A Simple CAD System for Faculty Timetabling Design

Andrei Rusu, Elena Rusu

Abstract

A computer-aided design (CAD) system is proposed in order to help in designing the timetabling of courses in a faculty. It is based on already used technologies such as Google Spreadsheets and Microsoft Excel and tools built-in in them: Server JavaScript, Visual Basic for Applications (VBA). The idea is to use a single sheet "Collision" in a spreadsheet in Microsoft Excel to deal with all information related to the timetable, and then, by parsing its cells, to generate all needed timetables for students, teachers, rooms, etc.

Keywords: Timetabling, CAD systems, VBA, JavaScript, mathematical models.

1 Introduction

The development of a timetable for courses in a faculty is a difficult task that in our days, in many faculties in Moldova, are still done in an old manner using sheets of paper for groups of students, for professors, and for rooms. Any single record related to the timetable of a group of students, or of a professor, or a room requires to make the corresponding modifications in all related timetables for students, for professors, and for rooms.

There are many available software systems on the market which can help to automatize the process to obtain quickly a qualitative timetable for the courses in a faculty, but for different reasons they are not used.

©2024 by Andrei Rusu, Elena Rusu

Among reasons we meet the following: the systems are difficult to use, they are costly, they do not complain with the requirements in the faculty, the result is not one that is expected, and so on. The surveys for timetable for universities are available in [1, 2].

The goal of the present paper is to present a relatively simple CAD system based on tools already known by the majority of faculty members, such as Microsoft Excel and Google Spreadsheets together with their built-in tools, such as VBA (Excel) and JavaScript (Google Docs).

2 The problem

The timetabling problem is a problem related to to the optimal allocation of resources and it has different variations. We consider the following one (a usual case in a faculty in Moldova or Romania). We consider for simplicity that all course activities are organized in the same way during a week.

There are some rooms $S = \{S_1, \ldots, S_{N_S}\}$, each one with some properties, such as number of seats, available projector, number of available computers, etc.

There are also some study groups of students $G = \{G_1, \ldots, G_{N_G}\}$.

There are some professors $P = \{P_1, \ldots, P_{N_P}\}$.

Course activities take place in any week-day $Z = \{Lu, Ma, Mi, Jo, Vi, Sa, Du\}$, and in some standard time interval $T = \{8-10, 10-12, 12-14, 14-16, 16-18, 18-20\}$. By a timeslot we understand the tuple of a given day and a given time interval.

There are also a list of courses $D = \{D_1, \ldots, D_{N_D}\}$ and type of activity associated to such courses $\{Course, Seminar, Laboratory\}$ that are specific (are allocated) to each group of students.

Each professor is allocated to teach some courses of the corresponding type to some groups of students. Each such tuple $\{(P_i, G_j, D_k)\}$ is called a timetable event or teaching activity.

The problem is to allocate day, time interval, and room to each timetable event such that it is complained with the so-called hard restrictions: (a) each professor is involved at most in one event for a

given timeslot, (b) each group of students is involved at most in one event for a given timeslot, (c) at most one event happens in a room at a given timeslot.

There are also so-called soft restrictions which are related to the preferences of students, professors, and administration. Often some soft restrictions are actually hard ones (preferences of administrations, availability of some professors, etc).

The problem is to find out a solution that respects all hard restrictions and as many as possible soft restrictions.

3 Proposed solution

The proposed solution is to use google spreadsheets, and register all events in a sheet "Collision", each column of which is the room, and each row is the timeslot (see Fig. 1).

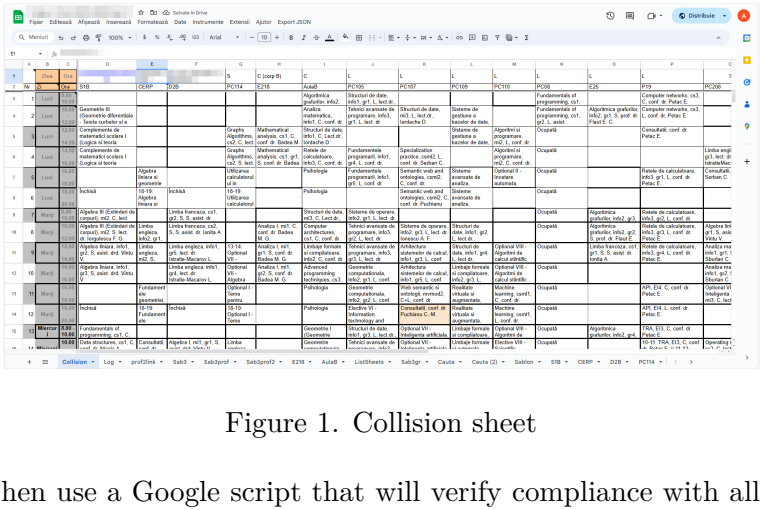


Figure 1. Collision sheet

Then use a Google script that will verify compliance with all your restrictions every time you register an event in Collision. Thus you finally get a feasible timetable. Then download the spreadsheet and use VBA to parse the Collision and generate all necessary timetables for students, for professors, by rooms, etc.

4 Conclusion

The proposed system is relatively simple to implement in a faculty, but it is not suitable to be used at a university level, or for other timetabling systems. The system is designed to help with the construction of the timetable, but it can be also adapted to automate some more things based on methods described in [2].

Acknowledgments. Supreme Council for Science and Technological Development has supported part of the research for this paper through the research project 011301 "CS systems based on AI".

References

[1] Hamed Babaei, Jaber Karimpour, and Amin Hadidi. *A survey of approaches for university course timetabling problem.* Computers & Industrial Engineering, vol. 86, pp. 43–59, 2015.

[2] M. C. Chen, S. N. Sze, S. L. Goh, N. R. Sabar, and G. Kendall. *A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities.* IEEE Access, vol. 9, pp. 106515–106529, 2021, doi: 10.1109/ACCESS.2021.3100613.

Andrei Rusu[1,3], Elena Rusu[2]

[1]Moldova State University, Vladimir Andrunachievici Institute of Mathematics and Computer Science
E–mail: andrei.rusu@math.md
ORCID: https://orcid.org/0000-0002-0259-3060

[2]Technical University of Moldova, Dep. of Mathematics
E–mail: elena.rusu@mate.utm.md
ORCID: https://orcid.org/0000-0002-2473-0353

[3]Ovidius University of Constanţa, Dep. of Mathematics and Informatics
E–mail: andrei.rusu@365.univ-ovidius.ro