



Research into a timetable forming tool for final classes

L. Pupeikienė, M. Rybokas

Lina Pupeikienė is with the Department of Information Technologies, Vilnius Gediminas Technical University, Vilnius, Lithuania

(phone: 3705-274-4832; e-mail: lina.pupeikiene@vgtu.lt).

Mindaugas Rybokas is with the Department of Information Technologies, Vilnius Gediminas Technical University, Vilnius, Lithuania

(e-mail: rybokas@vgtu.lt).

Abstract

In this paper a high-school timetable forming tool is described. Commercial products and software for this problem solving are compared. The problems of timetable forming are analyzed. A decision for initial data computing is described. A new tool for computing problems solution is described. The results of this tool are analyzed.

Keywords—high-school, initial data computation, timetable tools, timetable.



Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 9, No 2

editor@cirworld.com

www.cirworld.com, member.cirworld.com



INTRODUCTION

Now a days, making timetables is a very important task. All of us grow up to a schedule: wake up on time, eat on time, learn on time, play in time, etc. As we can see to make a good schedule for a day is a difficult and a rather complicated task [1].

To make a good school schedule is very important for all school members: teachers, pupils, and staff. The complexity of a school schedule task depends on the number of school members and the number of classrooms. A solution of this task is different in every country [2]. In this paper I will talk about the school scheduling task in Lithuania. The most complex task in Lithuanian schools is to make a good schedule for final classes.

PROBLEM DESCRIPTION

The final classes in Lithuanian schools are two last classes: 11th and 12th or III and IV gymnasium classes. In this paper they will be called 1st and 2nd level classes. Every pupil must learn between 28 and 36 hours per week at school [3]. Depending on the school, they have between 3 and 5 compulsory subjects. And they take between 10 and 18 hours. The remaining hours are occupied by elective subjects. In a school a list is formed for possible subjects and hours. Every pupil can choose the elective subjects from the list can choose subjects and times as they wish. As the result, every pupil has almost individual schedule. All pupil wishes are fixed and can be transformed to an XML format file. An example of pupil' wishes is displayed in Fig. 1.

Pupils of final classes must make their lists of wishes for two years. However, the schedule is created only for the current year. The program must remember all wishes of the pupils for the second year, while next year these data must be used for creating a new school timetable. It is very important to keep SubGroups of the first year if possible. So the program must remember the last timetable to create a timetable for last year's pupils, to consider wishes of new pupils for the last two years and to create timetables for pupils of both levels.

	A	B	C	D	E	F	G	H	I	J	K	L	M
	No.	Name	subj_1	subj_2	subj_3_a	subj_3_b	subj_4_a	subj_4_b	subj_5_a	subj_5_b	subj_6_a	subj_6_b	subj_7_a
1													
2	1	Pupil 1	2		5					3			3
3	2	Pupil 2	1	1	5					3			
4	3	Pupil 3		2	5		4						4
5	4	Pupil 4		2	5					3			3
6	5	Pupil 5		2	5					3			3
7	6	Pupil 6		2	5		4						3
8	7	Pupil 7	2		5				4				
9	8	Pupil 8		2	5		4						3
10	9	Pupil 9		2	5					3			
11	10	Pupil 10	1	1	5					3			
12	11	Pupil 11		2	5					3			
13	12	Pupil 12	2		5			3					
14	13	Pupil 13		2	5					3			3
15	14	Pupil 14	2		5				4				3
16	15	Pupil 15		2	5				4				3
17	16	Pupil 16		2	5		4						4
18	17	Pupil 17	2		5				4				3
19	18	Pupil 18		2	5					3			4
20	19	Pupil 19		2	5					3			3

Fig. 1. Part the table of pupils wishes.

In schools pupils learn in groups. One group can contain between 5 and 35 pupils. Pupils can choose different subjects (like subj_1 or/and subj_2, etc.) or different subject by the level (like subj_3_a or subj_3_b), and times of these subjects. Subjects with different levels are considered as different subjects. All pupils must be grouped by subject and subject times. Accordingly, these groups are different for every subject, and they are called SubGroups.

