CHILEAN JOURNAL OF STATISTICS

Edited by Víctor Leiva and Carolina Marchant

A free open access journal indexed by

Volume 11 Number 2
December 2020

ISSN: 0718-7912 (print)
ISSN: 0718-7920 (online)

Published by the
Chilean Statistical Society

Aims

The Chilean Journal of Statistics (ChJS) is an official publication of the Chilean Statistical Society (`www.soche.cl`). The ChJS takes the place of *Revista de la Sociedad Chilena de Estadística*, which was published from 1984 to 2000.

The ChJS covers a broad range of topics in statistics, as well as in artificial intelligence, big data, data science, and machine learning, focused mainly on research articles. However, review, survey, and teaching papers, as well as material for statistical discussion, could be also published exceptionally. Each paper published in the ChJS must consider, in addition to its theoretical and/or methodological novelty, simulations for validating its novel theoretical and/or methodological proposal, as well as an illustration/application with real data.

The ChJS editorial board plans to publish one volume per year, with two issues in each volume. On some occasions, certain events or topics may be published in one or more special issues prepared by a guest editor.

Contents

Chilean Journal of Statistics
Vol. 11, No. 2, December 2020, 151–162

Data Science and Education
Research Paper

A timetabling system for scheduling courses of statistics and data science: Methodology and case study

André Leite[1], Abel Borges[1], Geiza Silva[1] and Raydonal Ospina[1,*]

[1]Departamento de Estatística, CASTLab,
Universidade Federal de Pernambuco, Recife, PE, Brazil

(Received: 25 August 2020 · Accepted in final form: 25 October 2020)

Abstract

The purpose of this work is to present a decision support system for scheduling courses of statistics and data science to help educational institutions. Currently, an increasing demand of statisticians and data scientists around the world of businesses and organizations is observed. By distributing resources, such as the available time for teachers to form those human personnel, is challenging because of the many dependencies that can exist, which must be taken into account. We describe an integer programming formulation to handle a real instance of a courses-to-lecturers timetabling problem based on a case study. The proposed system is successfully applied by experimental runs using course offerings and classroom data from past semesters.

Keywords: Integer programming· Mathematical programming · Timetabling problem.

Mathematics Subject Classification: Primary 90C10 · Secondary 6207.

1. Introduction

Integer programming tries to allocate finite resources, in an optimal way. In usual applications, the problem description leads to obtaining an optimal value, either a maximum or minimum, for an objective function based on the number of units of resources allocated to each competing entity and constraints on the allocation of the resources. The parameters that describe the number of units are referred to as decision variables. 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 the literature of combinatorial optimization is the matching problem (Papadimitriou and Steiglitz, 1982; Even et al., 1975). In a particular situation, it is also known as the timetabling problem and can be formulated as a binary integer programming problem (MirHassani, 2006; de Werra, 1985; Selim, 1983).

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

*Corresponding author. Email: raydonal@de.ufpe.br

ISSN: 0718-7912 (print)/ISSN 0718-7920 (online)

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 (Burke and De Causmaecker, 2002). When handling educational institutions – schools and universities –, it is common to talk about educational timetabling. At this point, it is possible to distinguish three main classes of timetabling problems (Burke et al., 1997; Schaerf, 1999):

- School timetabling problem (STP): In this ranking are found the problems of the weekly schedule between teacher/class in schools/colleges. Here it is considered that the subjects are fixed for each class and the main objective is to avoid one teacher being allocated to two classes simultaneously or two classes having lessons with one particular teacher in one timetable.
- University course timetabling problem (UCTP): It has in view the weekly schedule of all subjects for all periods of the university courses by determining the relation (teacher/classroom/classroom). It differs from the school timetabling, because in this classification the students can choose the subjects (electives) they wish to enroll. The objective is also to minimize the overlap of any of the variables involved.
- Examination timetabling problem (ETP): It deals with the scheduling of exams for university courses, avoiding overlapping of exams of subjects that have students in common and keeping the exam dates as far away from students as possible.

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. The massive use of computers to solve timetabling problems probably started with the construction of class-teacher timetables in 1963 (Gotlieb, 1963). A survey conducted by the Automated Scheduling and Planning Group at the University of Nottingham in the year of 1995 obtained feedback from 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 is very popular because it promotes competitions such as the International Timetabling Competition (ITC), which have had three versions (2002, 2007 and 2011), Post et al. (2016). 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 an useful approach to this problem (Bakir and Aksop, 2008; Ferreira et al., 2011; Havas et al., 2013; Sánchez-Partida et al., 2014; Eledum, 2017)).

