Binary integer programming model for university courses timetabling: a case study

Abel Borges (Aluno)
Universidade Federal de Pernambuco
apdmbj1@de.ufpe.br

Prof. Raydonal Ospina (Co-orientador)
Universidade Federal de Pernambuco
raydonal@de.ufpe.br

Prof. Geiza Cristina (Co-orientadora)
Universidade Federal de Pernambuco
geiza@de.ufpe.br

Prof. André Leite (Orientador)
Universidade Federal de Pernambuco
leite@de.ufpe.br

ABSTRACT

It is proposed a binary integer programming model to handle a real instance of the courses-to-professors timetabling problem. The optimization criterion is the preferences of professors by courses and schedules. Besides the common set of constraints, specific features of this situation are considered. To decide about different optimal solutions, it is introduced a coefficient to measure (alternatively and independently of the approach used in the model) how well some solution met the preferences of professors by courses. The model was successfully applied twice, in semesters 2014.2 and 2015.1, by the Departamento de Estatística (DE) of the Universidade Federal de Pernambuco (UFPE). This paper is the result of an undergraduate research mentorship of the first author.

KEYWORDS. Integer programming, university timetabling, courses scheduling.

Main Area: OC - Combinatorial Optimization, EDU - OR in Education, OA - Other applications in OR

1. Introduction

A classic problem which appears often in literature of combinatorial optimization is the *matching problem* Papadimitriou and Steiglitz [1982]. It has been proved to be NP-complete (Even et al. [1975]), *i.e.* for which there is no polynomial time algorithm. In a particular situation, it is known as the *timetabling problem*, *cf.* de Werra [1985].

A huge part of the papers on timetabling are motivated by scheduling issues in universities or schools PATAT [2014]. Thus, it is not unusual in literature to treat the participants involved in the timetabling problem more particularly, as it is done here. Our problem may be defined and more well comprehended in the following terms.

The timetabling problem looks for the best schedule, according to some criteria, that indexes in time every element in a set of *resources*, which may countain professors, groups of students, classrooms or laboratories (or arrays combining these elements). Such time intervals usually have pre-defined structures that compose a set. A set of constraints defines the terms of *availability* of the different components, so determining the schedule rules, that is, how the resources must be allocated.

It is possible to distinguish two main classes of *educational* timetabling problems: exams timetabling and courses timetabling. A couple of fundamental differences between them are immediate, as pointed out by Burke et al. [1964].

Firstly, exams must be scheduled in such a way that no student has more than one exam at the same time, but it is possible to assign time periods to courses with no previous knowledge about the students subscription. On the other hand, if space is a limited resource, it is allowed when scheduling exams that more than one class of students share the same room, whereas that is an obvious constraint in courses timetabling.

The reader is referred to de Werra [1985], in which the author considers general formulations for the timetabling problem, starting from the simpler or less specific one and hence including common constraints in practical applications. Also, he solves this problem using an approach based on graph coloring methods. Despite these generalization difficulties (or maybe because of them), there is a wide scientific production on timetabling which deals with its theory and applications and several approaches are proposed (*cf.* Section 2).

We present a case study of this kind of problem, particularly courses timetabling. They are observed the characteristics of the Departamento de Estatística (DE) of the Universidade Federal de Pernambuco (UFPE, `www.de.ufpe.br`). Educational questions must be satisfied and we try to answer the professors' subjective preferences for the courses and schedules (weekly, in this case). So far, this process has been realized manually, taking some weeks until a conclusion. We attempt to promote its automation by implementing an optimization model that looks for the best schedule, in the sense to be explained.

2. State of the art

The main practical motivation of this research field could be regarded to the impracticability to solve the problem manually as it increases in size. The massive use of computers to solve timetabling problems probably started with Gotlieb's *The construction of class-teacher timetables*, in 1963 Gotlieb [1963].

A survey conducted by the Automated Scheduling and Planning (ASAP) Group at the University of Nottingham in the year of 1995 obtained feedbacks of 56 british universities on the use of computers to build timetables Burke et al. [1964]. Then, 42% of them were used to schedule manually, 37% assisted by computers and 21% totally automated.

The timetabling problem became more popular after the International Timetabling Competition (ITC), which have had three versions (2002, 2007 and 2011), *cf.* Post et al. [2013]. Such events positively impact the research community in the sense of stating common instances and so enabling comparisons between the models and algorithms proposed.

The binary integer linear programming model is a widely applied approach to this problem (see, for example, Bakir and Aksop [2008], Ferreira et al. [2011], Havas et al. [2013]). Whenever there exists a solution, the optimal algorithm available will find it. The downside appears very often: when the instance becomes relatively large, exact algorithms and the respective based models are very time consuming and are not so desirable.

