Research on Class Combination Algorithm Based on Genetic Algorithm

Linjing Yan* and Xiangming Kong
Beijing Polytechnic, Beijing 100176, China

*Corresponding author email: 101457@bpi.edu.cn

Abstract. The scheduling problem is a typical multi-objective optimization problem, and the core problem involved is the problem of combining classes. On the basis of a comprehensive analysis of the problem of class combination, this paper establishes a mathematical model of the three-dimensional class combination problem according to the needs, and constructs the chromosome expression in the genetic algorithm, and obtains the optimal solution of the problem. Experiments show that this algorithm can effectively obtain the optimal three-dimensional class combination program, and has strong practicability.

1. Introduction
The scheduling problem is the most critical problem in teaching management. In recent years, with the rapid development of computer technology, computer scheduling has become the most advanced educational management method. In China, due to the large differences in teaching facilities and other conditions among universities, there is no universal lesson scheduling system so far. In the existing computerized course scheduling systems, there is still a widespread phenomenon of dropping classes (the courses that cannot be placed in the schedule due to the inability to arrange the classroom or time), which seriously affects the use of the system. The workload of manual adjustment is not much smaller than the workload of rearranging classes. Among the factors that cause the phenomenon of class rejection, the problem of co-class is a particularly important factor. This problem has become a key problem that troubles and hinders the continued research and practical application of the class scheduling system. At the same time, the expansion of enrollment in my country's universities in recent years has made the problem of insufficient teaching equipment and teacher resources appear particularly prominent, and the problem of co-classes has become more and more important.

This paper studies the three-dimensional teaming problem and the promotion of the multi-dimensional teaming problem, combined with the solved professional difference minimization problem, can get a complete teaming plan. At this point, the problem of co-class has been completely resolved. The problem of three-dimensional co-class can be described as follows: a class needs to take the same three courses, and the number of lecture podiums for each course is known, and the number of classes in each podium is known, so the number of scattered podiums is the smallest For the goal, a reasonable combination of these classes is the problem of three-dimensional class combination. In the same way, if a class needs to take the same multiple courses, it becomes a multi-dimensional problem. In this paper, the genetic algorithm is used to solve the three-dimensional teaming problem, including establishing a reasonable mathematical model for the three-dimensional teaming problem, and designing the genetic operation method to construct a complete genetic algorithm.
2. Mathematical Model of Three-dimensional Class Combination Problem

There are three classes that require three courses, and the three courses are distributed on the podium (known). The number of pre-arranged classes for each platform is known, and they are set to \(a_1, a_2, \ldots, a_l\); \(b_1, b_2, \ldots, b_j\); \(c_1, c_2, \ldots, c_k\). In order to meet the minimum number of scattered podiums, there are the following objective functions:

\[
f(l) = \min \left( \sum_{XYZ} \text{sgn} l_{ijk} \right)
\]

Among them, \(l_{ijk}\) is the number of classes distributed in \(X\) course \(i\) podium, \(Y\) course \(j\) podium, \(Z\) course \(k\) podium.

Restrictions: s.t

\[
\sum_{XYZ} l_{ijk} = M
\]

\[
\sum_{YZ} l_{ijk} = a_i
\]

\[
\sum_{XZ} l_{ijk} = b_j
\]

\[
\sum_{XY} l_{ijk} = c_k
\]

\((i = 1, 2, \ldots, l; j = 1, 2, \ldots, m; k = 1, 2, \ldots, n)\)

3. The Genetic Algorithm Design of Three-dimensional Class Merging Problem

It can be seen from the above model that solving the problem of three-dimensional class combining is to reasonably allocate the classes that meet the constraints of each podium, so as to minimize the number of scattered podiums, that is, the least number of podiums occupied. Therefore, the genetic algorithm can be constructed as follows and combined with a simple example to illustrate its application.

3.1. Construct Chromosomes and Generate Initial Population

The purpose of constructing chromosomes is to generate the initial population, and each chromosome corresponds to a solution of genetic algorithm. According to the specific situation of this subject, the chromosome encoding method should adopt decimal encoding, and the chromosome representation adopts matrix form, as follows:

\[
\begin{pmatrix}
I_{111} & I_{112} & L & I_{11k} \\
M & M & M & M \\
I_{m1} & I_{m2} & L & I_{mk} \\
M & M & M & M \\
\end{pmatrix}
\]

Each element in the matrix is a combination of any three podiums belonging to three different courses. The entire matrix is arranged in a certain order, and the element values in the matrix are determined, and then a three-dimensional combined class plan is determined.

