Lecture and Tutorial Timetabling at a Tunisian University

Abdelaziz Dammak, Abdelkarim Elloumi, Hichem Kamoun

University of Sfax, FSEG, Route de l'Aerodrome, km 4, BP 1088, 3018 Sfax Tunisia
Department of Applied Quantitative Methods, GIAD Laboratory
{abdelaziz.dammak, abdelkarim.elloumi, hichem.kamoun}@fsegs.rnu.tn

1 Introduction

This paper deals with the Lecture and Tutorial Timetabling Problem at an institution in a Tunisian University. Our objective is to construct a feasible timetable for all lectures and tutorials taken by different groups of each subsection of any section in the institution. For this, we describe the timetabling problem of the institution considered and list all specific hard and soft constraints. We formulate the problem as a zero-one integer linear program in which we define two binary variables corresponding respectively to lectures and tutorials. The quadratic objective function proposed tries to eliminate a real problem of congestion leading to a waste of time and students' delays. Since the number of constraints is very large, the use of the heuristic procedures is of primary importance. We develop three heuristic procedures: first, we start by assigning all lectures of different student sections having the biggest size in a classroom with the smallest capacity that can fit the students. Second, we complete the output of the first phase by assigning the tutorials for different groups. Lectures and tutorials timetabling problem are correlated and cannot be treated independently if we desire to get a complete solution. The two first heuristics are illustrated with real data of one section at the Faculty of Economics and Management Sciences of Sfax in Tunisia and compared with those manually generated. Since there are several criteria which are preferably satisfied as much as possible, we will formulate the problem as a multiobjective mathematical program, and then we develop later a third heuristic in order to ameliorate the quality of the solution. The different criteria which can be taken into account are: minimize the number of free inter-meetings, maximize the professor preferences, minimize the distance covered by the students between the classrooms and exempt the students as much as possible in the half day.

Educational timetabling has been the subject of several papers in various scientific journals and the topic of many theses in different universities. This problem concerns essentially course and exam which are to be scheduled during the academic year. The course timetabling problem consists of scheduling a certain number of courses into a certain number of timeslots spread throughout the week in such away that hard and soft constraints are satisfied.

Various techniques have been used to solve Timetabling Problems (see Burke et al. [3], Carter and Laporte [8]). One of the earliest methods used to solve this problem is graph colouring in which vertices represent events and two vertices are connected if

E. K. Burke, H. Rudová (Eds.): PATAT 2006, pp. 384–390. ISBN 80-210-3726-1.

and only if there is a conflict. Welsh and Powell [18], Wood [19], Selim [16] and De Werra [12] proposed several formulations by graph colouring for a set of class-teacher timetabling problems and discussed the inherent complexity. Recently, Timothy [17] has used graph colouring to solve both course and exam timetabling.

Linear and integer programming models were frequently used to formulate the course time-tabling problem usually with binary variables (Diskalaki et al. [11], Diskalaki et Birbas (10] and Dimopoulou and Miliotis [13], [14]).

Burke and Petrovic [2] discuss some recent development in the field of automated timetabling. The discussion deals with both course and exam timetabling. Overviews of four types of approaches to timetabling problems that have been applied are given: sequential methods, cluster methods, constraint based methods and metaheuristic methods.

Another technique that has recently been successfully applied to course timetabling is Case-Based Reasoning (CBR). The origin of CBR dates back to 1977 with the work of Schank and Abelson [15]. CBR has also been well applied to scheduling and optimization problems.

Burke et al. ([1], [4], [5], [6]) were the first to adapt this approach to solve university timetabling problems. The main idea behind the use of CBR in timetabling is that previous timetabling problems and their appropriate solution procedures are stored in a knowledge base which is used to provide good solution for a new timetabling problem.