ANALYSIS OF OTHER PROGRAMS

In Lithuania schools use two programs: „MIMOSA”[7] or/and „aSc TimeTables”[4]. Both programs are not good enough to solve the timetable task of the final classes.

A. aSc TimeTables

This program is a product of the Slovak company „Applied Software Consultants s.r.o.” It has a very nice interface. It is

very easy put the data into the program, but it is impossible to enter wishes of pupils. There is no possibility to write one pupil and his choices, and the program cannot calculate the real number of pupils per lesson. And to make the schedule is impossible.

There is another way to enter data into the program. Pupils formally belong to some groups in the school. These groups are called Classes. In one Class there can be between 20 and 35 pupils. And all of them can have individual schedule made by the wishes. It means that all Class can be split into 20-35 groups. Therefore, if a user wants to split the Class into more than 6 small groups, the program has no such capability. And it makes it impossible to write all pupils wishes and make correct schedule.

This program is very good if a school is small (up to 200 pupils), or if a school does not let pupils choose their subjects.

B. MIMOSA

This program is a product of the Finnish company „Mimosa Software Ltd.“ It has a complicated interface, and it is difficult to understand for a user beginner. It is very tricky to change something in case of mistakes. This is the main reason why this program is not widespread in Lithuanian schools. Final schedules are acceptable for every user. There are personal schedules for every teacher, pupil, classroom, etc. But optimization options are limited.

IDENTIFYING NEEDS OF TIMETABLE MAKERS

A survey of initial data grouping has been performed. The aim was to find out how users want to group the initial data in the program: manually, automatically (by means of a random function), or in a mixed manner (some data entered manually and some automatically). The survey involved 150 schools. As shown in Fig. 2, there are 141 schools (94%) who wanted to work with the mixed method for grouping the initial data. Only 6 schools (3%) wanted to work with the manual method and 3 schools (2%) wanted to work with the automatic method.

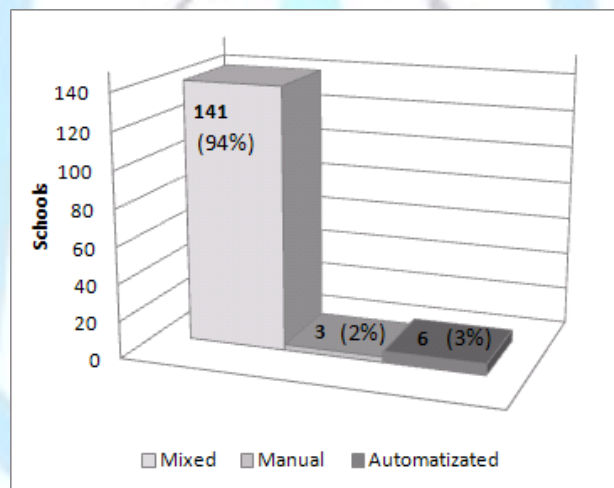


Fig. 2. Survey results of the schools

It is so because users can take all data from the previous years. And it is very easy to make a new schedule for the new school year.

After analysing the survey results it was decided to create the tool. This tool is created to help teachers to make the SubGroups for each subject. Section V contains more description of this tool. Some screenshots of this SubGroup making tool are shown in Fig. 3 and Fig. 4.

The reasons for such results have been analysed too. The fully automatical method was proposed and can be very good, since it can save very much time for creating SubGroups and assigning teachers to them. But this method was not suitable because:

- 1) The program does not know how to group pupils of the 1st level classes.
- 2) The program does not know how to group or regroup pupils of the last year classes.
- 3) If users want to change something in the subgroup (teacher, list of pupils etc.) – there are no possibilities. The subjective factors and so called soft constrains are important in this step.

The fully manual method was proposed and can be very good, while users group all SubGroups manually and assign teachers to them. But this method was not suitable because:

- 1) It takes a lot of time to group pupils of two levels.
- 2) There are no possibilities to copy part of one group to another subgroup.
- 3) There are no possibilities to form SubGroups of the previous year.
- 4) Users must check manually what teachers were teaching pupils and try to group all such pupils into a subgroup.

