INTEGER PROGRAMMING MODEL FOR UNIVERSITY COURSES TIMETABLING: A CASE STUDY

Abel Borges
Departamento de Estatística / UFPE
apdmbj1@de.ufpe.br

André Leite
Departamento de Estatística / UFPE
leite@de.ufpe.br

Raydonal Ospina
Departamento de Estatística / UFPE
raydonal@de.ufpe.br

Geiza Silva
Departamento de Estatística / UFPE
geiza@de.ufpe.br

RESUMO

Um modelo de programação linear inteira binária é proposto para tratar do problema de alocação de professores a disciplinas. O critério de otimização foi maximizar a preferência dos professores por cursos e horários. Além de restrições usuais, características específicas foram consideradas. Ao final, soluções múltiplas foram novamente ordenadas a partir de uma medida de aderência à lista de preferências dos professores. O modelo foi aplicado com sucesso nos semestres de 2014.2 e 2015.1 pelo Departamento de Estatística da Universidade Federal de Pernambuco.

PALAVRAS CHAVE. programação inteira, timetabling, problema da designação.

Área Principal: OC - Otimização Combinatória, EDU - PO na Educação, OA - Outras aplicações em PO

ABSTRACT

It is proposed a binary integer programming model to handle a real instance of the courses-to-professors timetabling problem. The optimization criteria is the preferences of professors by courses and schedules. Besides the common set of constraints, specific features of this situation are considered. A coefficient is introduced to decide about different optimal solutions (alternatively and independently of the approach used in the model). The model was successfully applied in semesters 2014.2 and 2015.1, by the Departamento de Estatística of the Universidade Federal de Pernambuco.

KEYWORDS. integer programming, university timetabling, courses scheduling.

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

1. Introduction

The tools and techniques available for integer programming are widely applied in real world problems. This potential has served to solve relevant questions in areas such as industrial research and economy, administration or scientific issues. Hence, within a good theoretical basis, it is possible to automatize general processes or even promote their optimization, as well as keep managers informed to support decision making.

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 also known as the *timetabling problem*, *cf.* de Werra [1985].

Periodically, universities, schools or human resources departments face the challenge of assigning tasks to their staff. There exists legal and institutional constraints which must be satisfied besides, eventually, the aim to optimize some aspect, according to a given criteria. Usually each instance has very specific features.

A huge part of the papers in this area 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.

Anyway, when handling educational institutions – schools and universities –, it is common to talk about *educational timetabling*. At this point, it is possible to distinguish two main classes of timetabling problems: exams timetabling and courses timetabling. A couple of fundamental differences between them are immediate, as pointed out by Burke et al. [1997].

Firstly, the 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. Therefore, exams timetabling depends more on the students information than courses timetabling. 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. Notice that the “students’ dynamic” are not the same in schools and universities. An undergraduate student enjoys more freedom in choosing his courses than the other one.

Now it is more clear why models built for specific problems do not claim to serve for all instances. 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 then 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, in particular the courses timetabling. The characteristics of the Departamento de Estatística (DE) of the Universidade Federal de Pernambuco (UFPE, `www.de.ufpe.br`) are observed. Educational questions must be satisfied and we try to take in account and 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 more well explained.

Section 2 reviews some works on timetabling, particularly on scheduling teaching tasks. Section 3 has brief comments on common and specific scheduling rules of the present instance regarding to matching professors to courses and times that must be considered. Section 4 describes the integer linear programming model proposed to this case. Finally, Section 5 presents the results obtained after using this model in two semesters in a row: 2014.2 and 2015.1.

2. State of the art

The main practical motivation of this research field could be regarded to the impracticability in solving 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., 1997]. Then, 42% of the (british) universities were used to schedule manually, 37% were 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]. These events had a positive impact in 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.

A alternative is heuristics and meta-heuristics based methods. In fact, proposals in this direction involve integer and mixed integer programming models. 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 such as:

- ▷ one, and 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;
- ▷ only assign adjacent shifts;
- ▷ 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

In the model, we consider the following decision and auxiliary variables and parameters:

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

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

Notice that the maximum hypothetical value assumed by the triple sum is 6 (six). This would be a reality if the considered professor teaches in the two time blocks of all three shifts. Combination of constraints regarding to control intervals between classes of each course and the professors' loads (to be described) avoid this result.

