HIGH SCHOOL TIME TABLE PROBLEM SOLVING AND COMPARISON WITH AUTOMATIC OPTIMIZED TIMETABLE

Lina Pupeikiene¹, Jonas Mockus²

¹Vilnius Gediminas Technical University, Sauletekio al. 11, LT-10223 Vilnius, Lithuania, Lina.Pupeikiene@gmail.com
²Institute of Mathematics and Informatics, Akademijos g. 4, LT-08663 Vilnius, Lithuania, JMockus@gmail.com

Abstract. In every high school it is most important to build school schedule for high school pupils. Every pupil of high school can choose individual lessons and has individual schedule. The problem is more complicated when the every pupil has possibilities to choose not only subjects, but hour per week of this subject too. However, as the number of teachers, number of pupils, number of different subjects, number of different subject hours, time slots and the constraints increases, the required time to find at least one feasible solution grows exponentially. All pupils, which have the same subject, are grouped to the subject-groups. Every subject-group has own teacher. The high school schedule is created from these subject-groups. The article includes such aims: development and estimation of real high school timetables program; comparison results of real high school schedule and automatically optimized scheduled.

Key words: High school timetabling, optimal scheduling, subject-groups forming, heuristics evaluation.

1 Introduction

A timetable specifies which people meet at which location and at what time. The timing of events must be such that nobody has more than one event at the same time. School timetabling as a term refers to the construction of weekly timetables for schools of secondary education [14]. Specific feature of school timetabling field is a great number of research papers and widely used commercial software. Therefore a discussion of new results will be.

The events are lessons in a subject, taught by a teacher to a group of pupils in a single room. The timetable assigns a teacher, a pupils group, a room, and a time slot to each lesson. The pupil groups are specific to the subject, we call them subject-groups. A high school is referred here as the last grades of a high school or gymnasium where the pupils can mostly choose their preferred learning profile subjects. Therefore, this task is more complex in comparison with a secondary school scheduling without high school classes.

Some combinations of assignments lead to acceptable timetables, constraints follow from conditions imposed by rooms, pupils or teachers. We distinguish two types of constraints: conditions that must be met (“hard” constraints) and desires that should be fulfilled as well as possible (“soft” constraints). An important set of soft constraints is defined by didactic reasons. For example, by placing “hard” subjects, such as mathematics or physics, into morning hours. The maximal number of daily hours Tmax is obviously a hard constraint. Timetabling can be generally defined as the activity of assigning, subject to constraints, a number of events to a limited number of time periods and locations such, that desirable objectives are satisfied as nearly as possible [26]. Educational timetabling can be divided into three main classes: school timetabling, course timetabling and exam timetabling [15]. The goal is to find a timetable that satisfies all the hard constraints and minimizes the violation of soft constraints.

2 Overview of publications

A survey on educational timetabling problems [23] gives an overview of the literature. Overviews on examination timetabling and university course timetabling are in [4, 12, 13]. A comprehensive overview of formulations and of state-of-the-art approaches is in the surveys [4, 7, 8, 13, 15], in the proceedings of the PATAT conferences [5 – 7, 9, 10] and in the Lecture Notes in Computer Science series [9 – 11]. The European working group on automated timetabling (EURO-WATT) maintains a website with information on timetabling problems [25].

3 New Elements

The first new element of this work is the application and systematic investigation of the Bayesian Heuristic Approach [20] for optimization of heuristic parameters. These include the initial temperature and the cooling rate of Simulating Annealing (SA) algorithm and the randomization parameter of the local search algorithm. The formulation of the objective function in terms of Pareto optimality seems to be new in the field of school scheduling. The paper describes apparently the first web-based platform-independent implementation of
the software. Java servlet provides conditions for application at any school with internet connection. Any web browser works, no additional software is needed. Note that efficiency of recent versions of Java is close to that of the most efficient programming languages [9].

4 Defining Optimization Problem

Ministry of Education of the Republic of Lithuania has confirmed basic rules for high school schedule forming. They can be complementary of each school's rules and restrictions. However, the main purpose of these limitations is to develop a schedule, which would evaluate of the Ministry of Education requirements. In addition, this schedule must be acceptable to both: pupils and teachers.

Required schedule restrictions (formed by the Ministry of Education):
1. Working days \(d\) per week must be \(d \leq 5\).
2. The teacher simultaneously cannot work in several different places.
3. The teacher cannot have more than 36 hours per week.
4. The pupil simultaneously cannot learn few different subjects.
5. A pupil \(i\) may have \(28 \leq i \leq 32\) lessons per week.
6. It cannot be more then \(p \leq 7\) lessons \(p\) per day.
7. Number of pupils \(i\) in one subject-group can be \(15 \leq i \leq 30\).
8. In each classroom simultaneously cannot be several different types of subjects (for example, mathematics and physics).
9. Subjects, requiring special measures or facilities, shall be taught in the special classrooms (for example, IT, chemistry etc.).

