# Development of a tool for university timetabling using an integrated spreadsheet

**J.A. Marmolejo · I. Aranzabal · S. Sanchez · K. Tellaeche**

**Abstract** This paper shows the development of a user-friendly tool based on a spreadsheet that uses an open access solver (external branch and cut solver) to perform classroom-time assignments (timetabling). This tool works as an interface that uses linear integer programming (0-1) to assign group blocks of teaching schedules and classrooms according to certain soft and hard constraints. The case study describes the problem presented in the Faculty of Engineering at the Anahuac University in Mexico, which considers 130 professors, 15 classrooms, 3 types of classroom, up to 500 groups, 10 timeslots and a set of 19 combinations of days of the week on which a teaching group may be sheduled. This situation is modeled as a problem of scheduling with 285,000 binary variables and 3,126 constraints. Finally, we describe the process of loading information into the proposed interface as well as the results obtained.

**Keywords** university timetabling · integer programming · spreadsheet

## 1 Introduction

The problem of time scheduling (timetabling) involves assigning multiple courses (teaching groups) to be thought by different teachers in different time slots and multiple classrooms. The solution of timetabling is very important for universities and schools as it is vital for the planning of their activities and the better use of their facilities.

Giving that the programming schedules in a university should consider each cases particular constraints, the same tool cannot be used for different

Anahuac University
Av. Universidad Anahuac 46. Lomas Anahuac. Huixquilucan, Estado de Mexico
Tel.: +52-55-562701210
E-mail: jose.marmolejo@anahuac.mx

problems (different universities), usually a specific tool is developed for each university.

However, the complexity of this problem makes it very difficult to be solved to optimality. This problem is quite complex because of its highly combinatorial nature. When the university-timetabling problem involves many features of the given university it is considered a Np-Hard problem [3,4,12]. So usually some simplification tools are developed to make it more treatable.

This paper presents a pilot project to automate the allocation of classrooms and subjects at the Faculty of Engineering of the University Anahuac Mexico. Currently, the Faculty of Engineering performs this scheduling manually. The main objective of this work is to develop a realistic mathematical model and easy to use tool based on a spread sheet to solve the problem of allocation. This, considering that the proposed tool is computationally efficient according to the number of variables and constraint the university has. This case study is organized as follows. In Section 2 a breef review of the literature of university timetabling is presented and the timetabling problem for the Faculty of Engineering of the University Anahuac Mexico is discussed. The mathematical model used is described in Section 3. Section 4 shows the spreadsheet used as well as the necessary information to be loaded into the developed interface. The use of Open Solver (external branch and cut solver) used as a plug in within the spreadsheet is briefly described in this section as well. We present the results obtained with the case of study as well as the calculation time required to reach a feasible solution. We conclude in Section 5 presenting the limitations we had during the development of the tool and discussing future suggestions.

## 2 A breef iterature review and the case of study

Approaches to solving the problem of University Course Timetabling (UCT) have been several. We can see heuristic approaches as simulated annealing, genetic algorithms and others [1,6,11]. Another approach used is multi agent systems. The agents are classified in: autonomous, intelligent, proactive, learner and others [5,8,10]. In [2,7,9] approaches to mixed-integer programming that solve the UCT are developed.

The main objective of our work was to develop a spreadsheet-based approach that would enhance scheduling effectiveness by capturing specific conditions of the Anahuac University while being easy to implement.

Throughout our studies we have noticed that administrative staff of Universidad Anahuac faces an increasingly complex issue regarding the allocation of class schedules. This situation is due to several factors such as:

- Continuous growth in the number of students at Universidad Anahuac's Engineering Faculty.
- The different needs of each course: classroom type, possible time.
- Limited number of available classrooms.
- Different Engineering degrees taught at Universidad Anahuac.

As a result of the above, some classes may be assigned to different buildings other than the one of engineering school, complicating the transfer of students and affecting the administrative operation of the Faculty. After analyzing the various factors we realized that using a mathematical programming model and a computational tool of easy access, we could improve the allocation of schedules so that the use of classrooms and teachers are optimized with the resources available at the University. Currently the allocation of rooms-times-Class is performed by the administrative staff of the Faculty of Engineering manually at the beginning of each semester and it not only takes a lot of time but also does not ensure that the given allocation maximizes the use of the university's resources. The main goal was to develop a tool using a spreadsheet that provides a feasible allocation of day, time and classroom to maximize the number of classes taught at the Faculty of Engineering. Another objective was to design the mathematical programming model that considers the constraints mentioned and proposing an alternative simple and accessible application that allows any user to use the tool developed.