Fig. 3. Creation of a new subgroup

The mixed method was chosen by the users because:

- 1) Users group the 1st level SubGroups manually and assign teachers (from the existing teachers list) to them (Fig. 3).
- 2) It takes less time to group pupils for the 2nd level year, while the program automatically takes all SubGroups from the previous year. If these SubGroups are not full, users can manually fill empty places (Fig. 4).
- 3) Users only check teachers for the 2nd level classes. If there are some changes, users can easy make them.
- 4) The method works slower then the fully automatic, but much faster then the fully manual.

Fig. 4. Modification of a subgroup

When SubGroups are created, classrooms are known, and users can make a school time table for the 1st and 2nd year pupils. To make a timetable is a tricky task. Therefore, in every school there are a bit different rules for making timetables. Hard constrains are the rules that are the same in all schools. Soft constrains are the rules that can be

different in every school.

C. Surveying for hard constrains

Timetable makers' survey was performed to determine what hard constrains are valid for all schools in Lithuania. The survey involved 150 timetable users and schools. The result of the survey data are presented in Fig. 5.

In section 4.2 these results are described in more detail.

D. Rules for schedule making

It is a difficult combinatorial problem [5]. The school time table d can be described as a binary four-dimensional array representing decisions of timetable makers [6]:

$$d = [d_{m,p,r,t}]_{M \times P \times R \times T}$$

where M is a set of teachers, P is a set of SubGroups, R is a set of classrooms, and T is a set of weekly time-slots. If $d_{m,p,r,t}=1$ it means that the lesson of the teacher m attended by the subgroup p in the room r at the time t . If $d_{m,p,r,t}=0$ it means there is no lesson.

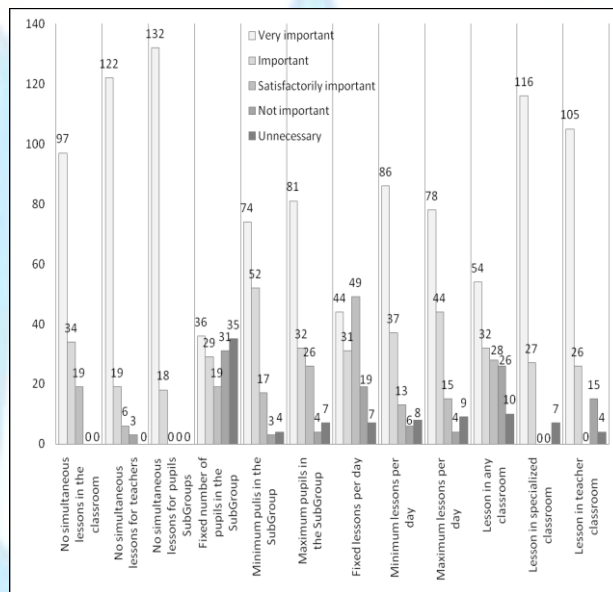


Fig. 5. Results of importance of hard constrains

There are some rules to get correct schedule. Rules can be divided in two types: rules that must be respected (hard constrains) and rules that are desirable to follow (soft constrains).

Hard constrains are:

- 1) There cannot be fewer lessons than S_{min} (condition (2)) and more lessons than S_{max} (condition (3)). Here I is a set of week-days, T_i denotes a set of time-slots of the i -th week-day [8].

$$h_1(d) = \sum_{m,p,r,t \in T_i} d_{m,p,r,t} \leq T_{max} \text{ for all } r, i \in I \tag{2}$$

$$h_2(d) = \sum_{m,p,r,t \in T_i} d_{m,p,r,t} \geq T_{min} \text{ for all } r, i \in I \tag{3}$$

- 2) No simultaneous lessons in the classroom (condition (4)).

$$h_3(d) = \sum_{m,p} d_{m,p,r,t} \leq 1 \text{ for all } r, t \tag{4}$$

- 3) No simultaneous lessons for teachers (condition (5)).