Heuristics and meta-heuristics based methods are alternatives. In fact, proposals in these directions involve integer and mixed integer mathematical programs. Actually, which has shown to be a really powerful approach is combine the good features of both exact and heuristics methods as done in Ahmed et al. [2015], Burke et al. [2010], Santos [2007].

3. Scheduling rules

Firstly, the model is based on the satisfiability approach of optimize the professors' preferences by courses and schedules. This paradigm guides the objective function. Secondly, as usual when building courses timetables, we consider common scheduling rules. For example, only one professor teaches each class; professors just may be in one place at a time; each course must be taught by the same professor; professors have a maximum load of courses to teach; try to maximize number of graduating students.

Additionally, we consider specific features. For instance, we need to deal with basic and external courses, which have a prefigured timetable by other departments, and manage the choice of courses to offer by semester. Also, classes of a same course should be properly spaced over the weekdays and so on.

4. The mathematical model

We now describe the structure of the proposed model. The particular contents and details of the computational implementation are not exhaustive. The notations present in Table 1 are considered, calling this sets and explaining as it is convenient in the text. Furthermore, variables and parameters are defined as follows.

Table 1: Description of sets and indexes used in the model.

Set	Index	Description
$\mathcal{T}, \mathcal{T}_d, \mathcal{T}_s$	t	Professors, department professors and assistants ones
$\mathcal{C}, \mathcal{C}_{\text{und}}, \mathcal{C}_{\text{ext}}$	c	All courses, undergraduate courses and external ones
$\mathcal{D}$	d	Weekdays
$\mathcal{S}$	s	Shifts: morning, afternoon, night
$\mathcal{B} \equiv \{1, 2\}$	b	First and second time blocks (or time slots) in a given shift
$\mathcal{P}, \mathcal{P}_b$	p	All semesters and semesters in which are offered *basic* courses
$\mathcal{C}_p$	c	Courses distinguished by semesters
$\mathcal{N}$	n	Students near graduation)
$\mathcal{F}_n$	c	Courses required by undergraduating student n
$\mathcal{D}_{\text{grad}}$	(t, d)	Days in which professor t teaches some graduate course
$\mathcal{H}$	(t, d, s, b)	Graduate schedules that must to be avoided
$\mathcal{L}$	(t, d, s, b)	Locked schedules of professor t
$\mathcal{A}_p$	(d, s, b)	External undergraduate courses schedules by semester
$\mathcal{E}$	(c, d, s, b)	External courses (offered to others departments) schedules

4.1. Decision variables and parameters

The model considers the following parameters and decision and auxiliary variables:

1. $u[t, c]$ (parameter): Ordinal utility of relation professor-course. It represents the preference of professor t about course c. Each professor informs a ordered list of preferred courses and the first one has the greater u value, the second one the second greater u value and so on;

2. $\text{load}[t]$ (parameter): This parameter regards the classes load that professor t must satisfy with undergrad courses (or external courses). In the model, its value provides an upper

limit to how many courses of this kind he must teach. This depends, for instance, on professor being assistant or not, teaching grad courses or having administrative responsibilities;

3. $x[t, c, d, s, b]$ (decision variable): Indicator variable of event "professor t is allocated to teach course c in day d, shift s and time slot b";

4. $y[t, c]$ (auxiliary variable): Binary variable which informs whether professor t is matched to course c;

5. $z[t, d]$ (auxiliary variable): Binary variable which indicates whether professor t teaches some class in day d.

4.2. Objective function

Here, building a timetable to professors' educational tasks is guided, first of all, by the following criteria: answer as much as possible the preferences of the professors for courses and schedules. Thenceforth the objective function is defined. It is intended to maximize the quantity

$$Q = \sum_{\mathcal{T}} \sum_{\mathcal{C}} u[t, c] y[t, c] - M \sum_{\mathcal{T}} \sum_{\mathcal{D}} z[t, d]$$

The constant M is a positive large number, and it promotes a penalization on Q when increasing z values, *i.e.* an adjacent purpose is to concentrate teachings of a given professor at the minimum feasible number of days. We remark that, by definition, $z[t, d]$ is the indicator variable of the event "*professor t teaches some class in the day d*".