The project focuses on a case study of the Faculty of Engineering considering all different degrees thought, as well as all the available classrooms in that building. However, it is intended so it can be scalable to consider different faculties and buildings in the future. We believe that the implementation of this project can improve the University' s operation and facilitate the scheduling process. The developed tool focuses only on making a better allocation and does not consider other administrative issued the school faces.

## 3 Mathematical model

We propose a model to maximize the number of assigned classes per semester (objective function) , complying with the availability of professors, the type and the number of classrooms, class schedules , and the days when the subjects are taught , avoiding the overlap of classrooms, professors, days and time slots in different subjects (constraints).

Based on the information gathered from the engineering faculty, the timetabling model is structured as follows:

### 3.1 Notation

*Sets*
Let $I$ be the set of teaching groups. An element $i \in I$ identifies a specific teaching group (professor-subject), e.g. $I$={1,2,3,4,5}. Let $J$ be the set of the day of the week on which a teaching group may be sheduled, e.g. $J$={1,2,3,4,5}.(See table 1). An element $k \in K$ is a specific timeslot, e.g. $K$={1,2,3,4,5}. (See table 2) Finally, let L be the set of type of classroom, a specific type of classroom is any $l \in L$, e.g. $L$={1,2,3} . Let $\mathbb{Z}$ denote the set of integers {0,1}.

*Parameters*

$F_{ijkl}$ = Matrix feasibility of the teaching group $i$ assignment to the alternative of days $j$ at the timeslot $k$ in the type of classroom $l$.

$$F_{ijkl} = \begin{cases} 1, \text{if the assignment of teaching group i is feasible} \\ 0, \text{otherwise} \end{cases}$$

$P_{ip}$ = Matrix professor-subject assignment . It takes the value of zero when the teacher is not assigned to that subject and one when is assigned. For the case study, it is assumed that the assignment of courses to professors precedes the timetabling process, and the teaching load of each teacher is an input to the timetablers.

$$P_{ip} = \begin{cases} 1, \text{when the teacher is assigned to the specific subject} \\ 0, \text{otherwise} \end{cases}$$

$b_l$ = Number of classrooms by type $l$.

*Binary decision variable*

$$x_{ijkl} = \begin{cases} 1, \text{when the teaching group i is assigned to the alternative} \\ \quad \text{of days j at the timeslot k in the type of classroom l} \\ 0, \text{otherwise} \end{cases}$$

According to the needs of a specific teaching group they should be assigned in a specific type of classroom.

*Objective function.* The objective function only considers the maximization of the number of teaching groups assigned in a feasible schedule.

$$Max \quad Z = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} F_{ijkl} x_{ijkl} \tag{1}$$

Subject to constraints:

*Uniqueness constraints.* Every classroom may be assigned to at most one course, one teacher and one group of students at a time.

$$\sum_{j \in J} \sum_{k \in K} \sum_{l \in L} x_{ijkl} \leq 1; \quad \forall i \in I \tag{2}$$

*Classrooms constraints.* Each type of classroom can only be assigned a limited number of teaching groups according to the available rooms. No classroom can hold more than one class at any time.

$$\sum_{i \in I} \sum_{j \in \Omega_j} x_{ijkl} \leq b_l, \quad \forall k = 1, l \in L \tag{3}$$

$$\sum_{i \in I} \sum_{j \in \Omega_j} x_{ijkl} + x_{i,j=1,k-1,l} \leq b_l, \quad \forall k \neq 1, l \in L \tag{4}$$

$$\sum_{i \in I} \sum_{j \in \Psi_j} x_{ijkl} \leq b_l, \quad \forall k = 1, l \in L \tag{5}$$

$$\sum_{i \in I} \sum_{j \in \Psi_j} x_{ijkl} + x_{i,j=2,k-1,l} \leq b_l, \quad \forall k \neq 1, l \in L \tag{6}$$

$$\sum_{i \in I} \sum_{j \in \Phi_j} x_{ijkl} \leq b_l, \quad \forall k = 1, l \in L \tag{7}$$

$$\sum_{i \in I} \sum_{j \in \Phi_j} x_{ijkl} + x_{i,j=3,k-1,l} \leq b_l, \quad \forall k \neq 1, l \in L \tag{8}$$