Technically any required restriction violations cannot be broken. There can be only some minor offenses necessary restrictions, if it significantly improves the quality of the schedule. To define with timetable is good or bad we use penalty points. The penalty point’s \(c_r\), which assessing these restrictions, should be imposed very strictly.

The main required penalty point’s restrictions function is as follows:
\[
F_f = \sum_r c_r N_r, \tag{1}
\]
here \(c_r\) – penalty for required restriction \(r\); \(N_r\) – number of required restriction. In this case \(r = 1, ..., 9\).

Some of required restrictions \(c_r\) can be evaluated by the individual rules of each school. Such requirements are called needful, or “soft” constrains. They are valued differently in each school.

The main needful restrictions of the schedule include:
- Elimination of “windows” in teacher’s schedule.
- Elimination of “windows” in pupil’s schedule.
- Unacceptable working hours.
- Unacceptable workdays.
- Unacceptable order of subjects.
- Changing of pupils in the formed subject-group.

Usually penalty points for these restrictions are as follows:
\(c_m\) – penalty for the “window” on teacher’s \(m\) schedule.
\(c_s\) – penalty for the “window” on pupils \(s\) schedule.
\(c_{mv}\) – penalty for “bad” hour \(v\) of teacher \(m\).
\(c_{md}\) – penalty for “bad” day \(d\) of teacher \(m\).
\(c_{sv}\) – penalty for “bad” hour \(v\) of pupil \(s\).
\(c_{pd}\) – penalty for violation of pedagogical didactic \(pd\).
\(c_{mg}\) – penalty of the list change of subject-group \(g\) taught by teacher \(m\).

“Bad” hour/day is the hour/day, when teacher/pupil already has a work hour. Pedagogical didactic evaluates the difficulty of subjects. Most difficult subjects must be in the 1-4 lessons during the day. Less important subjects – in the end of the day. The importance of every subject is written in initial data file.

The sum function of the needed restrictions penalty points is as follows:
\[ F_n = \sum_{m} c_{m} L_{m} + \sum_{s} c_{s} L_{s} + \sum_{m} \sum_{v} c_{mv} L_{mv} + \]
\[ + \sum_{m} \sum_{d} c_{md} L_{md} + \sum_{s} \sum_{v} c_{sv} L_{sv} + \sum_{pd} c_{pd} L_{pd} + \sum_{n} c_{n} L_{n}, \]

(2)

here \( L_{m} \) – number of “windows” on teachers \( m \) schedule; \( L_{s} \) – number of “windows” on pupils \( s \) schedule; \( L_{mv} \) – number of “bad” hours \( v \) on the teachers \( m \) schedule; \( L_{md} \) – number of “bad” days \( d \) on the teachers \( m \) schedule; \( L_{sv} \) – number of “bad” hours \( v \) on the pupils \( s \) schedule; \( L_{pd} \) – number of pedagogical didactic \( pd \) violations; \( L_{n} \) – number \( n \) of changing formed subject-group.

All physical restrictions and inconveniences are showed in Figure 1.

### Schedule

#### Required restrictions
- One teacher in one place
- One pupil in one lesson
- Number of the lessons per day
- Number of the classrooms

#### Needful restrictions
- Stability of the group
- Lesson in the special pace
- Free days for the teacher
- Gap for the pupil
- Gap for the teacher

---

**Figure 1. Restrictions for a creation of high school schedule**

A compromise solution is reached by defining penalties for violation of constraints and disregarding inconveniences. Therefore, penalty points are calculated:

\[ F = F_{j} + F_{n}, \]

(3)

where, \( F_{j} \) – is a sum of the penalties for the required restrictions; \( F_{n} \) – is a sum of the penalties for the needful restrictions (disregarding inconveniences). Optimal schedule will be schedule, which has as less as possible penalty points. To find such schedule, objective function \( F \) should be optimized. To not analyze the schedules with same number of penalty points, Pareto optimality was formulated. So we will get less variants to analyze and will save the users time. The optimization problem is

\[ \min_{\tau \in A} F(\tau), \]

(4)

where, \( F(\tau) \) is the total penalty of some schedule \( \tau \); \( A \) is the set of schedules satisfying the physical constraints. The penalties \( F(\tau) \) depend on expert evaluations, therefore we regard them as heuristics.

### 5 “School schedule optimization” program working steps

“School schedule optimization” program designed to high school scheduling.

