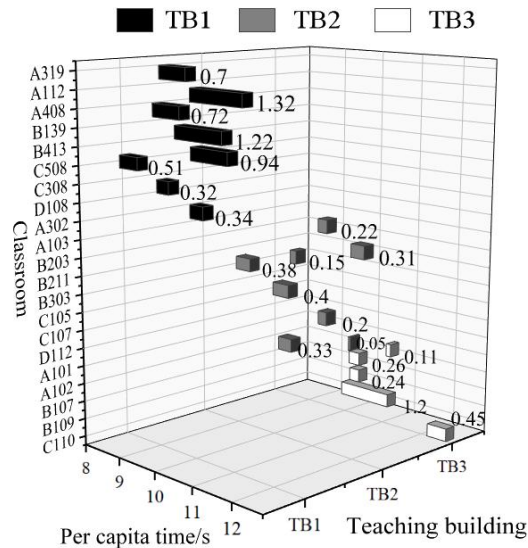


Figure 7. Change the average time of change in the classroom

Take Friday of the 8th week as an example, according to the status quo analysis to determine the value of each parameter, calculate the time consumption of each stage, according to the aforementioned optimization objectives, optimization principles for optimization, to determine the optimized classroom scheduling plan, and compare and analyze the status quo and optimized scheduling plan for students to change the classroom passage time consumption, Table 1 for the physical education scheduling plan before and after the optimization of the student's passage time between classes. Calculation shows that before and after the optimization of classroom scheduling plan, the per capita time consumed by students to change classrooms during class time is 10.04min and 6.93min, respectively, and the passage time consumption is reduced by 30.98%, and the number of the original classrooms is 115, and the number of optimized classrooms is 75, and the optimization plan reduces the passage time in all stages of the process of replacing classrooms.

Table 1. The time of the class is optimized before and after the students' time

Project	Queue up to enter and exit the classroom	
	Total time consumption/min	Per capita time consumption/min
Current situation	3145.66	0.63
Optimization	1922.56	0.45
Time consumption reduction rate /%	28.57%	
Project	Path passage	
	Total time consumption/min	Per capita time consumption/min
Current situation	28346.69	5.59
Optimization	19045.69	3.75
Time consumption reduction rate /%	32.92%	
Project	Fetching water during the break	
	Total time consumption/min	Per capita time consumption/min
Current situation	7216.69	1.48
Optimization	5134.95	1.09
Time consumption reduction rate /%	26.35%	
Project	Use the toilet during the break	
	Total time consumption/min	Per capita time consumption/min
Current situation	11324.98	2.34
Optimization	8243.42	1.64
Time consumption reduction rate /%	29.91%	
Project	In total	
	Total time consumption/min	Per capita time consumption/min
Current situation	50034.02	10.04
Optimization	34346.62	6.93
Time consumption reduction rate /%	30.98%	

4.2. Analysis of the effect of scheduling optimization

4.2.1. Pareto-optimal frontiers

Figure 8 shows the Pareto optimal frontier in the results of solving the algorithm, the Pareto frontier shows a fluctuating trend, the increase of a single number of additional conflicts may further reduce its satisfaction value, analyze the reason for this is that the time change and classroom change have different penalty values, when one additional conflict is allowed, classroom change can be carried out so as to satisfy the classroom stability requirements of a course, and when the number of additional conflicts is increased to two, the changes that can be carried out include, two classroom changes or one time change, and two classroom changes will further deteriorate classroom stability when there is less room for favorable changes. be made include, two classroom changes or one time change, and two classroom changes can further deteriorate classroom stability when there is less room for favorable changes. In this case, the minimum number of conflicts to return to the feasible solution is 19 with an objective value of 0.533, and the number of conflicts to return to the optimal solution is 33 with an objective value of 0.558, which is a 4.7% increase in the objective value with respect to the minimum conflict solution, so that the optimal solution can be restored when the number of additional conflicts is 14 in the algorithm, and the consideration of the additional conflicts can further optimize the objective value.