$$\sum_{i \in I} \sum_{j \in \Upsilon_j} x_{ijkl} \leq b_l, \quad \forall k = 1, l \in L \tag{9}$$

$$\sum_{i \in I} \sum_{j \in \Upsilon_j} x_{ijkl} + x_{i,j=4,k-1,l} \leq b_l, \quad \forall k \neq 1, l \in L \tag{10}$$

$$\sum_{i \in I} \sum_{j \in \Pi_j} x_{ijkl} \leq b_l, \quad \forall k = 1, l \in L \tag{11}$$

$$\sum_{i \in I} \sum_{j \in \Pi_j} x_{ijkl} + x_{i,j=5,k-1,l} \leq b_l, \quad \forall k \neq 1, l \in L \tag{12}$$

*Professor constraints.* Each teaching group is previously assigned a teacher, one teacher can be assigned more than one group but no teacher can have more than one teaching group scheduled at the same time.

$$\sum_{i \in I} W_{ip} \sum_{i \in I} \sum_{j \in \Omega_j} \sum_{l \in L} x_{ijkl} \leq 1, \quad \forall k = 1, p \in P \tag{13}$$

$$\sum_{i \in I} W_{ip} \sum_{i \in I} \sum_{j \in \Omega_j} \sum_{l \in L} x_{ijkl} + x_{i,j=1,k-1,l} \leq 1, \quad \forall k \neq 1, p \in P \tag{14}$$

$$\sum_{i \in I} W_{ip} \sum_{i \in I} \sum_{j \in \Psi_j} \sum_{l \in L} x_{ijkl} \leq 1, \quad \forall k = 1, p \in P \tag{15}$$

$$\sum_{i \in I} W_{ip} \sum_{i \in I} \sum_{j \in \Psi_j} \sum_{l \in L} x_{ijkl} + x_{i,j=2,k-1,l} \leq 1, \quad \forall k \neq 1, p \in P \tag{16}$$

**Table 1** Days of the week on which a teaching group may be scheduled

| Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|
| | | | | Friday |
| | Tuesday | | Thursday | |
| Monday | | Wednesday | | |
| Monday | | Wednesday | | Friday |
| Monday | | | | |
| | | Wednesday | | Friday |
| | | | Thursday | |
| | Tuesday | | | |
| | | Wednesday | | |
| Monday | Tuesday | | | |
| Monday | | | | Friday |
| | | | | |
| | Tuesday | Wednesday | Thursday | |
| Monday | Tuesday | Wednesday | Thursday | Friday |
| Monday | Tuesday | Wednesday | Thursday | |
| Monday | Tuesday | Wednesday | | Friday |
| | | Wednesday | Thursday | |
| | Tuesday | | Thursday | Friday |
| Monday | | Wednesday | Thursday | |
| Monday | Tuesday | Wedneday | | |

$$\sum_{i\in I} W_{ip} \sum_{i\in I} \sum_{j\in \Phi_j} \sum_{l\in L} x_{ijkl} \leq 1, \quad \forall k = 1, p \in P \tag{17}$$

$$\sum_{i\in I} W_{ip} \sum_{i\in I} \sum_{j\in \Phi_j} \sum_{l\in L} x_{ijkl} + x_{i,j=3,k-1,l} \leq 1, \quad \forall k \neq 1, p \in P \tag{18}$$

$$\sum_{i\in I} W_{ip} \sum_{i\in I} \sum_{j\in \Upsilon_j} \sum_{l\in L} x_{ijkl} \leq 1, \quad \forall k = 1, p \in P \tag{19}$$

$$\sum_{i\in I} W_{ip} \sum_{i\in I} \sum_{j\in \Upsilon_j} \sum_{l\in L} x_{ijkl} + x_{i,j=4,k-1,l} \leq 1, \quad \forall k \neq 1, p \in P \tag{20}$$

$$\sum_{i\in I} W_{ip} \sum_{i\in I} \sum_{j\in \Pi_j} \sum_{l\in L} x_{ijkl} \leq 1, \quad \forall k = 1, p \in P \tag{21}$$

$$\sum_{i\in I} W_{ip} \sum_{i\in I} \sum_{j\in \Pi_j} \sum_{l\in L} x_{ijkl} + x_{i,j=5,k-1,l} \leq 1, \quad \forall k \neq 1, p \in P \tag{22}$$

$$x_{ijkl} \in \mathbb{Z}, \quad \forall i \in I, j \in J, k \in K, l \in L \tag{23}$$