In the papers [4] and [6], the authors illustrate the use of attribute graphs to graphically represent a course timetabling problem. In this graph, the courses (events) are represented by nodes and the relationship that exists between these events (including hard and soft constraints) is indicated by edges. Then a similarity measure is used to indicate which part of the attribute graphs of the stored cases in the knowledge base has the most similar structure of the attribute graph of a new timetabling problem. Finally, the most appropriate solution procedure used for the selected stored case is adapted to solve the new problem.

In the paper [1], they keep using case-based reasoning approach for solving course timetabling problem but instead of using attribute graphs for constructing the knowledge base, a knowledge discovery process is performed based on a set of features that are judged to be most appropriate to describe the characteristics of the timetabling problem.

In the recent paper [5], they use the multiple-retrieved case-based reasoning approach to solve large scale timetabling problem which were until then unable to be solved by CBR in the earlier papers. The main idea is to decompose the attribute graph associated with the large timetabling problem into smaller attribute sub-graphs whose associated timetabling problem can be solved using CBR approach. Then the partial solutions are all combined to obtain a timetable for the large time- tabling problem.

In their article in press, Burke et al. [7] develop a graph-based hyper-heuristic (GHH) which has its own search space that operates in high level with the solution space of the problem generated by the so-called low level heuristics.

2 Problem description

The construction of course timetabling at the Faculty of Economics and Management Sciences of Sfax (FEMSS) is performed manually by administration staff twice in each academic year (first and second semester). There are thirty timeslots distributed along the six days of the week: Monday to Saturday. There are six timeslots in Monday, Tuesday, Thursday and Friday and only three timeslots in the morning in Wednesday and Saturday. Each timeslot has one hour and a half duration followed by fifteen minutes break except the third timeslot in the morning is followed by thirty minutes lunch break.

The lectures and tutorials are of two categories: there are some with only one period per week and others require two periods per week. Lectures with two periods cannot be held at the same day. There are lectures without tutorial, with only one period tutorial and with two-period tutorial. There are several sections divided into different subsections.

Each subsection with a big size is divided into a certain number of groups having a size no more than thirty students. The lectures are to be taught to a whole section or subsection while the tutorials are only taught to groups in small classrooms.

As any timetabling problem, there are both hard and soft constraints. The hard constraints are those that cannot be violated at any circumstances in order to obtain a feasible solution.

We consider these hard constraints:

- All courses (lectures and tutorials) included in the program of each section are insured.
- Any professor cannot teach more than one course at the same period.
- Any classroom cannot be used more than once in any period.
- Any group of any subsection of any section cannot be taught more than one course in any timeslot.
- Any subsection of any section cannot take two lectures in two consecutive timeslots.
- Courses with two periods cannot be taught twice in the same day.
- Any professor doses not teach three courses at three consecutive timeslots.
- Any group of any subsection of any section cannot be taught consequently in the third and fourth periods.
- Any professor cannot teach consequently in the third and fourth periods.

The soft constraints are restricted to:

- The time preferences of professors should be respected as much as possible.
- For any group of any section the rate of occupation of the seats should be maximized.

Other soft constraints can be considered:

- Minimize the number of free inter-meetings.
- Maximize the professor preferences.
- Exempt the student as far as possible in the half day.

3 Problem formulation

The course timetabling problem of the Tunisian institution was formulated as a zero-one linear integer program in which we define two binary variables corresponding respectively to lectures and tutorials. In this formulation, we have considered all hard constraints cited in section 2.

The objective function has a quadratic form in which we have considered a real problem of routing between classrooms and aims to minimize the distance covered by students between these classrooms.

4 Tutorials' Timetabling Heuristic (TTH)

This heuristic completes the one that has been developed by Dammak et al. [9] in which the authors solve the problem of lecture timetabling in the same institution. This new heuristic is composed of eight steps detailed as follow:

Step (1):

Arrange the set of sections in non-increasing order of the enrolled student size.

Arrange the classrooms in non-increasing order of their size.

Step (2):

For each group of each subsection of each section, we begin by assigning the first period of the two-period tutorial that needs to be taught to this group of the subsection.