$$h_4(d) = \sum_{m,r} d_{m,p,r,t} \leq 1 \text{ for all } p,t \quad (5)$$

4) No simultaneous lessons for pupils' SubGroups (condition (6)).

$$h_5(d) = \sum_{p,r} d_{m,p,r,t} \leq 1 \text{ for all } m,t \quad (6)$$

5) There cannot be fewer pupils in a subgroup than G_{min} (condition (7)) and more pupils than G_{max} (condition (8)).

$$h_6(d) = \sum_p d_{m,p,r,t} \geq G_{min} \quad (7)$$

$$h_7(d) = \sum_p d_{m,p,r,t} \leq G_{max} \quad (8)$$

6) All lessons are specialized. All classrooms J are specialized too. Therefore, lessons must be in a special type of classroom j . Sets of specialized classrooms list is called R_j . The number of all available classrooms j is $R_{j,max}$ (condition (9)).

$$h_8(d) = \sum_{m,p,r \in R_j} d_{m,p,r,t} \leq R_{j,max} \text{ for all } t,j \in J \quad (9)$$

Pupils accept only subjects and hours. They do not accept teachers. Teachers are assigned by hand for every subgroup. What teacher teaches certain SubGroups depends on teacher's qualification, experience and other subjective and objectives causes and school's internal rules of procedure. These rules are called soft constrains.

The list of soft constrains is still in the investigation level. Most popular soft constrains are:

- 1) Teachers want to have some free days;
- 2) The structure of a subgroup cannot change all week;
- 3) Pupils do not want to have free lessons;
- 4) Teachers do not want to have free lessons;
- 5) If a teacher have his or her own classroom, the lessons he/she wants to have in this classroom;
- 6) Etc.

These and other soft constrains will be analyzed in the future papers. The research is conducted on these needs.

TIMETABLE FORMING TOOL

After the analysis of teacher's survey, it was decided to create the timetable forming tool. This tool does not rely on a specific operating system. It is a web-based tool so users can work with it from any computer that has an internet connection. A timetable forming tool works on three administrative levels:

- 1) Highest-level administrator – it can create/connect new schools, delete old (if it is necessary) and add/edit/delete all information for the school: teachers, subjects, pupils, places etc.
- 2) Medium-level administrator – it can only administrate the assigned school. It can add/edit/delete all information about the current school, teachers/pupils/subjects in this school, time etc.
- 3) Lowest-level administrator – this includes pupils, parents of these pupils and teachers. Pupils can view the personal information about him/her. He/she can check the values, schedule, lessons time and other information and to choose possible subjects for one or two years. Teachers can view their personal information, check schedule, pupils in the SubGroup and write proposals for school administrator. Parents can view information about the child, notes from teachers, values, lessons time and other information connected with the child.

This paper deals mostly with the medium-level administrator working fields.

The tool stores all information about every pupil, teacher, possible lessons time, possible subjects and pupils choices of these subjects. The first step is to create a new school year. This is very important. Later, every year the user must only accept choices of these pupils and make SubGroups. It is easier to take in already existing pupils at the school and create new SubGroups. Therefore, it is very easy to compare a SubGroup from the past year with a SubGroup of the current year. If it is necessary, the user can copy the SubGroups from the past year to the current year.



The second step is to fill a list of the subjects for every possible subject for every level pupils of the current school year. The user can copy a list of these subjects from the past school year to the current year or to make a new list. This is very useful, since it helps to save the time for editing subjects for every pupil's level.

The third step can be performed in two ways:

- 1) The pupil chooses possible subjects from the list and makes an individual learning plan. In this way the pupil can make a personal learning plan and to calculate all possible decisions. The medium-level administrator will get ready plans, and will accept them and only create SubGroups.
- 2) The medium-level administrator chooses possible subjects from the list and makes individual learning plans for every pupil. In this way the medium-level administrator will have a lot of work and will not calculate the best learning plan. When the medium-level administrator creates an individual pupil's learning plan, he/she can only accept it. Therefore, the pupil cannot make any changes in this plan. Only the school medium-level administrator can make changes in these plans.