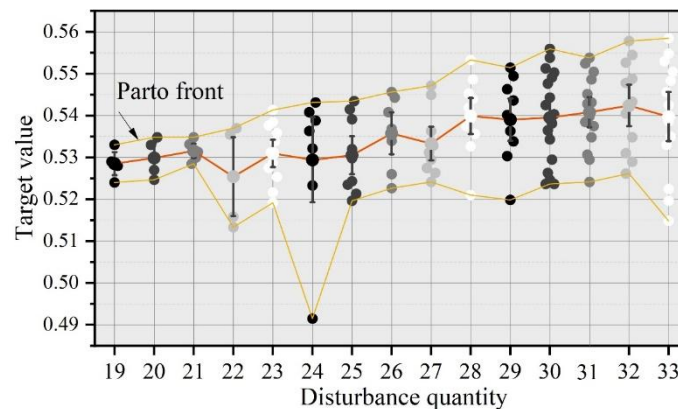


Figure 8. Pareto's best frontier

4.2.2. Sensitivity analysis of scheduling requirements

The pre-semester physical education course scheduling was weighted to sum the satisfaction of the four demands in order to serve as a target value, and a sensitivity analysis was performed in order to analyze the necessity of the four demands. The changes in the target value with or without considering each demand were analyzed separately to verify their necessity. Figure 9 shows the results of the sensitivity analysis of different scheduling needs, where the example is calculated separately under the objective function of different needs, and the objective values are calculated in a uniform way for comparison.

The changes in the objective values also differ when different demands are not taken into account, where the differences in the changes in the objective values of section superiority, discretization and classroom stability are small, while the differences in the changes in the objective values of walking distance are large. Without considering the demand for walking distance, the satisfaction value of walking distance decreased from 0.999 to 0.485, and the total objective value decreased from 0.678 to 0.548, which shows that this demand only needs a small weight to achieve a large quality improvement, and it is necessary to use this demand as part of the objective function, which further shows that the model's performance on the demand for scheduling is sensitive, and the quality satisfaction of the scheduling quality satisfaction is improved.

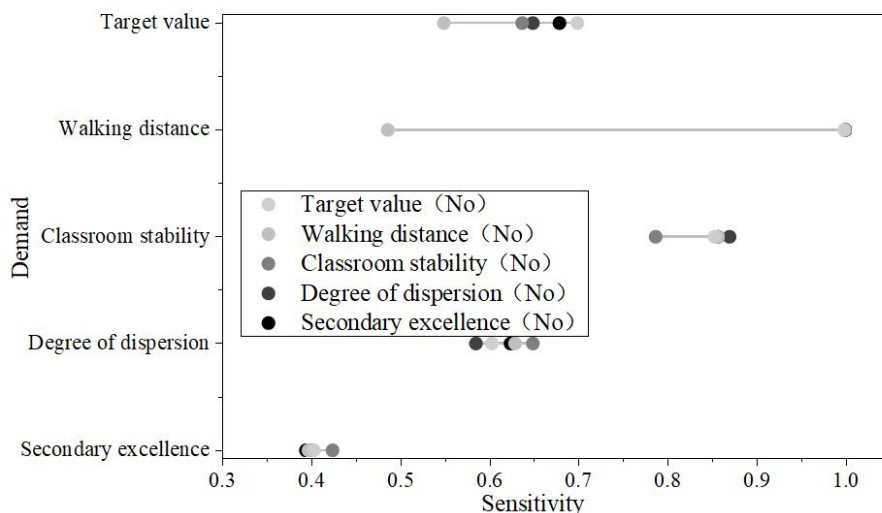


Figure 9. Analysis of demand sensitivity of different scheduling courses

5. Conclusion

In this paper, data flow equations and aggregation operations are used to solve the basic arithmetic problems of data flow, and two iterative solution algorithms are proposed for the use of data flow analysis techniques, namely, structured program and full path forward flow. The objective function is set, the relevant mathematical model of the sports course scheduling problem is established, the genetic method is used to solve it optimally, and the usability of the model is examined through simulation tests. The research results of this paper are as follows:

(1) In order to test the performance difference between the data stream algorithm used in this paper and the mainstream algorithm, data streams of different sizes are selected as variables to jointly test the energy consumption and running speed of the algorithm. The algorithm studied in this paper analyzes the energy consumption of 2.545% at 100byte, and the energy consumption span with 1000byte realization is 13.348%, and the remaining three algorithms under the same conditions, the computer energy consumption span is 18.671%, 30.421% and 22.323% respectively. Meanwhile, the running speed of this paper's algorithm is reduced from 259 Hz to 215 Hz with less change in running speed. The algorithm in this paper outperforms the remaining mainstream algorithms in terms of energy consumption as well as running speed.

(2) The demand sensitivity of scheduling is analyzed, without considering the walking distance, the satisfaction of the walking distance is reduced from 0.999 to 0.485, and the total objective value of 0.678 is reduced to 0.548, which shows that the quality of scheduling is improved, and also verifies that the objective function set by the algorithm is correct.

Funding

This research was supported by the Scientific research project of universities in Inner Mongolia Autonomous Region: Research on "integrated" teaching mode of physical education in middle and primary schools in Inner Mongolia (Project plan number: NJSY21020).

