Lecture Notes in Management Science (2013) Vol. **5**: 25–32
5th International Conference on Applied Operational Research, Proceedings

ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

Modeling and solving a timetabling problem considering time windows and consecutive periods

Diana Sánchez-Partida, José Luis Martínez-Flores and Elias Olivares-Benítez

UPAEP University, Puebla, México
diana.sanchez@upaep.mx; joseluis.martinez01@upaep.mx; elias.olivares@upaep.mx

Abstract. This paper presents a case study called UPAEP Timetabling. This problem arises in the allocation design of some or all of professors-courses-rooms-timeslots-groups variables, considering factors such as availability and capacity rooms. In the reviewed papers a general model that covers the requirements of any Institution have not been found due to operational rules set are determinants for constraints modeled; therefore normally a mathematical model is designed for each Institution. The proposed model is considered one of the most complete due to the kind of considerations in the moment of building the constraints. The model was validated at UPAEP University in the graduate education area, attempting to tackle a real-world problem, considering 85,223 variables, time windows of the professors, periods consecutive and capacity rooms, between other constraints. This document presents a mathematical model solved with commercial optimization software, which helped to found successfully the solution of the Timetabling Problem.

Keywords: integer programming; timetabling; combinatorial optimization; time windows

Introduction

The Timetabling Problem (TP) in Educational Institutions, has been studied extensively in the literature (Avella et al., 2007), due to the great importance of the consumption of time and human resources for their solution. The problem is to create an assignment schedule for courses-professors and rooms, trying to satisfy at the best possible conditions and requirements of institutional policies. The universities, in the search to solve this problem, commonly performed manual procedures or spreadsheet support, which is often tedious because it is a time consuming task, requiring several days to find an acceptable solution. The difficulty to find the solution to the problem is mainly due to its combinatorial nature, if one takes a number of "large" variables are difficult to solve exactly. There is no a general classification for this problem, due to different operational rules issued by each institution (Crovo et al., 2007). Consequently there is no a general model that covers each of the requirements of any institution, so it is necessary to design a special mathematical model that satisfies the requirements of each institution.

There are sub-problems such as scheduling course-professor and classroom assignment Crovo et al. (2007). Shaerf in (1999) mentions a third classification called scheduling exams (Tesfaldet, 2008). In the literature review several studies have been found, where some models are simulated based on assumptions and others are applications in the real-word. For instance Schimmelpfeng et al. (2007), using Integer Programming-Mixed, found the solution to the scheduling of timetable at a university in Germany at the School of Economics and Management, assigning 156 courses, 181 groups, 99 professors, 30 timeslots per week, 13 rooms with different capacities minimizing the violation of restrictions.

Papoutsis et al. (2003) implemented its integer programming model at a high school in Greece assigning 6 courses, 15 professors and 35 periods a week, minimizing the sum of the penalties. Another integer programming model is implemented in a developed secondary Birbas, Daskalaki and Housos (2009). The size of their instance was 95 courses, 12 groups, 23 professors, 35 weekly periods considering multi-periods, applied in two stages minimizing cost allocation of course-professor and professor-group coefficients.

Other solution methods used in this kind of problem are, the partitioning method supported in the paper (Sarin et al. 2010), the graph coloring (Burke et al. 2010), constraints programming approach see (Rudova, Müller and Murray, 2011), heuristic (Al-Yakoob et al. , 2010; Bouffard et al., 2007; Arabinda, 1984), goal-integer programming and genetic heuristic algorithm in Mirrazavi, et al (2003), Meta-heuristic used by Avella et al. (2007), artificial intelligence used by Schaerf (Tesfaldet, 2008).

This paper presents one of the most complete models among the literature reviewed, due to the kind of considerations Modeled in the constraints and the size of instance validated.

Framework

The remainder of this paper is organized as follows: Section 2 presents the problem description and mathematical model. In Section 3, we examine the computing experience and the results obtained by solving the aforementioned model. Finally, based on the above findings, the conclusions and proposals for future work are developed in Section 4.