**Table 2** Time period of a day on which a teaching group may be scheduled

| Timeslot |
|---|
| 7:00-8:30 |
| 8:30-10:00 |
| 10:00-11:30 |
| 11:30-13:00 |
| 13:00-14:30 |
| 14:30-16:00 |
| 16:00-17:30 |
| 17:30-19:00 |
| 19:00-20:30 |
| 20:30-22:00 |

## 4 Spreadsheet-based approach

The previously shown model was constructed in a spreadsheet in a smaller instance and was then scaled to the real size of the studied case. Because of the limited capacity of the excel solver, an Open Solver (external branch and cut solver) was used as a plug in to solve the model and it successfully did in a cpu time of 12 hrs. Once the model was built in a spreadsheet and proved to work, an interface was built used to load the information (feasibility matrix and teacher matrix) into the model so any user can use the model without having any previous linear programming knowledge.

In this section some figures of the spreadsheet are shown including the following:

- Feasibility matrix: A set of binary values loaded by the user trough the interface which shows with the value of 1 the schedules that are feasible for each teaching group.
- Professor Matrix: A set of sums and products (with the loaded parameters) which recreates the teachers restrains.
- Classroom matrix: A set of sums which recreates the classroom restrains.
- Assignment professor: A set of binary values loaded by the user trough the interface which shows the teacher assigned to each teaching group with the value of 1.
- Developed tool: The developed interface used to load the parameters into the model.
- Model on OpenSolve: The model and the solver tool used to solve it.

The figures show a small part of each matrix only for the reader to understand how it was that the model was built in the spreadsheat. In table 3 we can see the different characteristics of the model (|I|), (|J|),(|K|), (|L|) and (|P|). (See Sect. 3). The matrix of the fig.1 represents a two-dimensioned arrange, whhere the rows are (|I|) and the columns the different combinations of (|J|),(|K|) and (|L|).

**Table 3** Parameter's instances

| INSTANCE | (\|I\|) | (\|J\|) | (\|K\|) | (\|L\|) | (\|P\|) |
|---|---|---|---|---|---|
| INST-1 | 10 | 15 | 5 | 3 | 1 |
| INST-2 | 20 | 5 | 10 | 3 | 3 |
| INST-3 | 120 | 5 | 10 | 3 | 3 |
| CASE STUDY | 500 | 19 | 10 | 3 | 130 |