4.3. Constraints

I $\triangleright$ (Definition of the auxiliary variable $y[t, c]$) For each pair professor-course scheduled, the variable y equals 1 if, and only if, summing x over all triples day-shift-block equals 2, once each course considered here must has two time slots of classes per week.

$$\sum_{\mathcal{D}} \sum_{\mathcal{S}} \sum_{\mathcal{B}} x[t, c, d, s, b] = 2y[t, c], \quad \forall\ t \in \mathcal{T}, c \in \mathcal{C}$$

II $\triangleright$ (Definition of the auxiliary variable $z[t, d]$) It is intended to concentrate the professors' teachings at the minimum feasible number of days. This is done by means of the proposed penalization in Q and this unequality:

$$\sum_{\mathcal{C}} \sum_{\mathcal{S}} \sum_{\mathcal{B}} x[t, c, d, s, b] \leqslant 6z[t, d], \quad \forall\ t \in \mathcal{T}, d \in \mathcal{D}$$

Indeed, though we talk in *definition* of z, only this constraint does not guarantee that, if professor t_0 is scheduled for no classes on day d_0, $z[t_0, d_0] = 0$. But, once we are handling with an integer linear programming model, in the optimal solution the combined effects of this constraint and the penalization in Q act to make z work according to the interpretation we gave to it.

III $\triangleright$ Each department professor t should teach a maximum of $\text{load}[t]$ undergraduate or external courses. Hence,

$$\sum_{\mathcal{C}} y[t, c] \leqslant \text{load}[t], \quad \forall\ t \in \mathcal{T}_d$$

IV ▷ Some professors also cooperate with the Statistics Graduate Program (PPGE). In order to give to sum in z this information in the objective function, let $\mathcal{D}_{grad}$ be the set of couples (t, d) such that professor t teaches in day d some course on PPGE. So, we set

$$z[t, d] = 1, \quad \forall\ (t, d) \in \mathcal{D}_{grad}$$

Under the same argument, let $\mathcal{H}$ be the set of 4-tuples (t, d, s, b) such that professor t teaches some class on PPGE in (d, s, b). Note that $\mathcal{D}_{grad} = \{(t, d) : (t, d, s, b) \in \mathcal{H}\}$. The undergraduate schedule is subordinated to the PPGE one. So, to avoid time conflict between both programs, we set

$$\sum_{\mathcal{C}} x[t, c, d, s, b] = 0, \quad \forall\ (t, d, s, b) \in \mathcal{H}$$

V ▷ Given a specific time slot, a professor must be teaching not more than one class on it. This is a constraint present in almost all classes timetabling problems.

$$\sum_{\mathcal{C}} x[t, c, d, s, b] \leqslant 1, \quad \forall\ t \in \mathcal{T}, d \in \mathcal{D}, s \in \mathcal{S}, b \in \mathcal{B}$$

VI ▷ Each course must be ministered by the same professor. So,

$$\sum_{\mathcal{T}} y[t, c] = 1, \quad \forall\ c \in \mathcal{C}$$

VII ▷ It is necessary to avoid that instructors teach classes on extreme shifts in a day. Let $\mathcal{T}_d$ be the set of the department professors. Once in our case none of them showed interest by courses supposed to be offered at night, let's just lock this shifts setting

$$\sum_{\mathcal{C}} \sum_{\mathcal{D}} \sum_{\mathcal{B}} x[t, c, d, \text{night}, b] = 0, \quad \forall\ t \in \mathcal{T}_d$$

VIII ▷ We want to avoid classes of a same course happening two days in a row, as well as in two consecutive time blocks at a same shift and day. Put in other words, each course has classes in different and properly spaced weekdays. Then:

$$\sum_{\mathcal{T}} \sum_{\mathcal{S}} \sum_{\mathcal{B}} x[t, c, d, s, b] + \sum_{\mathcal{T}} \sum_{\mathcal{S}} \sum_{\mathcal{B}} x[t, c, d+1, s, b] \leqslant 1, \quad \forall\ c \in \mathcal{C}, d \in \mathcal{D}$$

IX ▷ (Basic courses constraints) The *basic courses* (Calculus, Linear Algebra and Analytic Geometry) are offered by a specific department. Their schedules are preset and the concerned departments look to conform their timetables. Let $\mathcal{C}_p$ be the set of undergraduate courses of semester p and $\mathcal{A}_p$ be the set of triples (d, s, b) scheduled for basic courses in semester p. Hence, to each semester we prohibit undergraduate courses be matched to this time slots:

$$\sum_{\mathcal{T}} \sum_{\mathcal{C}_p} \sum_{\mathcal{A}_p} x[t, c, d, s, b] = 0, \quad \forall\ p \in \mathcal{P}_b$$

X ▷ The following formulation guarantee that, in each semester p, a time block is filled with a maximum of one professor teaching one undergraduate course:

$$\sum_{\mathcal{T}} \sum_{\mathcal{C}_p} x[t, c, d, s, b] \leqslant 1, \quad \forall\, p \in \mathcal{P}, d \in \mathcal{D}, s \in \mathcal{S}, b \in \mathcal{B},$$

in which $\mathcal{C}_0$ is the set of optional courses, with no defined semesters.