Model and general description of the problem

Description of the UPAEP Case Study

The case study UPAEP is developed in order to create an Automated Timetabling to reduce the high costs in time and labor effort in developing it. The case study was the reproduction of the courses offered by the Graduate education area to the students in the spring 2013 period, with the purpose of generating a general model that can satisfy the constraints formulated based on institutional policies, and thus compare results with those generated by the department responsible for this work.

Masters and PhD programs offered quarterly have 14 Coordinators belonging to one of the five schools, each of these Coordinators organize their courses under three different modes face-to-face, video and mixed, this information is sent to the department attendant to prepare the schedules. In order to perform this scheduling, the following considerations are taken into account:

1. There are 89 courses in the total of the three modes.
2. There are 56 courses under the face-to-face modality (requiring normal room capacity).
3. There are 23 courses under the modality face-to-face and mixed (requiring virtual room capacity).
4. There are 10 courses in the videoconference modality (requiring virtual room regardless capacity).
5. There are two courses that must be taught from 8:00 am - 14:00 pm (6 consecutive hours) on Saturday.
6. There are three courses that must be taught from 16:00 pm - 19:00 pm (3 consecutive hours) one course on Tuesday, another on Wednesday and the third on Friday.
7. There are two courses that must be taught from 17:00 pm - 22:00 pm (5 consecutive hours) one course on Thursday and another course on Friday.
8. There are two courses that must be taught from 19:00 pm - 22:00 pm (3 consecutive hours) on Monday.
9. The rest of the courses should be taught from 19:00 pm - 22:00 pm (3 consecutive hours) on any day of the week, or 8:00 am - 11:00 am or 11:00 am - 14:00 pm on Saturday.
10. There are 13 courses that must be taught in specialized rooms called special rooms at building E.
11. There are 31 courses that must be taught in laboratories at building F.
12. There are 45 courses should be taught in classrooms without special requirements called normal rooms at building F.
13. There are 8 rooms considered normal, 5 virtual at building F and 13 especial rooms at building E.
14. There are professors who teach more than one course, so courses should not be taught in the same period of time.
15. There are two professors who do not have an open schedule to teach their class, so that an available time windows should be considered for them

Therefore this research when making scheduling considers aspects such as capacity and special equipment for rooms, modality and number of students enrolled in each course, number of courses taught by the professor, the time windows available by professor and special requests schedules of some courses by program. The objective seeks the maximization of the assignment of courses to rooms, generating appropriate schedules required by the members of the academic units (Coordinators, professors and students), and so in this way have more effective response avoiding the overlapping problem commonly found in this type of timetabling.

Mathematical model for the UPAEP case study

Definition of sets

I = Set of courses that belong to one modality and school, which previously have assigned a professor {1...*I*}.

J = Set of normal rooms, special rooms and laboratories belonging at building E and F {1...*J*}.

K = Set of total weekly periods {1...*K*}.

Hence the construction of the model using the sets above declared, results in 83, 304 decision variables. Trying to optimize but not losing sight of the problem, subsets are declared instead of the sets *I, J, K,* these are defined below:

Isub_modo Sets of modalities. Face-to-face, video or mixed regardless of the school to which they belong.

Isub_cc Sets of courses that require specific hours per week applied by the Academic Program Coordinator.

Isub_cat Sets of courses taught by the professors, no matter the modality and the school to which they belong.

Jsub_req Sets of rooms classified by their properties as normal, special and laboratory buildings belonging at buildings F and E where can be accommodated courses.

Ksub_dia Sets of *k* periods for each day of the week in which can be scheduling the courses.

Ksub_vtc Sets of *k* periods where professors cannot teach courses.

h_i Weekly hours requested by the Academic Program Coordinator.

h_m_i Daily hours requested in consecutive periods or multi-periods by the Academic Program Coordinator.

