

Arranging course automatically algorithm based on multi-factor priority

Zhiwei

Huilongguan crew center
Beijing Subway
Beijing, China

Fenghuaping

The department of the information engineering
The gengdan institute of beijing university of
technology
Beijing, China

Abstract—The problems of course timetabling mainly involve time, curriculum, the number of classrooms and so on. A new algorithm based on multi-factor priority is proposed in this paper. The algorithm use time division, the compute of the priority, the provisional matrix and quick sort to achieve the function of arranging course automatically. Finally, the paper gives some examples to prove the feasibility of the algorithm.

Keywords- arranging course automatically algorithm; time division; the compute of the priority; quick sort; provisional matrix

I. INTRODUCTION

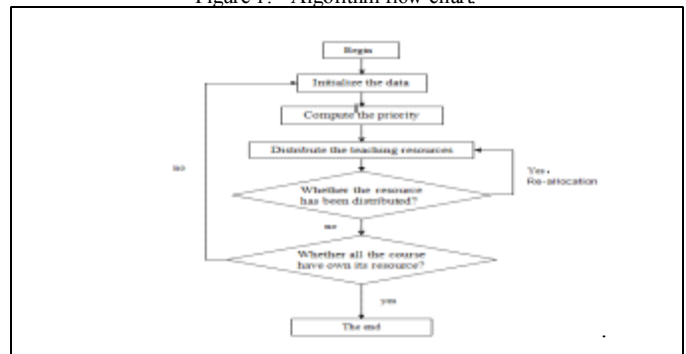
Arranging course is a complex process. There are many algorithms can solve this problem at home and abroad now, such as genetic algorithms^[1], greedy algorithm^[2], arranging algorithm based on artificial intelligence^[3]. Although these algorithms are feasible to improve the performance and the efficiency of the course timetabling system, they take up more space and very difficult to design. This is mainly because course timetabling problem involves many factors, and these factors bring a lot of difficulties to the process of design. Colleges and universities need to consider some factors such as time, curriculum, classroom, classes and teachers when they arrange the course for the students. This paper proposed a priority of arrangement algorithm based on a number of factors on the basis of a variety of arrangement algorithm researches. The algorithm can achieve a simple and easy arrangement function according to the time division, the compute of the priority, the provisional matrix and quick sort to.

II. BASIC IDEA of ALGORITHM

The basic idea of the algorithm is to calculate their priority according to the specific information of the course (such as: the importance of course, weekly hours, and class attendance), and then the algorithm sorts these courses by the quick sort based on the priority. Finally, on the basis of the results it inquires the time table to distribute the teaching resources .In addition, in order to reducing conflict, the algorithm uses two-dimensional matrix to identify the allocation of resources. This method can avoid distribute the

resources which has been occupied by other courses. The algorithm specific steps are shown in Figure 1.

Figure 1. Algorithm flow chart



The specific steps of the algorithm :

- Initialize the data. Initializing the information of the course and the matrix;
- Compute the priority. According to the priority of each course, weekly hours, and class attendance, using formula to compute the priority(the formula is given in the next section);
- Distribute the teaching resources. According to the classroom types and weekly hours, distributing the resources by the appropriate module;
- Search time matching table. Inquiring available time and checking classes - time matrix, to confirm whether the class can use this time.
- Resolve the conflict. After the time has been distributed, then randomly to distribute classroom to the course and check the classroom - time matrix to determine the distribution of classroom whether can be use.

III. KEY ALGORITHMS

A. The Time Division

The algorithm provisions that there are five days have class each week, and each day is divided into five time periods, the first and second lesson is the first time section,

3rd and 4th is the second, 5th and 6th is the third, 7th and 8th is the fourth, and last the 9th and 10th is the fifth section. From Monday to Friday, a week can be divided into 25 time sections, and the 25 time sections will be numbered from 1 to 25. It will be shown in table 1.

TABLE I. TIMESLOT TABLE

		Monday	Tuesday	Wednesday	Thursday	Friday
a.m.	1-2	1	6	11	16	21
	3-4	2	7	12	17	22
p.m.	5-6	3	8	13	18	23
	7-8	4	9	14	19	24
evening	9-10	5	10	15	20	25

B. The Identify of Course Priority and Priority

Courses priority: In the teaching process, according to the importance, the different courses can be classified into public courses, professional courses, elective courses, sport and so on. The algorithm also provides that: 1. public courses as level 1; 2. professional courses as level 2; 3. sport as level 3; 4. elective courses as level 4.

Priority: Course timetabling is actually related to the class, course, time, teachers, and classroom and so on. The relationships among teachers, classes and courses have been identified in the teaching plan submitted by the teacher. Therefore, the algorithm is mainly considered how to distribute time and classroom for the courses. Before distribute the teaching resources, we must compute the priority of every course. The design principles of priority are statement in the following paragraph:

First, public courses must get the resources before the elective courses;

second, the courses have more weekly hours will get the resources earlier;

third the courses have more people attend will get resources earlier;

Fourth, the factors should be considered more balance. According to these principles, we can see that priority will be identified through the importance of the linear relationship among these factors, the linear relationship shown in Figure 2.