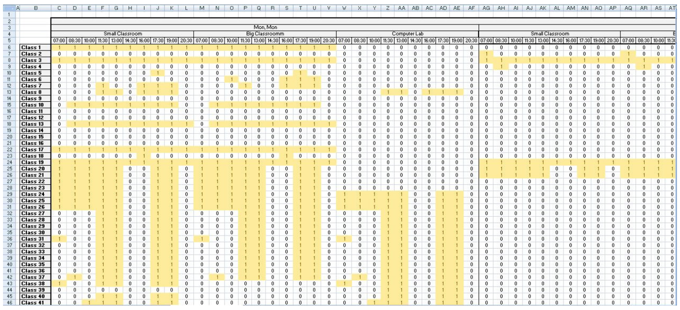

**Fig. 1** Feasibility matrix

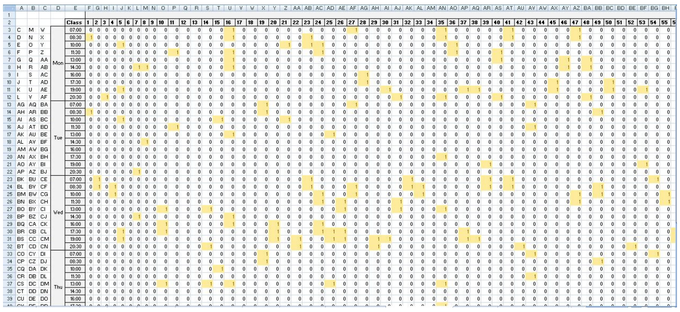

**Fig. 2** Professor matrix

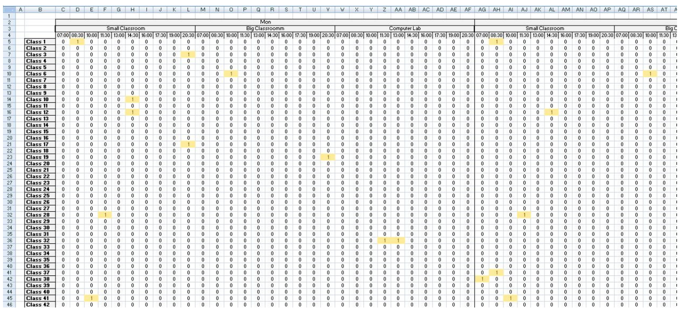

**Fig. 3** Classroom matrix

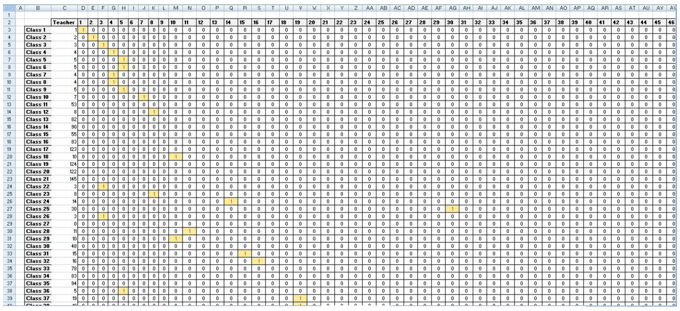

**Fig. 4** Assignment professor matrix

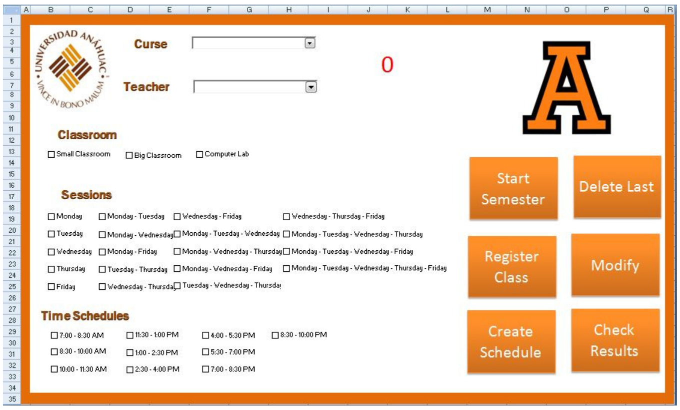


**Fig. 5** Tool developed

## 5 Conclusions

A timetabling is effective when the the solution proposed is feasible and may be performed by the university. In our case of study, the solution obtained is a good feasible solution to schedule the subjets in the classrooms.

During this project we succeeded in developing a tool in a spreadsheet using mathematical programming that is capable to generate an allocation that maximizes the number of classes that can be delivered, subject to their respective constrains which are: teachers, schedules and classrooms.

The developed tool successfully solves the actual problem of timetabling for the Faculty of Engineering of Universidad Anahuac. The developed tool found a feasible solution and in our case study were still solvable by open solver.

One of the advantages of this tool is that can be used by anyone with basic knowledge of using a spreadsheet without having any familiarity with mathematical programming or understanding the developed model.

As future modifications to the project we propose the following:

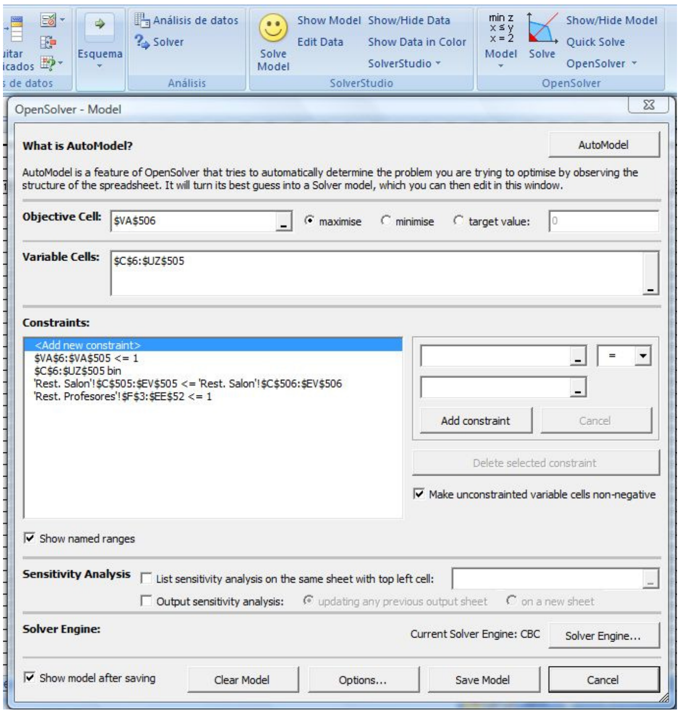


**Fig. 6** Open Solve

1. Adding to the tool the automatic assignment of a specific classroom, not only the classroom type. This can be done with spreadsheet programming and no modifications to the model.
2. Adding a restrain similar to the teachers restrain that would allow the user to assign a number to any teaching group and restrain groups with the same number not to be scheduled at the same time. This could be used for classes that have to be taken by the students on the same semester.
3. Use linear programming software to solve the model while you keep the interface and generate the parameters on the spreadsheet. A plug in called SolverStudio is very helpful for this king of interface and will possibly by used by the authors on this project in the future.
4. Adding more set of day options to make the model more flexible for the user including a 1 session option for each day of the week.

## References


1. Al-Betar, M.A., Khader, A.T., Zaman, M.: University course timetabling using a hybrid harmony search metaheuristic algorithm. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **42**(5), 664–681 (2012). DOI 10.1109/TSMCC.2011.2174356
2. Daskalaki, S., Birbas, T., Housos, E.: An integer programming formulation for a case study in university timetabling. European Journal of Operational Research

**153**(1), 117 – 135 (2004). DOI http://dx.doi.org/10.1016/S0377-2217(03)00103-6. URL http://www.sciencedirect.com/science/article/pii/S0377221703001036
3. Even, S., Itai, A., Shamir, A.: On the complexity of timetable and multicommodity flow problems. SIAM Journal on Computing **5**(4), 691–703 (1976). DOI 10.1137/0205048. URL http://dx.doi.org/10.1137/0205048
4. McCollum, B., Burke, E.K.: The practice and theory of automated timetabling. Annals of Operations Research **218**(1), 1–2 (2014). DOI 10.1007/s10479-014-1614-6. URL http://dx.doi.org/10.1007/s10479-014-1614-6
5. Nandhini, M., Kanmani, S.: Implementation of class timetabling using multi agents. In: Intelligent Agent Multi-Agent Systems, 2009. IAMA 2009. International Conference on, pp. 1–2 (2009). DOI 10.1109/IAMA.2009.5228065
6. Obit, J.H., Ouelhadj, D., Landa-Silva, D., Vun, T.K., Alfred, R.: Designing a multi-agent approach system for distributed course timetabling. In: Hybrid Intelligent Systems (HIS), 2011 11th International Conference on, pp. 103–108 (2011). DOI 10.1109/HIS.2011.6122088
7. Pedroso, A.: School timetabling: A case in large binary integer linear programming. Management Science **30**(12), 1473–1489 (1984). DOI 10.1287/mnsc.30.12.1473. URL http://dx.doi.org/10.1287/mnsc.30.12.1473
8. Pedroso, J.: A multi-agent system for automated timetabling with shared resources. In: J. Cha, R. Jardimgoncalves, A. Steigergarcao (eds.) Concurrent Engineering: Advanced Design, Production And Management Systems, pp. 359–363 (2003)
9. Qualizza, A., Serafini, P.: Practice and Theory of Automated Timetabling V: 5th International Conference, PATAT 2004, Pittsburgh, PA, USA, August 18-20, 2004, Revised Selected Papers, chap. A Column Generation Scheme for Faculty Timetabling, pp. 161–173. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
10. Strnad, D., Guid, N.: A multi-agent system for university course timetabling. Applied Artificial Intelligence **21**(2), 137–153 (2007). DOI 10.1080/08839510601147554
11. Timothy, R., Gary, M.: Schedulexpert: Scheduling courses in the cornell university school of hotel administration. Interfaces **32**(6), 45–57 (2002). DOI 10.1287/inte.32.6.45.6477. URL http://pubsonline.informs.org/doi/abs/10.1287/inte.32.6.45.6477
12. de Werra, D., Asratian, A.S., Durand, S.: Complexity of some special types of timetabling problems. Journal of Scheduling **5**(2), 171–183 (2002). DOI 10.1002/jos.97. URL http://dx.doi.org/10.1002/jos.97