Data science is currently getting attention because of the Big Data trend. Different sectors of society are now able to continuously gather data from websites, mobile devices, social media tools and legacy systems, not to mention the burgeoning Internet of Things (IoT). Institutions, governments and businesses today are drowning in data. This is information that, when examined with discerning eyes, ostensibly reveals the trends that could help better serve customers, increase sales, keep their businesses growing and help save lives. In global emergencies like the coronavirus disease (COVID-19) pandemic (Ghebreyesus, 2020), open science policies can remove obstacles to the free flow of research data and ideas, and thus accelerate the pace of research critical to combating the disease (Zastrow, 2020). In this sense, Statisticians and Data scientists are in demand because there is a shortage of qualified data science professionals on the market today.

In this work, we design a decision support system for scheduling courses of statistics and data science. We focus to help educational institutions to distributing resources efficiently such as the available time spent for teachers to form those human resources, once dependencies that can exist in the process of distribution need to be taken into account to reduce tensions that confront teachers of statistics and data science in practice (Cobb, 2011; Batanero and Díaz, 2012).

The objective of this paper is to provide a case study of the university courses timetabling class. We examine the Department of Statistics from the Federal University of Pernambuco in Brazil (`https://www.ufpe.br/dep-estatistica`) because it has a similar structure to other undergraduate and graduate programs in statistics and data sciences in Latin America and other countries. Educational questions must be satisfied and we try to take into account and answer the lecturers' 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 schedules.

The rest of this paper is organized as follows. Section 2 has brief comments on common and specific scheduling rules of the present instance regarding matching lecturers to courses and times that must be considered and describes the integer linear programming model proposed in this case. Section 3 introduce a coefficient for multiple solutions to choose the optimal allocation based on satisfaction of lecturers. Section 4 presents the results obtained after using this model in several semesters. Section 5 concludes the paper.

2. Scheduling rules and mathematical model

Firstly, the model is based on the satisfiability approach of optimizing the lecturer's 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 (there may be more):

(a) One, and only one, lecturer teaches each class.
(b) Lecturers just may be in one place at a time.
(c) Each course must be taught by the same lecturer.
(d) Lecturers have a maximum load of courses to teach.
(e) Only assign adjacent shifts.
(f) 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 the same course should be properly spaced over the weekdays and so on.

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

In our approach we consider the following decision and auxiliary variables and parameters:

(a) $u(t, c)$ (parameter): Ordinal utility of relation professor-course. It represents the preference of the lecturer t about course c. Each lecturer 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.

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

Set	Index	Description
$\mathcal{T}, \mathcal{T}_d, \mathcal{T}_s$	t	Lecturers, department lecturers 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 lecturer 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 lecturer 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

(b) load(t) (parameter): This parameter regards the classes load that lecturer 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 lecturer being assistant or not, teaching courses or having administrative responsibilities.
(c) $x(t, c, d, s, b)$ (decision variable): Indicator variable of event "lecturer t is allocated to teach course c in day d, shift s and time slot b".
(d) $y(t, c)$ (auxiliary variable): Binary variable which informs whether lecturer t is matched to course c.
(e) $z(t, d)$ (auxiliary variable): Binary variable which indicates whether lecturer t teaches some class in day d.

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

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

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

Next, we introduce the following notations and settings to determine the constraints of the problem:

(a) Definition of the auxiliary variable $y(t, c)$: For each pair lecturer-course scheduled, the variable y equals one if, and only if, summing x over all triples day-shift-block equals two, once each course considered here must have two time blocks of classes per week, which is formulated as

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

(b) Definition of the auxiliary variable $z(t,d)$: It is intended to concentrate the lecturers' teachings at minimum feasible number of days, which is done by means of the proposed penalization in Q and an inequality stated as

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

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

Indeed, though we talk in definition of z, only this constraint does not guarantee that, if lecturer 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.