**Figure 2. Forming subject-groups to teachers**

**Figure 3. Time table for teachers creation**

Figure 2 illustrates how subject-groups are assigned to teachers. Here pupils \( s_{ij} \) from groups \( G_{i} \) are grouped to the groups with identical subject \( D_{t} \). Identical subject has same name and same hours per week. These groups are called subject-groups (with \( x \) pupils in the group) and assigned to the teacher \( M_{l} \). Figure 3 shows how teacher’s timetables are created. The subject-groups \( D, M[S], S_{x} \) with teacher \( M_{l} \), subject \( D_{t} \) and pupils of this subject-group \( S_{x} \) are putted to the free class-room and to school timetable. When process is finalized, the optimization process is ready to start.
After optimizing, we can see such results of this program:

- school schedule;
- individual pupils schedules;
- individual teachers schedules;
- individual room schedules;
- subject-group schedules;

All results user can see in the program (on working time), or download them as archive personal computer. The program does not require much effort to the user, the payment to work with a computer, or a lot of time to understand how system works.

6 Optimization Methods

6.1. Defining Neighbourhood

Many different definitions can be used defining neighbourhood in a set A of feasible timetables \( d \). The definition is important because local search is performed in the neighbourhood of the given point. We search for better timetables by subsequent closing of gaps for pupils and teachers. In this case the neighbours of a timetable \( d' \) are all timetables \( d'' \) that can be reached from \( d' \) by a sequence of closing gap operations. This way we obtain locally optimal \( d^*(d') \) that depends on the initial point \( d' \).

Local search can be randomized by selecting current candidate (a pupil or a teacher) for gap closing with some probability \( x_0 \). Closing gaps for randomly selected pupils and teachers, we modify the search sequences. However, this not helps to reach the global optimum since the neighbourhood remains the same.

6.2. Escaping Neighbourhood

Simplest algorithm to search for global optimum is just random search with uniform distribution of observations (observation is calculation of the objective function at some fixed point). The advantages are simplicity and convergence to a global minimum of continuous functions. A well-known way to escape the local minimum is Simulated Annealing [1, 2, 14, 19, 21, 22]. Denote

\[
\delta_n = F(d^{n+1}) - F(d^n),
\]

Here \( d' \) is a current timetable, \( d^{n+1} \) is a new timetable generated by closing gap operation. Define the probability

\[
p_n = e^{-\delta_n / (n + x_2)}, \quad \text{if} \quad \delta_n > 0,
\]

\[
p_n = 1, \quad \text{if} \quad \delta_n < 0,
\]

where parameter \( x_1 \) is the “initial temperature”, parameter \( x_2 \) defines the “cooling rate”. SA algorithm means:

\[
\text{go to new timetable } d^{n+1} \text{ with probability } p_n
\]

To apply the SA to a specific problem, one must specify the parameters \( x_1 \) and \( x_2 \). The choice can have a significant impact on the method's effectiveness. Unfortunately, there are no choices of these parameters that will be good for all problems.

![Figure 4. The best results of SA using different parameters](image-url)
Analyzing Figure 4, we see different results using different initial parameters. Here difference of penalty points (between initial and optimal schedules) is calculated. Every column is received after 100 experiments with fixed initial parameters (Iterations, $x_1$ and $x_2$). In the left side of Figure 2 the results are grouped by $x_2$ when $x_1$ was between 100 and 1000. In the right side, the results are grouped by $x_1$ when $x_2$ was between 1 and 10. There are showed only best results after.

6.3. Bayesian Heuristic Approach

The Bayesian Heuristic Approach was designed for automatic optimization of heuristic parameters by filtering the noise during optimization of multi-modal functions [20]. We need to optimize three heuristic parameters $x = (x_0, x_1, x_2)$. Optimal parameters are obtained using the data of some specific school.

We cannot see optimal parameters $x_1$, $x_2$ of SA. Optimal results depend on the initial soft constrains and number of iteration. A way to adapt these parameters to a given problem is automatic optimization. This is not an easy problem since we need optimize multi-modal function with considerable noise. Here the Bayesian Heuristic Approach (BHA) [20] is useful.

Figures 5 and 6 illustrate efficiency of automatic adaptation of SA parameters using BHA. In these figures, the difference between initial and optimal timetable is showed. There we see 100 experiments with every different SA iteration. SA parameters were set automatically. Figure 3 shows, that method is more efficient as more SA iteration are used. Figure 4 illustrates the best results what was shown during 100 experiments with every different SA iteration. There we can see, that the best results we will get when it will be many SA and BHA iterations.

However, the results can be used in similar schools as an approximation.

7 Comparison of results

Here are compared such results: real schedule created in a Lithuanian high school and, from pupils and teachers wishes, created and optimized schedule. Schedule was automatically optimized with Bayes method the results we can see in Figure 7. Both, schedule and data are from the same school and same classes.