The fourth step is to fill the teacher list. The medium-level administrator should write teacher's name, last name and the subjects that he/she can teach. Of course, this information can be filled by the teacher personally. Therefore, this information can be changed any time by the teacher. This information will be necessary while forming the SubGroups.

The fifth step is to fill SubGroups. A few forms of schedule forming tool are shown in Fig. 3 and Fig. 4. SubGroups are filled by the medium-level administrator. He/she splits pupils of one level to groups in the following way:

- 1) Subject name – a subject that pupils have chosen,
- 2) Teachers name – a teacher who can teach the subject,
- 3) List of pupils – a list of pupils names, who have chosen same subject.

When a SubGroup is formed, the medium-level administrator can give the name to this group. The list of same subject pupils is decreasing (pupils from the formed SubGroup are gone from the list). Teachers work hours are automatically increasing (these hours are coded in the subject name). And both functions work automatically when the medium-level administrator saves the SubGroup with a certain name.

Therefore, it is easy to control a SubGroup structure, and it is easy to calculate teachers work hours. After forming all SubGroups, the medium-level administrator can see what pupils are left (who do not belong to any SubGroup). It can happen that it is impossible to create some SubGroup. The medium-level administrator can warn such pupils and let them choose other subjects.

It is easy to notice what subject teacher is necessary. Therefore, this tool helps to administrate the personnel: see what kind of specialist is needed or not needed for the current school year.

The next step will be to create a timetable from the SubGroups. In this step hard constrains are very important; they are described in Section IV Part B. Soft constrains are important too but not obligatory. This is the subject for future research. The solution of the timetabling problem will be attempted using the graph theory [9] and optimization methods [10].

CONCLUSION

The survey of the timetable makers has demonstrated that:

- 1) The tool for subgroup creation should work in a mixed manner: automatically with manual techniques.
- 2) It is useful to design an interface for easy and comfortable SubGroups forming.
- 3) It is necessary to establish hard constrains for initial timetable making.

The tool works online and the data is saved on a server. That enables work from any computer that has an internet connection.

REFERENCES:

- [1] Abramson, D. (1991). Constructing school timetables using simulated annealing: Sequential and parallel algorithms. *Management Science*, 37, 98–113.
- [2] A. Didem, C. Sen (2011). Rule-based Genetic Algorithm for In-Service Training Curriculum Plan. *Recent Advances in Computers, Communications, Applied Social Science and Mathematics*, ISBN: 978-1-61804-030-5, pp. 160-166
- [3] L. Pupeikiene, J. Mockus, (2005). School schedule optimization program. *Information Technology and Control*, 34, 161–170.
- [4] ASC (2010). Timetables ASC. <http://www.asctimetables.com/>.
- [5] D. Kalpic, T. Rajnovic, V. Mornar (2005). Proceedings of the 1st WSEAS / IASME Int. Conf. on EDUCATIONAL TECHNOLOGIES, Tenerife, Canary Islands, Spain, December 16-18, 2005, pp57-63



- [6] J. Mockus, L.Pupeikienė (2012). On Multi-Start Algorithms for Optimization of High School Timetables. *Informatica*, Vol. 23, No. 3, 2012, pp. 399-419
- [7] Mimoso (2012). MIMOSA Scheduling Software. <http://www.mimosoftware.com/>
- [8] L. Pupeikiene, J. Mockus, (2010). School Scheduling Optimization, Investigation and Applications. Lambert Academic, Saarbrücken.
- [9] T. Redl (2009). On Using Graph Coloring to Create University Timetables with Essential and Preferential Conditions. *Advances in Marketing, Management and Finances*, ISSN: 1790-2769, pp. 162-167
- [10] Nawat Nuntasen, Supachate Innet (2007). A Novel Approach of Genetic Algorithm for Solving University Timetabling Problems: a case study of Thai Universities, 7th WSEAS International Conference on APPLIED COMPUTER SCIENCE, Venice, Italy, November 21-23, 2007, pp. 246-252