XI ▷ External courses, *i.e.* courses offered by DE to others departments, also have a preset schedule. Let $\mathcal{E}$ be the set of 4-tuples (c, d, s, b) such that external course c is scheduled to time block b, shift s and day d. Then,

$$\sum_{\mathcal{T}} x[t, c, d, s, b] = 1, \quad \forall\, (c, d, s, b) \in \mathcal{E}.$$

XII ▷ It often happens that students are subscribed to courses which belong to different semesters. So, it is mainly considered the request of courses coming from students near to achieve undergraduate level. Let $\mathcal{N}$ denote the set of potential undergraduating students and let $\mathcal{F}_n, n \in \mathcal{N}$, be the set of courses required by student n. Hence, we set

$$\sum_{\mathcal{T}} \sum_{\mathcal{F}_n} x[t, c, d, s, b] \leqslant 1, \quad \forall\, n \in \mathcal{N}, d \in \mathcal{D}, s \in \mathcal{S}, b \in \mathcal{B}$$

which is equivalent to avoid time conflicts between every pair of distinct courses in $\mathcal{F}_n$.

XIII ▷ The following formulation avoid time blocks that professors prefer do not teach. Let $\mathcal{B}$ be the set of 4-tuples (t, d, s, b) such that professor t prefers do not be scheduled on (d, s, b). Hence,

$$\sum_{\mathcal{C}} x[t, c, d, s, b] = 0, \quad \forall\, (t, d, s, b) \in \mathcal{B}$$

4.4. An alternative coefficient of preferences meeting

We propose a coefficient for each solution i in a set $\mathcal{I}$ of optimal solutions of some instance, called here G_i, to judge whether some solution is better than an other, calculated by the following algorithm. To optimize the understanding, the reader may want to take a look at Table 3 to visualize the process. For particulars solution i and professor t, compute for the j-th allocation

$$g_j \triangleq g_j[i, t] = \frac{l_{j,t} - k_{j,t} + \delta_j}{l_{j,t} - 1 + \delta_j},$$

in which δ_j is the indicator variable of the event *"the element j is the last one of the list"* and $k_{j,t}$ is the ranking of the j-th course, in order of preference, matched to professor t in the list without all courses up to the $(j-1)$-th assigned course, whose length is $l_{j,t} = l_{1,t} - (j-1)$. Let us say that professor t was matched to $q_t \leqslant \text{load}[t]$ disciplines. Note that each $g_j \in (0, 1], j \in \{1, 2, \ldots, q_t\}$, measures the meeting of the j-th preference given that the previous allocations have already been considered.

Thus, set $G_{i,t} \triangleq \sum_{j=1}^{q_t} g_j / q_t$ and so define $G_i \triangleq (\#\mathcal{T})^{-1} \sum_{\mathcal{T}} G_{i,t},\ \forall\ i \in \mathcal{I}$, as a *coefficient* which measures how well the solution i meet the professors' preferences by *courses*. Then, given the distinct optimal solutions in $\mathcal{I}$ of some instance, we will say that $\max_i\{G_i : i \in \mathcal{I}\}$ indicates which is the *best* solution.

For instance, one may compute $G_{1,t}$, with some $t \in \{1, 2, \ldots, 18\}$, for the unique optimal solution presented in Table 3 and obtain for professor 13 $G_{1,13} = \frac{1}{2}(g_1 + g_2) = \frac{1}{2}\left(\frac{5-1}{5-1} + \frac{4-2}{4-1}\right) = \frac{5}{6}$.

5. Application and results

The model was implemented in AMPL and solved by means of the Gurobi 5.6 software (www.gurobi.com). To deal with the semester 2014.2, it was necessary 903 binary variables and 1,760 linear constraints. In 2015.1, the formulation had 2,514 binary variables and 1,055 constraints. In both cases, a unique optimal solution was found after less than 2 seconds using a Linux computer with AMD quad-core processor and 4GB RAM.

Table 3 shows the preferences list of each professor, in which the courses and department professors are labeled by integers from 1 to 32 and 1 to 18, respectively, besides one assistant professor. Courses in orange and red (square) labels were later attached to the original lists by a commission to handle feasibility. Disciplines of this kind which were matched to the respective professor have red labels.