The following uses a rule to generate the initial population: ① Determine the value of the elements in the matrix one by one according to an order in the matrix. ②The method is to find the smallest number of classes held by the corresponding three podiums, as the value of the element in the corresponding matrix, and the original class number of each podium minus the value of the element in the matrix as the new podium accommodating class number. Continue in sequence until the number of
classes that can be accommodated in each stage is 0, that is, a chromosome is determined. It can be seen that a chromosome is generated by determining a sequence. In order to ensure the diversity of the population, we generate several chromosomes according to the order of the rows (change the position of the starting line one by one or alternate between each other), and also generate several chromosomes according to the order of the columns, according to the diagonal of the two directions. Several chromosomes are generated in a line sequence until the initial population size is met, that is, the initial population is generated.

3.2. Determination of Fitness Function
In evolution, the genetic algorithm judges whether to enter the next generation population based on the fitness value of each individual. Therefore, the setting of the fitness function directly affects the convergence speed of the genetic algorithm and whether the optimal solution can be found. In this algorithm, for each chromosome in a generation population, the objective function value is obtained. Since the fitness function requires the largest and non-negative conditions, the fitness function is determined as, where is the objective function, and the fitness The chromosome with the largest function value corresponds to the three-dimensional teaming plan with the smallest objective function, which is what we seek.

3.3. Natural Selection
Using fitness ratio selection and elite retention strategy, the process is as follows: Arrange a total of \( L \) chromosome in each generation population according to fitness value \( f_i \) from largest to smallest \( (h=1, 2, L, n) \), the top individual has the best performance, copy it and enter directly The next generation population. The other \( L-1 \) chromosomes of the next generation population are generated from the \( L \) chromosomes of the previous generation population using a roulette method, which can ensure that the best person survives to the next generation, and can avoid the difference in fitness values between individuals to be selected There is a huge disparity in opportunities to maintain the diversity of the next-generation population, thereby effectively improving the convergence speed of the algorithm.

3.4. Chromosome Cross Recombination
For the new population generated in 3.3, select individual pairs for single-point crossover with probability \( p_c \). This article chooses \( p_c = 0.6 \). Due to the constraint of the three-dimensional teamwork problem, the particularity of the crossover is determined. After the crossover, appropriate adjustments must be made to meet the constraints. condition. The specific process is as follows: Generate two random numbers in the range of ranks and columns, one representing a row and one representing a column, so that it corresponds to an element in the matrix. Exchange the value of the element in the two chromosomes. After the exchange, use the constraint condition to test, and the non-conformity The constraint conditions are automatically adjusted to meet the constraint conditions and the crossover is completed. Calculate the fitness function value of each chromosome for the offspring produced after the crossover, and compare it with the parent to select the individual with the larger fitness function value to enter the population.

3.5. Mutation Operator
In the population generated after the above crossover, the chromosome is mutated at a mutation rate of \( p_m = 0.02 \), and a special mutation strategy is also used. The specific process is as follows: Use the crossover operator to generate a random number to find an element, check the value of the value and the size of the class number of the three podiums corresponding to the element, if the value is less than the cross value, then the value will be changed to the cross value, Then automatically adjust the values of other elements to meet the constraint conditions; if the value is equal to the intersection value, set the value to 0, and then adjust to meet the constraint conditions, and the mutation ends. Compare the fitness value of the successfully mutated chromosome with the value of
the fitness function of the parent body, and select the larger fitness function value to enter the population.

3.6. Judging the Conditions for Stopping Evolution
The last step of genetic algorithm design is the control parameters and the algorithm termination conditions. The crossover rate and mutation rate have been given above. The maximum evolution algebra is set to 150. The judgment process is as follows: determine whether the iteration algebra is the maximum evolution algebra 150, and if so, stop the evolution, Choose the best chromosome as the best three-dimensional hybrid optimal solution, or if the fitness function values of the best chromosomes of 20 consecutive generations are equal, stop, and output the best chromosome as the best three-dimensional hybrid optimal solution.

4. Multi-dimensional Team Promotion
Using the above method, as long as it is ensured that the elements in the matrix representation of the chromosome contain all the multi-dimensional combinations, and then these elements are arranged in any order, the multi-dimensional teaming problem can be solved.