Evaluating both types of schedules, penalty points were calculating for:

- pupil window – 5;
- teacher window – 300;
- teachers wished free time – 10;
- exceeding maximum hour limit – 2000;
- pedagogical didactic – 5.

Sum of seted penalty points for the real schedule was 380 020. It is always same, because after finishing the creation process it can’t be changed. Sum of penalty points after optimization process (was seted same penalty points) are showed in the Figure 5. There are few results after optimization with different initial parameters of optimisation method Bayes. The results are different while every time schedule is created from the new point.
As we can see, the optimization results are much better as real schedule result. It is so, while optimization program creates and optimizes schedule only for high school classes. However, teacher can work in basic school to. However, in Lithuanian schools schedule creating starts from high school classes schedule. “School schedule optimization” program is working same way.

8 Optimization in Commercial Software

We discuss optimization possibilities of the following three commercial timetabling systems currently used in Lithuanian high schools: “Mimosa 2009”, “aSc TimeTables 2009”, and “Rector 2009”.

“Mimosa 2009” [18] is the product of the Finnish company “Mimosa Software Ltd”. “Mimosa” provides convenient GUI for manual timetabling and reports constraints violations.

Figure 8 shows a fragment of the output. In the upper-left side we can see pupils schedule, under it – pupils of the subject-group and in the right side – individual schedules of every pupil in the subject-group. The form is acceptable for Lithuanian schools. For example, “Ch3BK” means a chemistry lessons, pupils from 3-rd level, will learn as basic course. Optimization is limited to closing some gaps in teacher’s schedules. The software is popular in basic schools. Application in upper classes of high schools is possible within some strict limitations by setting individual pupil schedules. Long and hard manual work is needed if the school is large. Any penalty points are calculated in this program.

To compare results of different automatic optimization methods we need procedures for evaluation of undesirable factors in some fixed scales. In this paper, it is done in the framework of Pareto optimality [16]. The commercial software does not support this, since no direct comparison of decisions quality cannot be made.
“Rector 2009” [24] is the product of the Russian company “P. Yu. Smykalov”. Figure 9 shows a fragment of output in the format similar to MS “Excel” forms used in local schools. In the upper side the subject for the group 12a are showed. Under it – all groups, lessons per week, subjects and teachers are showed. Green colour means, that no one works at the same time in two places. Reports, if one is trying to insert data to wrong place, are showed in red colour. Convenient for basic school scheduling. No automatic optimization.

Figure 9. A fragment of “Rector 2009” output

“aSc TimeTables 2009” [3] is the product of the Slovak company “Applied Software Consultants s.r.o”. A fragment of resulting timetable for Monday and Tuesday in a compact form for eight pupil subject-groups is in Figure 10. The results of experimental calculations are in Table 1. They show that the software works well in basic schools and is not practical in large high schools. Any penalty points are calculated.

Table 1. Testing „aSc TimeTables 2009“

<table>
<thead>
<tr>
<th>Settings of the program</th>
<th>Small high school (50 pupils)</th>
<th>Medium high school (150 pupils)</th>
<th>Large high school (350 pupils)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>Small Average Hard</td>
<td>Small Average Hard</td>
<td>Small Average Hard</td>
</tr>
<tr>
<td>Viewed options</td>
<td>142800 183265 165236</td>
<td>975736 5324895</td>
<td></td>
</tr>
<tr>
<td>Left subjects</td>
<td>00:13:02 00:10:07 05:53:13</td>
<td>13:23:51 05:32:13</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>00:08:40 02:10:07 16:23:51</td>
<td>00:41:32 02:10:07 05:53:13</td>
<td></td>
</tr>
</tbody>
</table>

A timetable that satisfies all necessary conditions is regarded as feasible. A feasible timetable is optimal if it minimizes all undesirable factors. To compare the quality of different feasible timetables we must evaluate at least the most important undesirable factors. The difficulty is that desirability is subjective by definition and depends on the local conditions. This prevents comparison of results obtained by automatic optimization with decisions made by human operator.

9 Concluding Remarks

- The new element of this work is application and systematic investigation of the Bayesian Heuristic Approach (BHA) [20] to optimization of heuristic parameters (with penalty points). These include the initial temperature and the cooling rate of SA algorithm and the randomization parameter of the local search algorithm.
• BHA is intended for global optimization of functions with noise what is typical in optimization of heuristic parameters.
• The formulation of the objective function in terms of Pareto optimality seems to be new in the field of school scheduling.
• Application in some large schools shows some advantages comparing with commercial software. The web-site: http://soften.ktu.lt/~mockus and accompanying web-sites include corresponding.

References