(c) Each department lecturer t should teach a maximum of $\text{load}(t)$ undergraduate or external courses, which is established by

$$\sum_{\mathcal{C}} y(t,c) \leqslant \text{load}(t), \quad \forall\ t \in \mathcal{T}_d.$$

(d) Some lecturers also cooperate with the statistics graduate program. In order to sum in z this information in the objective function, let $\mathcal{D}_{\text{grad}}$ be the set of couples (t,d) such that lecturer t teaches in day d some course on statistics graduate program, which is formulated as

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

Under the same argument, let $\mathcal{H}$ be the set of 4-tuples (t,d,s,b) such that lecturer t teaches some class on the statistics graduate program on day d, shift s and time block b. Notice that $\mathcal{D}_{\text{grad}} = \{(t,d) : (t,d,s,b) \in \mathcal{H}\}$. Generally, the undergraduate schedule is subordinated to the statistics graduate program. Then, to avoid time conflict between both programs, we set

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

(e) Given a specific time slot, a lecturer must be teaching not more than one class on it. This is a constraint present in almost all classes timetabling problems and stated as

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

(f) Each course must be delivered by the same lecturer, which is established by

$$\sum_{\mathcal{T}} y(t,c) = 1, \quad \forall\ c \in \mathcal{C}.$$

(g) 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 lecturers. Once in our case none of them showed interest by courses supposed to be offered at night, we write this shift setting

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

(h) We want to avoid classes of a same course happening two days in a row, as well as in two consecutive time blocks at the same shift and day, that is, each course has classes in different and properly spaced weekdays. Then, this can be formulated as

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

(i) Basic courses constraints: These courses (for example, calculus, linear algebra, and analytic geometry) are offered by a specific department. They are necessary to many other departments of exact sciences. Then, basic courses schedules are preset and thenceforth the concerned departments look for conforming their timetable. Thus,
   (1) Let $\mathcal{C}_p$ be the set of undergraduate courses of semester p and
   (2) Let $\mathcal{A}_p$ be the set of triples (d,s,b) scheduled for basic courses in semester p.
   Hence, in each semester, we restrict undergraduate courses to be matched to time blocks by means of

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

(j) We must guarantee that, in each semester p, a time block is filled with a maximum of one lecturer teaching one undergraduate course. Therefore, for all $p\in\mathcal{P}, d\in\mathcal{D}, s\in\mathcal{S}, b\in\mathcal{B}$, we have

$$\sum_{\mathcal{T}}\sum_{\mathcal{C}_p} x(t,c,d,s,b)\leqslant 1,$$

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

(k) External courses must be also considered, that is, courses offered by the Department of Statistics 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, we get that

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

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

(l) It often happens that students are subscribed to courses which belong to different semesters. Some difficulties arise from this fact when making decisions about which courses offer in each semester. Hence, 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 a preference. Let $\mathcal{N}$ denote the set of potential undergraduating students and let $\mathcal{F}_n$, for $n\in\mathcal{N}$, be the set of courses required by a student n. Then, for all $n\in\mathcal{N}, d\in\mathcal{D}, s\in\mathcal{S}, b\in\mathcal{B}$, we set

$$\sum_{\mathcal{T}}\sum_{\mathcal{F}_n} x(t,c,d,s,b)\leqslant 1,$$

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

(m) We must avoid also time blocks in which lecturers prefer not teach. Let $\mathcal{L}$ be the set of 4-tuples (t, d, s, b) such that lecturer t prefers do not be scheduled on (d, s, b). Therefore, we have that

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

If all the constraints are considered, we set it is a constraint satisfaction problem.

3. An alternative coefficient for multiple optimal solutions

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 another, 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 particular solutions i and lecturer t, compute for the j-th allocation the expression given by

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

where δ_j is the indicator variable of the event "the j-th course is the last one of the list". Here, $k_{j,t}$ is the ranking of the j-th course, in order of preference, matched to lecturer 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 lecturer t is matched to $m_t \leqslant \text{load}(t)$ disciplines. Note that each $g_j \in (0, 1]$, for $j \in \{1, 2, \ldots, m_t\}$, 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 a satisfaction index

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

as a coefficient of how well the solution i meet the lecturers' preferences by courses, where $|\cdot|$ denotes "cardinality of the set". For instance, one may compute $G_{1,13}$ for lecturer $t = 13$ to the unique optimal solution $i = 1$ presented in Table 4 and thus obtain

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

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.