References

1. Mcveagh, H., Smith, M., & Randall, V. (2022). 'It's like it doesn't really matter': Are teachers accountable and equipped for teaching primary physical education?. *Education 3-13*, 50(2), 225-237.
2. Griggs, G., & Randall, V. (2019). Primary physical education subject leadership: Along the road from in-house solutions to outsourcing. *Education 3-13*, 47(6), 664-677.
3. Xing, Z., & Ge, C. (2025). The relationship between physical exercise and social adjustment in Chinese university students: the sequential mediating effect of peer attachment and self-esteem. *Frontiers in Psychology*, 16, 1525811.
4. Guo, W. (2020). Reflection on the lack of physical education accomplishment of college students and teaching enlightenment. *International Journal of New Developments in Education*, 2(9.0).
5. Zhou, Z., Guo, K., Guo, S., & Chen, L. (2024). Relationship between physical exercise and college students' social adaptation: the chain mediating role of self-esteem and peer attachment. *Frontiers in Psychology*, 15, 1453675.

6. Edo, B., Tadesse, T., & Mulugeta, E. (2019). Students' and Teachers' perceptions and experiences of course scheduling in undergraduate sports sciences program: An Ethiopian case study. *Journal of University Teaching and Learning Practice*, 16(3), 1-21.
7. Casado-Robles, C., Mayorga-Vega, D., Guijarro-Romero, S., & Viciano, J. (2020). Sport education-based irregular teaching unit and students' physical activity during school recess. *The Journal of Educational Research*, 113(4), 262-274.
8. Ferry, M. (2016). Teachers in school sports: between the fields of education and sport?. *Sport, Education and Society*, 21(6), 907-923.
9. Zhu, X., Ennis, C. D., & Chen, A. (2011). Implementation challenges for a constructivist physical education curriculum. *Physical Education and Sport Pedagogy*, 16(1), 83-99.
10. Hariri, B., Asmawi, M., & Setiakarnawijaya, Y. (2021). The Effect of Physical Education Learning Schedule, Facilities, and Motivaton on Learning Outcomes of Eighth Grade Student. *Gladi: Jurnal Ilmu Keolahragaan*, 12(02), 128-137.
11. Kilborn, M., Lorusso, J., & Francis, N. (2016). An analysis of Canadian physical education curricula. *European Physical Education Review*, 22(1), 23-46.
12. Wu, Y., Van Gerven, P. W., de Groot, R. H., O. Eijnde, B., Seghers, J., Winkens, B., & Savelberg, H. H. (2023). The Association between Academic Schedule and Physical Activity Behaviors in University Students. *International Journal of Environmental Research and Public Health*, 20(2), 1572.
13. Mazzucca, S., Hales, D., Evenson, K. R., Ammerman, A., Tate, D. F., Berry, D. C., & Ward, D. S. (2018). Physical activity opportunities within the schedule of early care and education centers. *Journal of Physical Activity and Health*, 15(2), 73-81.
14. Zhang, T., Chen, A., Ennis, C., Loflin, J., Hong, D., Wells, S., & Schweighardt, R. (2014). Class Schedule and Learning Achievement Through Constructivist Physical Education. *Research Quarterly for Exercise and Sport*, 85(S1), A164.
15. Nettlefold, L., McKay, H. A., Warburton, D. E. R., McGuire, K. A., Bredin, S. S., & Naylor, P. J. (2011). The challenge of low physical activity during the school day: at recess, lunch and in physical education. *British journal of sports medicine*, 45(10), 813-819.
16. Su, J., & Zhong, Y. (2022). Artificial Intelligence (AI) in early childhood education: Curriculum design and future directions. *Computers and Education: Artificial Intelligence*, 3, 100072.
17. Batini, C., Nardelli, E., & Tamassia, R. (2012). A layout algorithm for data flow diagrams. *IEEE Transactions on Software Engineering*, (4), 538-546.
18. Cantabella, M., Martínez-España, R., Ayuso, B., Yáñez, J. A., & Muñoz, A. (2019). Analysis of student behavior in learning management systems through a Big Data framework. *Future Generation Computer Systems*, 90, 262-272.
19. Wang, T., Xiao, B., & Ma, W. (2022). Student behavior data analysis based on association rule mining. *International Journal of Computational Intelligence Systems*, 15(1), 32.
20. Chen, M., Huang, X., Chen, H., Su, X., & Yur-Austin, J. (2023). Data driven course scheduling to ensure timely graduation. *International Journal of Production Research*, 61(1), 336-361.
21. Qu, Y., & Kutscher, A. (2018, October). Apply Data Analytics to Schedule Best-suited Classes for Students with Different Academic Histories. In *2018 IEEE Frontiers in Education Conference (FIE)* (pp. 1-7). IEEE.
22. Çaylı, O., & Yilmaz, A. (2025). Smart Classroom Scheduling and Resource Optimization for Educational Institutions: Integrating AI and Multi-Objective Decision Support. *The European Journal of Research and Development*, 5(1), 95-112.
23. Diallo, F. P., & Tudose, C. (2024). Optimizing the scheduling of teaching activities in a faculty. *Applied Sciences*, 14(20), 9554.
24. Yuan, B. (2023). Optimization of physical education teaching strategies in colleges and universities in the context of modern information technology. *Curric Teach Methodol*, 6, 47-53.
25. Soares, J., Dias, C., Fernandes, E., & Antunes, H. (2021). Incongruences in the scheduling of Physical Education classes. *Journal of Physical Education and Sport*, 21(4), 1935-1942.
26. Harris, J., Cale, L., & Musson, H. (2012). The predicament of primary physical education: A consequence of 'insufficient'ITT and 'ineffective'CPD?. *Physical Education and Sport Pedagogy*, 17(4), 367-381.
27. Mohammed, Z. (2016). The impact of the lack of physical education and sports lesson on fitness health-case female students. *European Journal of Physical Education and Sport Science*.
28. Turner, L., Johnson, T. G., Calvert, H. G., & Chaloupka, F. J. (2017). Stretched too thin? The relationship between insufficient resource allocation and physical education instructional time and assessment practices. *Teaching and Teacher Education*, 68, 210-219.
29. Osborne, R., Belmont, R. S., Peixoto, R. P., Azevedo, I. O. S. D., & Carvalho Junior, A. F. P. D. (2016). Obstacles for physical education teachers in public schools: an unsustainable situation. *Motriz: Revista de Educação Física*, 22, 0310-0318.
30. Demirhan, G., Bulca, Y., Saçlı, F., & Kangalgil, M. (2014). Physical education teachers' problems in practice and suggested solutions. *Hacettepe University Journal of Education*, 29(2), 57-68.
31. Tao, S., Sheng-Ping, Z., & Meng-Yuan, W. (2025). Optimization of school physical education schedules to enhance long-term public health outcomes. *Frontiers in Public Health*, 13, 1548056.
32. Tan, L., Chen, Q., Wu, J., Li, M., & Liu, T. (2025). Optimizing physical education schedules for long-term health benefits. *Frontiers in Public Health*, 13, 1555977.
33. Zhang, Q. (2022). An optimized solution to the course scheduling problem in universities under an improved genetic algorithm. *Journal of Intelligent Systems*, 31(1), 1065-1073.