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 ▷ Each department professor t should teach a maximum of 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 on day d, shift s and time block b. Notice 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. Ergo,

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

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. They are necessary to many others departments of exact sciences. Then, basic courses' schedules are preset and thenceforth the concerned departments look to conform their timetable. Let

(i) $\mathcal{C}_p$ be the set of undergraduate courses of semester p and

(ii) $\mathcal{A}_p$ 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 blocks:

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

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,

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

Thus, we guarantee that exactly one professor is matched to every external course.

XII ▷ It often happens that students are subscribed to courses which belong to different semesters. Some difficulties arises from this fact when making decisions about which courses offer in each semester. So, in a first moment, it is considered the request of courses coming from students near to achieve undergraduate level; secondarily, students with no disapprovals have preference. 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 in which professors prefer do not teach. Let $\mathcal{L}$ be the set of 4-tuples (t, d, s, b) such that professor t prefers do not be scheduled on (d, s, b). Hence,

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

4.4. An alternative coefficient of preferences meeting

It is possible that distinct solutions yield the same optimal value of the objective function. This would mean that the problem has multiple optimal solutions, which is not a bad picture. In order to introduce a model-independent criterion to judge whether some solution is better than an other, let $\mathcal{I}$ be the set of optimal solutions of some instance.

We propose a coefficient for each solution i, called here G_i, calculated by the following algorithm. To optimize the understanding, the reader may want to take a look at Table 4 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 j-th course 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 $m_t \leqslant \text{load}[t]$ disciplines. Note that each $g_j \in (0, 1], j \in \{1, 2, \ldots, m_t\}$, measures the meeting of the j-th preference given that the previous allocations have already been considered.

Thus, setting

$$G_{i,t} \triangleq \frac{1}{m_t} \sum_{j=1}^{m_t} g_j, \qquad \forall\ i \in \mathcal{I}, t \in \mathcal{T}$$

we may define

$$G_i \triangleq |\mathcal{T}|^{-1} \sum_{\mathcal{T}} G_{i,t}, \qquad \forall\ i \in \mathcal{I}$$

as a coefficient of how well the solution i meet the professors' preferences by *courses*, where $|\cdot|$ denotes "cardinality of". 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 $G_{1,13}$ for professor $t = 13$ to the unique optimal solution $i = 1$ presented in Table 4 and thus obtain

$$G_{1,13} = \frac{g_1 + g_2}{2} = \frac{1}{2}\left(\frac{5-1}{5-1} + \frac{4-2}{4-1}\right) = \frac{5}{6}.$$

Alternatively, as proposed in Ferreira et al. [2011], let $e[t,c]$ be the ranking of the course c in the preferences list of professor t and keep M_t for the set of courses matched to professor t, so that $M_t = \{c \in \mathcal{C} : y[t,c] = 1\}$. Then,

$$I_{i,t} = \frac{\sum_{M_t} e[t,c]y[t,c]}{1 + 2 + \ldots + |M_t|}, \qquad \forall\ i \in \mathcal{I}, t \in \mathcal{T},$$

is the satisfaction index of professor t in solution i and

$$I_i = |\mathcal{T}|^{-1} \sum_{\mathcal{T}} I_{i,t}, \qquad \forall\ i \in \mathcal{I},$$

measures the satisfaction of all professors in a given solution i.

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 case, a unique solution was found after less than 2 seconds using a Linux computer with AMD quad-core processor and 4GB RAM.

For example, until reach the solution obtained for 2014.2 problem, which is presented in Table 4, 273 nodes of the polyhedron defined by constraints were visited and 19,587 iterations of simplex method were done in 1.47 seconds.

Table 4 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 about 10% of them (professors 9 and 10) were not answered in the first three preferences. Tables 2 and 3 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.

Days	**2013.1**	**2013.2**	**2014.1**	**2014.2**	**2015.1**
2	75%	48%	75%	**89%**	**75%**
3	17%	33%	17%	**11%**	**25%**
4	8%	19%	8%	**0%**	**0%**

Table 3: 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 4: 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, for this model, the influence of the parameters $u[t, c]$ in solutions and even in computational complexity. Also, a parallel approach of the problem based on the coefficients G_i, in the sense of being specific about the required quality of the solutions, may has interesting features.

Acknowledgments

The authors would like to thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) via Universidade Federal de Pernambuco (UFPE) and the Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE) (APQ-0446-3.08/12) for the financial support to this project.

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, 1997.

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.