The Application of Graph Theory and Uniform Design Algorithm in the Test Arrangement of University

Mengqi Zhang[1, a], Yuejie Geng[2, b], Zhigang Zhang[3, c]

[1, 3] School of Mathematics and Physics, USTB, Beijing 10083, China

[2] Academic Affairs Office, USTB, Beijing 100083, China

[a]zhangmengqigood@163.com, [b]gyj@admin.ustb.edu.cn, [c]zzgcyf@263.net

Key words: test arrangement; graph theory; uniform design

Abstract: Test arrangement problem is a kind of NP hard problems, which has been proved. It's a combination of constraints, nonlinear, and fuzzy multi-objective optimization. The results can be found based on all kinds of known constraints, but it spends too much time. And the effect of the test is not certain, too. Above all, this article combines two methods in solving this problem, and proposes a reasonable solution. This solution also improves effective of test arrangement, and illustrates the rationality.

1 Description of test arrangement

The problem of test arrangement involves 5 elements represented as sets, classes of exams' set $C=\{c_1,c_2,\cdots,c_m\}$; courses of exams' set $L=\{l_1,l_2,\cdots,l_n\}$, test times' set $T=\{t_1,t_2,\cdots,t_x\}$, the invigilators' set $P=\{p_1,p_2,\cdots,p_y\}$, test locations' set $P=\{p_1,p_2,\cdots,p_y\}$ [1].

This problem also has hard constraint conditions and soft constraint conditions.

Hard constraint conditions include those:

I. An examination course must be arranged in a time period of examination;

II. All classes of one course must be arranged in the same time period of examination;

III. Every class must have one test course in one period of time;

IV. Every location must have one test course in one period of time.

Soft constraint conditions include those:

I. The arrangement of every class could be regular as far as possible so that the students could review their lessons more effective;

II. In order to use the time periods and classrooms, the number of the students who take the test for every day could be more regular;

III. The exam table of every student could be precisely, to make sure that the students who retake a course could not have a conflict.

In this article, it doesn't consider the effect of the invigilators and the test locations. So the hard constraint conditions are reduced to I~III.

2 A Survey of the Algorithm Design

2.1 A Graph Coloring Algorithm

Given an undirected graph $G=(V,E)$, where V is a set of vertices, E is a set of edges. The graph coloring problem is to divide the V into K sets of colors. Every set is single. In the problem of test arrangement, it's available to use this method to make connections with the test courses [2].

The process of test-time arrangement is as follows. Firstly, choose the largest degree in the first coloring group, for example l_{n_1} and arrange it in Time 1, then delete the edges which connect with l_{n_1}.The degrees of vertices which is connected with l_{n_1} all minus one. Secondly, choose the class c_{m1} which has the course l_{n_1} and it have the largest number of test courses. Then we pick up the courses, the degrees and the number of the courses that c_{m1} must take, and calculate the time difference if the courses are arranged average. In the courses that c_{m1} must take, choose l_{n_2} which in the first coloring group to arrange it. Thirdly, we delete the edges which connect with l_{n_2}, the degrees of vertices which are connected with l_{n_1} all minus one. We choose the class c_{m2} which has the course l_{n_2} and it have the largest number of test courses. Then we pick up the courses, the degrees and the number of the courses that c_{m2} must take, and calculate the time difference if the courses are arranged average. If c_{m2} have other test courses arranged, then we could put l_{n_2} who has the largest degree in the courses that c_{m2} must have in the time period of the sum of the time who has already be arranged and the time difference. And so on, until the last exam [3] [4].

2.2 Uniform Design Algorithm

Uniform design is an experimental design method, called uniform design test method. It is a kind of experimental design methods that considers the test points in the test range can be uniform spread. It was proposed by Professor Fang Kaitai and Mathematician Wang Yuan [5] [6].

The process of test-time arrangement using uniform design algorithm is as follows. Firstly, initialize the exam courses in a certain order. Secondly, generate the corresponding uniform design table so that we can make a test courses' sequence. Thirdly, make a three dimensional vector set of classes, courses and times. Choose the maximum norm of vector in the set, arrange the course. If the norms are equal, according to the curriculum - class - time sequence to arrange the examination. Then delete the course in the test courses' sequence. And so on, until the last exam [7].

3. Application of the algorithm in practical problems

Typically, the freshman and the sophomore have the most courses, so we select the appropriate proportion of courses to make experiments. For example, we choose 20 courses randomly, in these courses, 8 of them are for the freshmen, 6 of them are for the sophomore, 4 of them are for the juniors and the rest of them are for the graduates. We choose 332 classes in corresponding. Then we numbered the courses $L_1 \sim L_{20}$, the classes $C_1 \sim C_{332}$, and the time periods $T_1 \sim T_{20}$, in these periods, the multiple of 4 don't arrange any tests, which are only as a tool to determine whether it's a reasonable exam-arrangement.

3.1 Graph coloring algorithm for examination arrangement

Firstly, making the graph of the test courses.