Step (2.1):

We look for the first classroom which can hold the current tutorial and having the smallest size.

Step (2.2):

This classroom is assigned to this tutorial if and only if:

We find a period in which this classroom is available and at the same time the group of the subsection is free in the current period. If this current period is the third (respectively the fourth) in the day then the group has to be free the fourth (respectively in the third) period.

In case no such period exists, we check the availability of the immediately precedent classroom.

If there is no classroom available that can fit this tutorial, we have to divide the group into smaller groups.

Also we consider the availability and time preferences of professors that can teach this tutorial.

Step (2.3):

For a certain period, we check if the professor has taught in the two consecutive preceding periods or in the two consecutive following periods or in the two periods corresponding to the previous and the following periods.

Step (2.4):

If the current professor is busy or step (2.3) is satisfied then choose another professor.

Step (3):

We assign the one-period tutorials. We follow the same procedure used in step (2.1) to step (2.4) (respectively) denoted step (3.1) to step (3.4) (respectively).

Step (4):
We assign the second-period of the two-period tutorials that has to be taught by the same professor. We follow the same procedure of steps (2.1) and (2.2) (respectively) denoted steps (4.1) and (4.2) (respectively). In addition, we have to prevent the assignment of the second period tutorial during the same day in which the first period tutorial is scheduled.

5 Numerical Example

We restrict our numerical example on only one group chosen from the first subsection of the first section of the institution. We denote C_{ijk} the lecture k taught by the subsection j of section i, D_{ijt} the tutorial t taught by the subsection j of section i, s the classroom, and h the professor. The output of our heuristic is summarized in the timetable 1:

Timetable 1

Day / Hour	08 – 9:30	09:45-11:15	11:30-13:00	13:30-15:00	15:15-16:45	17:00-18:30
Monday	C_{111}, s = 3, h=1	D_{112}, s = 28, h=17		C_{113}, s = 3, h=5	D_{116}, s = 28, h=29	D_{114}, s = 28, h=6
Tuesday	C_{112}, s = 3, h=2	D_{118}, s = 28, h=36	C_{114}, s = 3, h=6		C_{115}, s = 3, h=7	D_{117}, s = 28, h=37
Wednesday	C_{111}, s = 3, h=1	D_{111}, s = 28, h=14				
Thursday	C_{112}, s = 3, h=2	D_{114}, s = 27, h=6		C_{113}, s = 3, h=5		D_{113}, s = 24, h=20
Friday	C_{116}, s = 3, h=9					
Saturday						

Timetable 2

Day / Hour	08 – 9:30	09:45-11:15	11:30-13:00	13:30-15:00	15:15-16:45	17:00-18:30
Monday			D_{112}, s =31, h=2	C_{115}, s = 2, h=7	D_{114}, s = 46, h=6	
Tuesday	D_{118}, s =35, h=36			C_{116}, s = 3, h=8	C_{113}, s = 2, h=5	
Wednesday	C_{114}, s = 2, h=6	C_{111}, s = 1, h=1				
Thursday	C_{112}, s = 4, h=2	D_{114}, s = 44, h=6		D_{117},s=36, h=33		
Friday	C_{112}, s = 3, h=2	C_{111}, s = 1, h=1	D_{116}, s =58, h=28		C_{113}, s = 3, h=5	D_{113}, s = 43, h=19
Saturday	D_{111}, s =52, h=12					

To illustrate the performance of our heuristic, we compare the results presented in the timetable one with those generated manually by the administration presented in the timetable two.

From these two timetables, we can draw the following remarks:

1. In the manual solution, the number of half days with single lesson is equal to 4. However, in the solution provided by the heuristic (TTH), there is only one half day with one meeting. It is preferable to schedule at least two courses per half day in order to prevent the student moving for only one meeting.
2. The advantage of the output of our heuristic consists of releasing students as far as possible during the weekend (morning of Friday). For this, the amelioration of the solution is easier to perform in the heuristic solution than in the manual one.
3. The constraint of excluding third and fourth timeslots in each of the four completes days is rigorously satisfied by heuristic solution but not considered by hand-made one.
4. Finally, this comparison is far from being definitive and conclusive since this work is considered partial for the following reasons. First, we considered only one group of one subsection of one section; a thorough test will include all sections. Second, data for at least three recent academic years need to be used in the test.