4. Application and results

The model was implemented in `AMPL` (a mathematical programming language) an algebraic modeling language to describe and solve high-complexity problems for large-scale (Fourer et al., 1987) and solved by means of the `Gurobi` (https://www.gurobi.com) an optimizer software (Bixby, 2007) linked with the `R` statistical software https://www.r-project.org. A computer with `Linux Ubuntu` 19.04 - Disco Dingo with `AMD quad-core` processor with a frequency of 2.80 GHz and 4GB RAM was used to carry out the tests.

The proposed model was applied in three consecutive semesters. In the first semester (Instance 4), there were 903 binary variables and 1,760 linear constraints. For the second semester that the model is applied (Instance 5), the formulation had 2,514 binary variables and 1,055 constraints. In both cases, a unique solution was found after less than 2 seconds. Instance 6 presents a bigger problem with 3,750 binary variables. Therefore, we observe an increase in the size of the instances over the semesters.

Table 4 (Instance 5) shows the preference list of each lecturer, in which the courses and department lecturers are labeled by integers from 1 to 32 and 1 to 18, respectively, besides one assistant lecturer. Courses with square shape labels were later attached to the original lists by a commission to handle feasibility.

For Instance 4, about 70% of total lecturers had first preference met and about 10% of them (lecturers 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 lecturers were scheduled to teach, the secondary criteria. As we said before, the scheduling process was manually executed up to Instance 3. Also, for Instance 5, as high as 80% of total lecturers had first preference and, as Instance 4, 10% were not answered in the first three preferences. Therefore, for Instance 6, about 60% of total lecturers had first preference met and about only 8% of them were not attended in the first three preferences.

Table 2. Distribution of how many days were allocated to each lecturer.

	Institution					
Days	1	2	3	4	5	6
2	75%	48%	75%	89%	75%	83%
3	17%	33%	17%	11%	25%	17%
4	8%	19%	8%	0%	0%	0%

Table 3. Distribution of how many days were allocated to each lecturer whose load is at least two courses.

	Institution					
Days	1	2	3	4	5	6
2	64%	21%	50%	80%	87%	83%
3	24%	50%	33%	20%	13%	17%
4	12%	29%	17%	0%	0%	0%

Table 4. Solution for Instance 4.

Lecturer	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**

where the notations used indicate that:

0 allocated course;

0 allocated course which were not in original list;

0 course added to the original list;

0 course allocated to assistants.

5. Conclusions and perspectives

In this paper we present a scheduling problem at the Department of Statistics of the Federal University of Pernambuco. The case study developed aims to allocate teachers to subjects according to their teaching preferences and reduce the number of days of classes. An entire programming model was proposed and three real instances were tested, relative to three consecutive semesters of the course. The optimal solutions obtained were applied in the department (with good acceptance among teachers).

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. Therefore, during these difficult times of mitigations measures given by the COVID-19 pandemic, we are studying different adaptations of the timetabling formulation present here to include Social Distancing in the re-opening processes of courses in institutions of safe form and include time-space structures as a preferences of teachers, information asymmetries on real-time updates, and travel times.

Acknowledgements

The authors thank Editors and Referees for their valuable suggestions. We gratefully acknowledge partial financial support from CNPq (grant 305305/2019-0) and CAPES, Brazil.

References

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

Batanero, C. and Díaz, C., 2012. Training school teachers to teach probability: reflections and challenges. Chilean Journal of Statistics, 3, 3–13.

Bixby, B., 2007. The gurobi optimizer. Transportation Research Part B, 41, 159–178.

Burke, E. and De Causmaecker, P., 2002. Practice and theory of automated timetabling. Proceeding of the IV 4th International Conference, Gent, Belgium.

Burke, E., Jackson, K., Kingston, J., and Weare, R., 1997. Automated university timetabling: The state of the art. The Computer Journal, 40, 565–571.

Cobb, G.W., 2011. Teaching statistics: Some important tensions. Chilean Journal of Statistics, 2, 31–62.

de Werra, D., 1985. An introduction to timetabling. European journal of operational research, 19, 151–162.

Eledum, H., 2017. The university timetabling problem: Modeling and solution using binary integer programming with penalty functions. International Journal of Applied Mathematics and Statistics, 56, 164–178.

Even, S., Itai, A., and Shamir, A., 1975. On the complexity of time table and multi-commodity flow problems. Proceeding of the 16th Annual Symposium on Foundations of Computer Science, pp. 184–193.