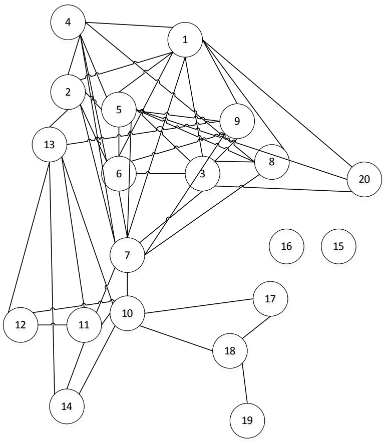

Fig1. The undirected graph of test courses

In the figure, we can find that course 7's degree was the maximum, so we color course 7 the first color, so as course number 1, 2, 3,4,5,6,8,9,10,11,20. In the rest of the courses, we pick course 13 to color the second color, so as course number 12 and 14. Then color courses 17, 18, 19 with the third color. In the end, because the courses 15 and 16 have the degree of zero, we could mark them in two colors.

Using graph algorithm, the arrangement is shown in Table1.

Table1. The arrangement using graph coloring algorithm

Weeks test screening	1	2	3	4	5
1	7，15	5	6,12	1	10
2	13,16	9,17	8	11，18	3,4
3	14	2	20	19	18

3.2 Test arrangement with uniform algorithm

Firstly, we establish the uniform design table shown in Table2.

Table2. The Unicode table using uniform design algorithm

Serial number	**Courses**	**Class number**
1	1	2
2	2	4
3	3	6
4	4	8
5	5	10
6	6	12
7	7	14
8	8	16
9	9	18
10	10	20
11	11	1
12	12	3
13	13	5
14	14	7
15	15	9
16	16	11
17	17	13
18	18	15
19	19	17
20	20	19

As shown in Table 2, we also define that all classes which have the test course *i as b(i),* and if we sort the test courses in grades, the arrangement is shown in Table3. If we sort the courses in the number of degree using graph color method, the arrangement is shown in Table4.

Table3. The arrangement using uniform design algorithm (1)

weeks test screening	1	2	3	4	5
1	20,10	9,14	7,13	12	11
2	19,8	18,4	17	6	5
3	15,2		3	16	1

Table4. The arrangement using uniform design algorithm (2)

weeks test screening	1	2	3	4	5
1	7,16	1	6,19	5	2,8
2	18,15	10	11	13	9,12
3	4	3	17	20	14

3.3 The establishment of evaluation’s function

To determine whether the students' examination arrangement is reasonable or not, we can establish a function. For a fixed student, we can count total number of the tests n_j, if the arrangement is uniformly dispersed, then the time interval will be $\Delta s_j = [\frac{20}{n_j}]$. Then we arrange the courses in chronological order and calculate the distance between the two courses as follows $\Delta d_{j1}, \Delta d_{j2}, \cdots, \Delta d_{jn_j-1}$. Then we calculate $\sum_{j=1}^{r}\sum_{i=1}^{n_j-1}(\left|4-\Delta d_{ji}\right|)$ as a reasonable basis judge of the students' examination arrangement.

To determine whether the courses' arrangement is reasonable or not, we can establish a function as well. Calculate the number of people who take the exam as *m*. Define a course that involves the largest number of students as $c_{\max}$. If the arrangement is uniformly dispersed, then $\Delta m = [\frac{m}{5}]\text{-}\, m_{c_{\max}}$ could be the sign. Then for every period, calculate $\Delta m_{t_q} = m_{t_q}\text{ - }\Delta m$ and $\sum_{t_q=1}^{10}\Delta m_{t_q}$.

Considering more on the students, the function of students' is given higher weights as follows: $0.7\times\sum_{j=1}^{r}\sum_{i=1}^{n_j-1}(\Delta s_j - \Delta d_{ji}) + 0.3\times\sum_{t_q=1}^{10}\Delta m_{t_q}$ .This is the final evaluation's function.Using it, we compare the results of the two algorithms and find out that the arrangement is more closing to the best answer using the uniform design algorithm.

3.4 Comparison of time complexity

In graph theory, every time we arrange a course, we must going through the course list. Then the time complexity is to $T(n) = n + (n\text{-}1) + \cdots 1 = \frac{n^2+n}{2}$. In uniform design algorithm, the dimension is 2, the number of points is *n*. Every time we arrange a course, we need to find the largest norm of the uniform design table, so the time complexity is to $T(n) = 2n + 2(n\text{-}1) + \cdots 1 = n^2 + n$.

4 Conclusions

Considering that, the test arrangement problem is a kind of NP hard problems, we could divide this problem into several parts. In this article, we focus on the time arrangement. First of all, classifying the courses using graph theory, then use the uniform design algorithm to arrange them.

However, if we have too many points in the uniform design algorithm, the amount of computation will be great. So we may use the result to be the initial solution in the intelligent algorithm to find better results.

Acknowledgement

Acknowledgements: This article is funded by the Beijing University of Science and Technology Education Research Project (JG2015M60).

References:

[1] Oscar Chavez-Bosquez, Pilar Pozos-Parra: Fundamenta Informaticae , Vol. 113 (2011), p. 1–18.

[2] Lili Wang: Historical development of graph theory . Shandong University, 2012.

[3] E. Burke, B. McCollum, A. Meisels, S. Petrovic, Rong Qu: European Journal of Operational Research, 2005, 1761.

[4] Jianxing Dong, Luan Yong, Junzheng Yan: Computer Systems & Applications, issue 05 (2011), p. 177- 179.

[5] Kaitai Fang: Mathematical Statistics and Management, issue 01 (1994), p. 57- 63, 56.

[6] Kaitai Fang: Mathematical Statistics and Management, issue 03 (1994), p. 52- 55.

[7] Bin Chen, Yang Liu: Journal of Harbin University of Science and Technology, issue 05 (2011), p. 106- 112.