References

1. Burke E. K, MacCathy, B. Petrovic, S. and Qu, R. knowledge discovery in a hyper-heuristic for course timetabling using case-based reasoning, In Burke, E.K. and De Causmaeker, P. (editors), PATAT'02, LNCS 2740 (2002) 90-103
2. Burke, E.K. and Petrovic, S. Recent research directions in automated timetabling. European Journal of Operations Research, 14(2) (2002) 266-280
3. Burke, E.K., Jackson, K., Kingston, J. and Weare, R. automated university timetabling: the state of the art. The computer journal, 40(9) (1997) 565-571
4. Burke, E.K., MacCathy, B., Petrovic, S. and Qu, R. case-based reasoning in course timetabling: an attribute graph approach, In Proceedings of 4th International Conference on Case-Based reasoning (ICCBR 2001). LNAI 2080 (2001) 90-105
5. Burke, E.K., MacCathy, B., Petrovic, S. and Qu, R. multiple-retrieval case-based reasoning for course timetabling problems. Journal of the Operational Research Society (2005) 1-15
6. Burke, E.K., MacCathy, B., Petrovic, S. and Qu, R. structured case in case-based reasoning-re-using and adapting cases for timetabling problems. Knowledge-Based Systems. 13 (2000) 159-165
7. Burke, E.K., McCollum, B., Meisels, A., Petrovic, S. and Qu, R. A graph-based hyper-heuristic for educational timetabling problem. European Journal of Operational Research, article in press (2006) 1-16
8. Carter, M.W. and Laporte, G. recent developments in practical course timetabling, In Burke, E. K. and Carter M. W. (editors) PATAT'97, LNCS 1408 (1997) 3-19
9. Dammak, A., Elloumi, A. and Kamoun, H. Lecture timetabling at a Tunisian university. Technical report No 4, GIAD Laboratory (2006) 1-25
10. Daskalaki, S. and Birbas, T. Efficient solutions for university timetabling problem through integer programming. European Journal of Operational Research. 160 (2005) 106-120

11. Daskalaki, S., Birbas, T. and Housos, E. An integer programming formulation for a case study in university timetabling. European Journal of Operational Research. 153 (2004) 117-135
12. De Werra D. Some combinatorial models for course scheduling, In Burke, E. K. and Ross, P. (editors) PATAT'95, LNCS 1408 (1995) 296-308
13. Dimopoulou, M. and Miliotis, P. An Automated Course Timetabling System developed in a distributed Environment: a Case Study. European Journal of Operational Research. 153 (2004) 136-147
14. Dimopoulou, M. and Miliotis, P. Implementation of a University Course and Examination Timetabling System. European Journal of Operational Research, 130 (2001) 202-213
15. Schank, R.C. and Abelson, R.P. Scripts, plans, goals and understanding. Erlbaum, Hillsdale, New Jersey, US (1977)
16. Selim, S.M. Split Vertices in Vertex colouring and their application in developping a solution to the faculty timetable problem. The Computer Journal, 31(1) (1988) 76-82
17. Timothy, A.R. A Study of university timetabling that blends graph coloring with the satisfaction of various essential and preferential conditions, Ph.D. Thesis. Rice University (2004)
18. Welsh, D. J. A. and Powell, M. B. An upper bound for the chromatic number of graph and its application to timetabling problems. The Computer Journal, 10 (1) (1967) 360-364
19. Wood, D. C. A technique for colouring a graph applicable to large scale timetabling problems. The Computer Journal, 12 (4) (1969) 317-319