Decision variables of the model

The model is built on a set of decision variables defined below:

∀ *i* ∈ *Isub_modo, Isub_cc, Isub_cat;* ∀ *j* ∈ *Jsub_req;* ∀ *k* ∈ *Ksub_dia, Ksub_vtc*

Binary variable

X_{ijk} = 1; if the course *i* (=1,…,*Isub_modo, Isub_cc, Isub_cat*) is scheduled in the room *j* (=1,…, *Jsub_req*) in any timeslot *k* (=1,…, *Ksub_dia, Ksub_vtc*).

= 0, otherwise.

Objective function

The Objective function is formulated to maximize the assignment of courses in the scheduled hours and rooms.

$$MAX = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} x_{ijk}$$

Model constraints

Once have defined institutional policies for timetabling, we can now develop these in mathematical form.

Number of hours per week required of the different courses.

$\forall i \in Isub\_cc$

$$\sum_{j \in J} \sum_{k \in K} x_{ijk} = h_i$$

This restriction is classified as hard because it ensures the allocation of the exact amount of hours required per course.

Courses that require certain number of hours in consecutive periods.

$\forall i \in Isub\_cc$

$$\sum_{j \in J} \sum_{k \in Ksub\_dia1} x_{ijk} \leq hm_i$$

The mathematical representation above is classified as soft because it satisfies the number of consecutive periods requested by the Academic Program Coordinator only where there are periods available.

Courses that require mandatory number of hours in consecutive form or multi-period.

$\forall i \in Isub\_cc$

$$\sum_{j \in J} \sum_{k \in Ksub\_dia2} x_{ijk} = hm_i$$

This constraint is advised to take few courses in inflexible schedule, to avoid the infeasible program.

Room Capacity. The following constraint helps to find a room with capacity for each of the courses, according to the number of students enrolled.

$\forall i \in I,\ \forall j \in J,\ \forall k \in K$

$$a * x_{ijk} \leq b_j$$

Room and overlap. Each room can hold at most one course for each timeslot.

$\forall j \in J,\ \forall k \in K$

$$\sum_{i \in I} x_{ijk} \leq 1$$

Professor and overlap. When the professor gives more than one course, those should not overlap.

$\forall j \in J,\ \forall\ k \in K$

$$\sum_{i \in Isub\_cat} x_{ijk} \leq 1$$

Professor and Time Windows. Periods where the professors should not teach their course or courses.

$\forall i \in Isub\_cat,\ \forall k \in Ksub\_vtc$

$$\sum_{j \in J} x_{ijk} = 0$$

Rooms not required. Set of rooms not required to teach certain courses.

$\forall i \in Isub\_modo$

$$\sum_{j \in Jsub\_req} \sum_{k \in K} x_{ijk} = 0$$

Rooms required. Selection of rooms to teach certain courses.

$\forall j \in Jsub\_req,\quad \forall k \in K,$

$$\sum_{i \in Isub\_modo} x_{ijk} \leq 1$$

Once the model was obtained, it was solved with the optimization software to analyze the results.

Computational experience and results

Results of UPAEP case study

This instance includes the quarterly courses offered by the different schools of the UPAEP. The instance was reproduced corresponding to spring 2013, considering 89 courses with specific requirements, 26 rooms with capacities and specialized equipment and 36 periods a week, taking into account the time windows of the professors.

The optimization software called Lingo 13 unlimited version was used, it was installed on a workstation with 4.00 GB RAM, hard drive total size of 1397 GB and Intel (R) Core (TM) i7-3770 3.40GHz CPU processor. The solution method used was the default Branch and Bound solving the problem classified as an Integer Linear Programming model, using 30,580 K memory and generating85, 223 constraints.

The results were globally optimal with 277 decision variables and a solution time of 2.2 seconds. Current data of the University was used to generate 2 instances more in order to assess the quality of the model. Afterwards the results obtained were confirmed satisfactory. In the Table 1 the results are shown.