-
34. Olmos, L., Kaplan, A., & Jalalitar, M. (2024, December). Course Scheduling Optimization Using NLP and Algorithmic Approaches. In *International Conference on Computational Science and Computational Intelligence* (pp. 151-165). Cham: Springer Nature Switzerland.
 35. Zheng, H., Peng, Y., Guo, J., & Chen, Y. C. (2022). Course scheduling algorithm based on improved binary cuckoo search. *The Journal of Supercomputing*, 78(9), 11895-11920.
 36. Subang, K. N., Balaba, E. I., & Agoylo Jr, J. C. (2024). Optimizing Course Scheduling with Genetic Algorithms: A Dynamic Approach. *SAR Journal* (2619-9955), 7(4).
 37. Chen, X., Yue, X. G., Li, R., Zhumadillayeva, A., & Liu, R. (2021). Design and application of an improved genetic algorithm to a class scheduling system. *International Journal of Emerging Technologies in Learning (iJET)*, 16(1), 44-59.
 38. Zhang Yan & Li Ningchuan. (2022). Analysis of Passenger Flow Characteristics of Urban Rail Transit Based on Spatial Data Dynamic Analysis Technology. *Mathematical Problems in Engineering*, 2022, <https://doi.org/10.1155/2022/1601169>.
 39. S. M. Demidov. (2025). Iterative Algorithms for Solving the Direct and Inverse Problems of the Positions of Mechanisms with Parallel Structure. *Journal of Machinery Manufacture and Reliability*, 54(3), 231-238. <https://doi.org/10.1134/S1052618825700025>.
 40. Adler Nascimento, Alexandre Maul & Wagner Moreira Lupinacci. (2025). A novel approach to advancing the application of genetic algorithms in sequential seismic inversion. *Computational Geosciences*, 29(5), 42-42. <https://doi.org/10.1007/S10596-025-10381-7>.
 41. He Ren, Wenyi Liu, Yonghua Jiang & Xiuping Su. (2017). A novel wind turbine weak feature extraction method based on Cross Genetic Algorithm optimal MHW. *Measurement*, 109, 242-246. <https://doi.org/10.1016/j.measurement.2017.05.059>.