5. Examples and Case Analysis
In order to better illustrate the operation process of the three-dimensional integrated genetic algorithm designed in this article, a simple example is given below:

Example: There are 12 classes all taking the same 3 courses. According to the actual situation, the three courses have 3, 4, and 2 podiums. The number of classes required for each podium is as follows:

| Podium | Course 1 | Course 2 | Course 3 |
|--------|----------|----------|----------|
| 1      | 4        | 4        | 6        |
| 2      | 4        | 4        | 6        |
| 3      | 4        | 2        | 0        |
| 4      | 0        | 2        | 0        |

Solution:
1) The initial population size is set to 40.
2) Taking the first row as the first row and the second row as the first row respectively, the two chromosomes generated in a row-by-row order are as follows:

\[
\begin{pmatrix}
4 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 2
\end{pmatrix}^T \tag{7}
\]

\[
\begin{pmatrix}
0 & 4 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2
\end{pmatrix}^T \tag{8}
\]

3) Generate 10 in the order of row, column, and diagonal in both directions to determine the initial population.
4) Calculate the fitness function values of the two chromosomes as \( f_k = 1/ f(l) = 1/5 \) and \( f_k = 1/ f(l) = 1/7 \) respectively.
5) Use the natural selection method designed in this paper to generate the population to be crossed.
6) Taking the above two chromosomes as an example, suppose the two random numbers generated are 5 and 2, exchange the values in the 5th row and the 2nd column of the two chromosome matrices to obtain two new chromosome matrices. Measured with constraint conditions, it is found that these two co-working schemes do not meet the requirements of the question. After adjustment, the following two chromosome matrices are obtained:
Calculate the value of the fitness function of the chromosome as $f_k = 1/f(l) = 1/5$ and $f_k = 1/f(l) = 1/6$, and take the larger fitness function value of the two generations to enter the next generation, so the second chromosome is replaced with the new second chromosome.

7) Take the first chromosome as an example to mutate it: set the random number to be 6 and 1, and check that the value in the sixth row and the first column is 2, which is less than $I_{221}$, the minimum value 4 of the three podiums, so this value Take 4, and adjust the values of other elements in the matrix to make it meet the requirements of the question. The new chromosome matrix is obtained as:

$$
\begin{pmatrix}
0 & 4 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 2 & 0 \\
2 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 2 & 0 & 2
\end{pmatrix}^T
$$

(9)

$$
\begin{pmatrix}
2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 0
\end{pmatrix}^T
$$

(10)

Calculate its fitness function value is also $f_k = 1/f(l) = 1/5$, so there is no need to replace, but if it is greater than $1/5$, replace the original chromosome with the mutated chromosome to enter the next generation.

8) The new population is re-run as the parent cycle until the constraint conditions are met, and the optimal chromosome is output, which is the optimal three-dimensional teaming plan we seek.

In addition, this article conducted experiments on actual data from a certain university. The results of a set of data are as follows: Three courses are advanced mathematics, computer basics, and ideological and moral training. There are a total of 90 classes. According to the actual situation of our school, three courses There are 17, 18, and 14 podiums, but the number of classes required for each podium is not the same (known). After data statistics, it is found that the number of podiums occupied by this three-dimensional combined class in our school’s actual class schedule is 37. The result of the class is too scattered, and the effect of co-working is not satisfactory. However, using the three-dimensional teaming algorithm designed in this paper to perform teaming, the result is more ideal, and the number of platforms occupied is only 28. Obviously, the results obtained by the genetic algorithm designed in this paper are more ideal.

At the same time, the full search was used to verify. In this example, the genetic algorithm designed in this article can search for the optimal solution. In addition, in the verification of other 15 groups of courses, the probability of searching for the best is more than 95%. This method can find a satisfactory solution to the three-dimensional teamwork problem, which is a better method.

6. Conclusion

In this paper, genetic algorithm is used to efficiently solve the problem of combining classes in the scheduling system, which can be well applied in the actual timetable scheduling process, and the three-dimensional class combining algorithm in this paper can be promoted smoothly, which also improves the practicality of this algorithm. The actual verification shows that the design of co-working according to this algorithm is reasonable and fast, and has strong practicability.

References

[1] Zhang Zhigang, et al. The Mathematical Model and Algorithms of the Class Schedule Problem. Computer Development [J]. June 2004, Volume 14:131-136

[2] Li Minqiang, et al. The basic theory and application of genetic algorithm [M]. First edition. Science Press, 2002: 34-47.
[3] Asratian A.S., Werra D. de. A generalized class-teacher model for some timetabling problems. European Journal of Operational Research [J]. 2002, 143: 531-542.