Table 1. Characteristics of the instances

#	*Courses*	*Selected variables*	*# Decision variables*	*Constraints*	*Iterations*	*Computational time*	*Solution*
1	89	277	83304	85223	14787	2.2 SEC	GLOBAL
2	108	347	101088	103116	17732	3.1 SEC	GLOBAL
3	126	395	116064	118204	24639	3.4 SEC	GLOBAL

The results meet the capacity room constraints, consecutive periods, and hours required per course, specific times and special rooms requested by the Coordinators among other constraints that prevent overlap of courses taught by the same professor respecting their available time windows, see Table 2.

Table 2. Partial list of courses of professor 1

Professor_1 Time Window available (any timeslot of T, W, TH, F, S)			
Course	*Assigned room*	*Assigned day*	*Assigned hour*
MIXED MODE_7	Laboratory 2	Wednesday	16:00 - 19:00
MIXED MODE_8	Laboratory 2	Saturday	8:00 - 11:00
FACE-TO-FACE MODE_15	Room 4	Tuesday	19:00 - 22:00

Conclusions and future work

Finally the conclusions after everything was analyzed, tested and registered, is that the University UPAEP has equipment able to schedule the totality of the quarterly courses. The model presented is one of the most complete models among the literature reviewed, due to the type of considerations modeled in the constraints and the size of the validated instance.

The model proposed in this paper satisfies all operational rules of the institution, and contribute well in the efficiency of the development of timetabling. So in this way one of the future works is to implement this model in UPAEP University. Moreover other future work is to continue the gradual inclusion of scheduling of approximately 4,400 courses in each of the Schools Bachelor level, 83 periods and over 200 rooms. This implies the inclusion of the group constraint and thereby achieving the timetabling integrally. Probably, in a larger or more constrained instance, the timetabling cannot fulfill all constraints. Therefore in a future work the model could be extended with soft constraints and penalties to handle those cases.

References

Al-Yakoob SM, Sherali HD and Al-Jazzaf M (2010). A mixed-integer mathematical modeling approach to exam timetabling. Comput ManagSci 7:19–46

Arabinda T (1984).School timetabling case in large binary integer linear programming. Management Science30:12, 1473-1490

Avella P, D'Auria B, Salerno S andVasil'ev I (2007). A computational study of local search algorithms for Italian high-school timetabling. J Heuristics13:543–556

Birbas T, Daskalaki S and Housos E (2009). School timetabling for quality student and teacher schedules. J Sched12: 177-197

Bouffard V, Ferland A and Jacques A (2007). Improving simulated annealing with variable neighborhood search to solve the resource-constrained scheduling problem. J Sched10: 375-386

Burke EK, Mareˇcek J, Parkes AJ and Rudová H (2010). A supernodal formulation of vertex colouring with applications in course timetabling.Ann Oper Res179: 105–130

C Sarin S, WangY and Varadarajan A (2010). A university-timetabling problem and its solution using Benders' partitioning—a case study. J Sched13:131–141

Crovo AS, Martín COS and Rojas LP (2007). Modelos de Programación entera para un problema de programación de horarios. Ingeniare: Revista Chilena de Ingeniería 15: 3, 245-259

Mirrazavi SK, Mardle SJ and Tamiz M (2003). A two-phase multiple objective approach to university timetabling utilizing optimisation and evolutionary solution methodologies. Journal of the Operational Research Society54:1155–1166

Papoutsis K, Valouxis C and Housos E (2003). A column generation approach for the timetabling problem of Greek high schools. Journal of the Operational Research Society54: 230–238

Rhydian L (2008). A survey of metaheuristic-based techniques for University Timetabling problems. OR Spectrum 30: 167–190

Rudová H, Müller T and Murray K (2011). Complex university course timetabling. J Sched14: 187–207

Schimmelpfeng K and Helber S (2007). Application of a real-world university-course timetabling model solved by integer programming. OR Spectrum29:7, 83–803

Schaerf A (1999). A Survey of Automated Timetabling. Artificial Intelligence Review 13:87–127

Tesfaldet BT (2008).Automated lecture timetabling using a memetic algorithm. Asia - Pacific Journal of Operational Research25: 4, 451-475.