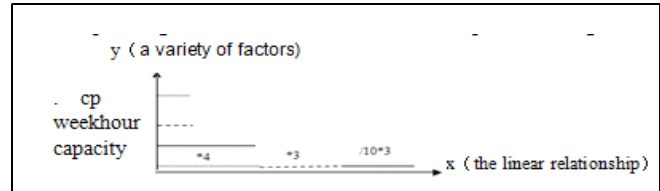


Figure 2. the linear relationship.

According to the above principles, the calculation of the priority expressed by the equation (1):

$$\text{priority} = \text{cp} * 0.4 + \text{weekhour} * 0.3 + \text{capacity} / 10 * 0.3 \quad (1)$$

In the equation(1), cp represents for the Courses priority, and it has been introduced at the begin of this section; weeklyhour represents for the number of course times per week; capacity represents for how many people will attend the course. According to the principles of priority, we may set the proportion of the different factors in weight coefficients. Regard the total of the weight coefficients as 10, in order to balance the various factors into account so they cannot share that much difference. The effect of the cp account for 40%, the effect of the weekly hour account for 30% and the effect of the capacity account for 30%.The coefficient 4 presents the proportion of the course priority, coefficient 3 presents the proportion of the weekly hour. Because the capacity is bigger than other factors, it will be reduce the times. From the formula we can see that course priority have the greatest impact on the priority, the effect of the weekly hour is lower than it, and capacity is lower than the weekly hour. The value of the course is large so the priority is high. So the priority of each course will emerge big gaps, then we can use the quick sort to sort the course follow by the order of the priority.

C. The Design of the Timeslots Level Table

According to the effect of class, timeslots will be divided into several levels. This method can let algorithm become more efficient, so different weekly hour will have it own timeslots level table .when design the timeslots level table, the following principles should be satisfied:

- a). One student should not have more than one class one day.
- b). One student should have a class at morning.
- c). One student should have a class on next day.
- d). More lessons in one day will not have a good effect.
- e). Afternoon does not suit for teaching. As shown in Table 2, the algorithm use an integer as the code for the time level.

TABLE II. TIMESLOTS LEVEL TABLE

weekhour	priority	Mon	Tues	Wed	Thurs	Fri
2	1	11			41	
2	2	11			42	
4	1	11		31		51

D. The Design of the Provisional Matrix

Priority can be computed by the formula. When the priority of all courses has been calculated, the order of acquire resource will be determined. When classroom and time are in the allocation of resources, the two matrices were used: Class - Time Matrix and classroom - time matrix. These will be used to mark the allocated resource information. The two matrices play a role in isolating the conflict, when the resource has been distributed. These two matrices are shown as Figure 3 and 4. In the matrix, 0 represents occupied, 1 represents unoccupied.

Figure 3. Class - Time Matrix

Figure 4. Classroom - Time Matrix

IV. CASE STUDY

In order to confirm the feasibility of the algorithm, we can test the algorithm by import data and operate the algorithm. First, import the teaching plan in the interface. The teaching plan is shown as Figure 5.

CourseID	Semester	TeacherID	AcademicHour	WeekHour	type	Capacity	ClassID
200901	1	1001	32	2	1	60	701
200901	1	2003	60	4	2	28	708
200905	1	1002	64	4	3	60	701

Figure 5. The teaching plan.

Then, operate the algorithm in the course timetabling interface, shown as Figure 6.

Figure 6. Course timetabling interface.

After the operation, we can inquire the curriculum schedule successfully in the inquire interface, shown as Figure 7 and Figure 8.

Figure 7. Teacher table.

Figure 8. Classroom table.

V. CONCLUSION

Course timetabling problem is a very difficult problem. The problem is converted to the classroom, time and classes. This paper presents a simple algorithm, which is a priority of algorithm based on a number of factors to solve the problem. Test results show that using this algorithm can simplify the arranging algorithm and reduce the time complexity of problem-solving. In the process of the course timetabling, the system can give a satisfactory results.

REFERENCES

[1] Safaai D, Sigeru O. Incorporating constraint propagation in genetic for universitytimetable problem[J]. Engineering Application of Artificial Intelligence, 1999,241-253.

[2] "An Average Case Approximation Bound for Course Scheduling by Greedy Bipartite Matching" Gary Lewandowski1, Prakash Ojha2, Jennifer Rizzo1, and Abigail Walker1 Xavier University, Mathematics and Computer Science Department.

- [3] Pot hen A , Simon H , Liou K2P. Partitioning sparse matrices with eigenvectors of graph [J] . SIAM J Matrix Anal Appl , 1990 , 11(3) : 430-452.
- [4] Hartuv E. and R. Shamir , A Clustering algorithm based on graph connectivity [J] , Information Processing Letters , vol . 76 , nos. 4-6 , 2000 : 175-181.
- [5] S. Even, A. Itai, A. Shamir. On The complexity of timetable and multicommodity flow problems . SIAM Journal on Computing , 1976, 5(4) : 691-703.
- [6] Jennings N R, Wooldridge M , Agent-oriented software engineering. J. Bradshaw(ed) Hand book of Agent Technology, AAAI / MIT Press, 2001.