Ferreira, P., Karas, E., Palucoski, F., Ribeiro, A., and Silva, A. (2011). Application of integer programming in the distribution of teaching charges in educational institutions. Trends in Applied and Computational Mathematics, 12, 135–144.

Fourer, R., Gay, D.M., and Kernighan, B.W., 1987. AMPL: A mathematical programming language. AT & T Bell Laboratories Murray Hill, New Jersey.

Ghebreyesus, T.A., 2020. WHO Director-General opening remarks at the Member State Briefing on the COVID-19 pandemic evaluation. Available at https://bit.ly/36nRIvG. Retrieval date: July 9, 2020.

Gotlieb, C., 1963. The construction of class-teacher timetables. Proceeding IFIP, Vol. 62, pp. 73–77, Amsterdam.

Havas, J., Olsson, A., Persson, J., and Schierscher, M. (2013). Modeling and optimization of university timetabling: A case study in integer programming. Gothenburg University Publisher, Gothenburg.

MirHassani, S., 2006. A computational approach to enhancing course timetabling with integer programming. Applied Mathematics and Computation, 175, 814–822.

Papadimitriou, C.H. and Steiglitz, K., 1982. Combinatorial Optimization: Algorithms and Complexity. Dover, New York.

Post, G., Di Gaspero, L., Kingston, J., McCollum, B., and Schaerf, A., 2016. The third international timetabling competition. Annals of Operations Research, 239, 69–75.

Schaerf, A., 1999. A survey of automated timetabling. Artificial Intelligence Review, 13, 87–127.

Selim, S., 1983. An algorithm for producing course and lecturer timetables. Computers and Education, 7, 101–108.

Sánchez-Partida, D., Flores, J.L., and Olivares-Benitez, E., 2014. An integer linear programming model for a university timetabling problem considering time windows and consecutive periods. Mathematical Methods of Operational Research, 6, 158–173.

Zastrow, M., 2020. Open science takes on the coronavirus pandemic. Nature, 581, 109–110

Information for authors

The editorial board of the Chilean Journal of Statistics (ChJS) is seeking papers, which will be refereed. We encourage the authors to submit a PDF electronic version of the manuscript in a free format to the Editors-in-Chief of the ChJS (E-mail: chilean.journal.of.statistics@gmail.com). Submitted manuscripts must be written in English and contain the name and affiliation of each author followed by a leading abstract and keywords. The authors must include a"cover letter" presenting their manuscript and mentioning: "We confirm that this manuscript has been read and approved by all named authors. In addition, we declare that the manuscript is original and it is not being published or submitted for publication elsewhere".

Preparation of accepted manuscripts

Manuscripts accepted in the ChJS must be prepared in Latex using the ChJS format. The Latex template and ChJS class files for preparation of accepted manuscripts are available at http://chjs.mat.utfsm.cl/files/ChJS.zip. Such as its submitted version, manuscripts accepted in the ChJS must be written in English and contain the name and affiliation of each author, followed by a leading abstract and keywords, but now mathematics subject classification (primary and secondary) are required. AMS classification is available at http://www.ams.org/mathscinet/msc/. Sections must be numbered 1, 2, etc., where Section 1 is the introduction part. References must be collected at the end of the manuscript in alphabetical order as in the following examples:

Arellano-Valle, R., 1994. Elliptical Distributions: Properties, Inference and Applications in Regression Models. Unpublished Ph.D. Thesis. Department of Statistics, University of São Paulo, Brazil.

Cook, R.D., 1997. Local influence. In Kotz, S., Read, C.B., and Banks, D.L. (Eds.), Encyclopedia of Statistical Sciences, Vol. 1., Wiley, New York, pp. 380-385.

Rukhin, A.L., 2009. Identities for negative moments of quadratic forms in normal variables. Statistics and Probability Letters, 79, 1004-1007.

Stein, M.L., 1999. Statistical Interpolation of Spatial Data: Some Theory for Kriging. Springer, New York.

Tsay, R.S., Peña, D., and Pankratz, A.E., 2000. Outliers in multivariate time series. Biometrika, 87, 789-804.

References in the text must be given by the author's name and year of publication, e.g., Gelfand and Smith (1990). In the case of more than two authors, the citation must be written as Tsay et al. (2000).

Copyright