For 2014.2 semester, about 70% of total professors had first preference met and only about 10% of them (professors 9 and 10) were not answered in the first three preferences. Table 2 compare over the semesters the number of days in which professors were scheduled to teach, the secondary criteria. As we said before, the scheduling process was manually executed up to semester 2014.1. For 2015.1, as high as 80% of total professors had first preference and, as 2014.1, only 10% were not answered in the first three preferences.

Table 2: Distribution of how many days were allocated to each professor whose load is at least two courses.

Days	2013.1	2013.2	2014.1	2014.2	2015.1
2	64%	21%	50%	**80%**	**87%**
3	24%	50%	33%	**20%**	**13%**
4	12%	29%	17%	**0%**	**0%**

Table 3: Solution for semester 2014.2. Legend: 0 Allocated course; 0 allocated course which were not in original list; 0 course added to the original list; 0 course allocated to assistants.

Professor	Days	Preferences list
1	2	8, 4, 3, 1, 7, 9, 5, 10, 6, 14
2	2	18, 21
3	2	11, 5, 6, 13, 23
4	2	7, 5, 1
5	2	2, 9, 14, 22, 19, 24, 25
6	2	2, 11, 12, 20, 16, 17, 18, 24, 32, 25
7	2	25, 26
8	2	21, 16, 17, 25
9	2	2, 8, 5, 4, 10, 1, 18, 20,23
10	3	4, 5, 8, 9, 10, 22, 23, 12, 25
11	2	1, 7, 21, 16, 25
12	2	12, 5, 25, 26
13	2	5, 8, 10, 21, 19
14	2	9, 11, 15, 20, 18, 21, 23, 24
15	2	16, 17, 25
16	2	6, 15, 5, 10, 11, 12, 13, 19, 20, 23, 24
17	2	4, 10, 2, 23, 8
18	3	19, 1, 3, 7, 21, 32, 25
Assistant	2	31, 32, 23

6. Conclusions and perspectives

First of all, building feasible timetables became a trivial task. The model makes simpler to deal with professors, if new conditions are proposed, by testing different scenarios. Both schedules for semesters 2014.2 and 2015.1, when the model was used, were well accepted and implemented by the department.

For future works, it is intended to study the influence of the parameters $u[t, c]$ in solutions and even in computational complexity. Also, a parallel model based on the coefficients G_i, in the sense of being specific about the required quality of the solutions, may has interesting features.

References

L.N. Ahmed, E. Özcan, and K. Ahmed. Solving high school timetabling problems worldwide using selection hyper-heuristics. *Expert Systems with Applications*, 42(13):5463 – 5471, 2015.

M.A. Bakir and C. Aksop. A 0-1 integer programming approach to a university timetabling problem. *Hacettepe Journal of Mathematics and Statistics*, 37(1):41 – 55, 2008.

E. Burke, K. Jackson, J.H. Kingston, and R. Weare. Automated university timetabling: The state of the art. *The Computer Journal*, 40(9):565 – 571, 1964.

E.K. Burke, J. Mareek, A.J. Parkes, and H. Rudová. Decomposition, reformulation, and diving in university course timetabling. *Computers & Operations Research*, 37(3):582 – 597, 2010. Hybrid Metaheuristics.

D. de Werra. An introduction to timetabling. *European Journal of Operational Research*, (9):151 – 162, 1985.

S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multi-comodity flow problems. 1975.

P.S. Ferreira, E.W. Karas, F.L. Palucoski, A.A. Ribeiro, and A.L. Silva. Aplicação de programação inteira na distribuição de encargos didáticos em instituições de ensino. *Trends in Applied and Computational Mathematics*, 12(2):135 – 144, 2011.

C. Gotlieb. The construction of class-teacher timetables. In *IFIP*, pages 73–77, Amsterdam, 1963.

J. Havas, A. Olsson, J. Persson, and M.S. Schierscher. Modeling and optimization of university timetabling - a case study in integer programming. *Göteborgs universitets publikationer*, 2013.

C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1982.

PATAT, editor. *Practice and Theory of Automated Timetabling (PATAT)*, 2014. http://www.patatconference.org/.

Gerhard Post, Luca Di Gaspero, JeffreyH. Kingston, Barry McCollum, and Andrea Schaerf. The third international timetabling competition. *Annals of Operations Research*, pages 1–7, 2013. ISSN 0254-5330. doi: 10.1007/s10479-013-1340-5. URL http://dx.doi.org/10.1007/s10479-013-1340-5.

H.G. Santos. Formulações e algoritmos para o problema de programação de horários em escolas, 2007.