

# Constraint Priorities - a Way of Getting an Optimal Timetable Fully Automatically. - Demo Storyboard

**Wieslaw Dudek**

Wieslaw Dudek Timetables, Krakow, Poland  
Wieslaw.Dudek@gmail.com

## Abstract

The [www.school-timetable.eu](http://www.school-timetable.eu) system is a modern fully automated timetable scheduler based on constraint priorities. This storyboard document presents some interesting aspects of its construction and the idea behind it and also shows practically how, in simple steps, a user can enter his timetable criteria and receive a solution. The demo comprises showing the architecture of the system, the applied resolutions to common problems and some practical applications in schools and other institutions working with timetables.

## Introduction

For a more detailed description of [www.school-timetable.eu](http://www.school-timetable.eu) scheduler refer to extended abstract prepared for this event. The demo has two main parts. The first one

is a presentation of the system architecture. The second part presents some practical uses of the system.

## System Architecture

It will be presented the goal of the application (generating timetables fully automatically, simply by clicking "Generate" button) and the architecture used in the implementation as shown in Figure 1

## Groups Of Students Defining

In the [www.school-timetable.eu](http://www.school-timetable.eu) scheduler the way of defining groups depends on the initial data we have. There are possibilities to define groups both in case of having full information (see Figure 2) and when some information is missing (Figure 3, 4, 5 and 6).

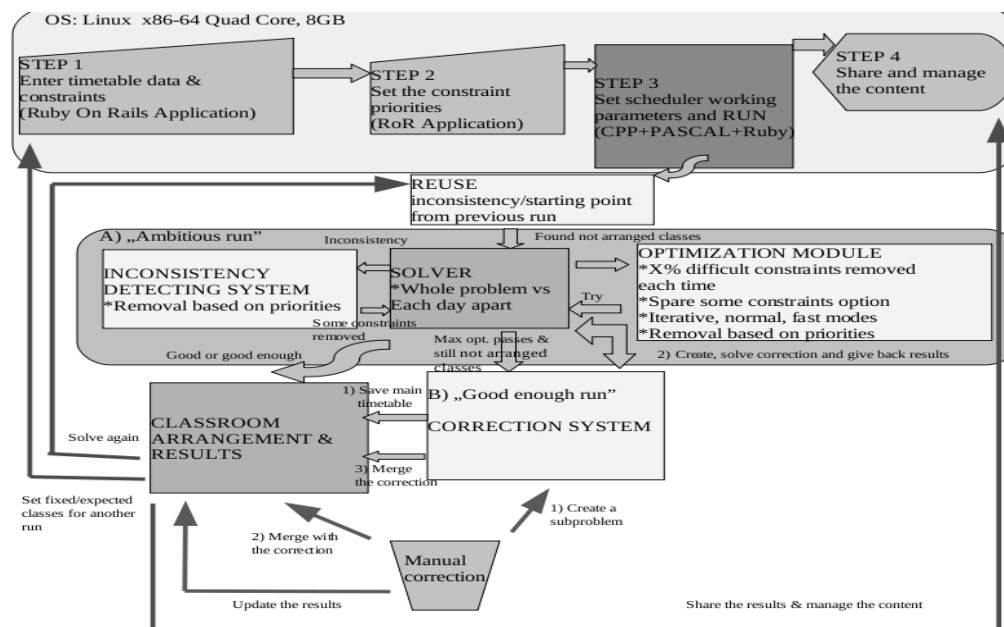


Figure 1: Overview of the [www.school-timetable.eu](http://www.school-timetable.eu) system architecture

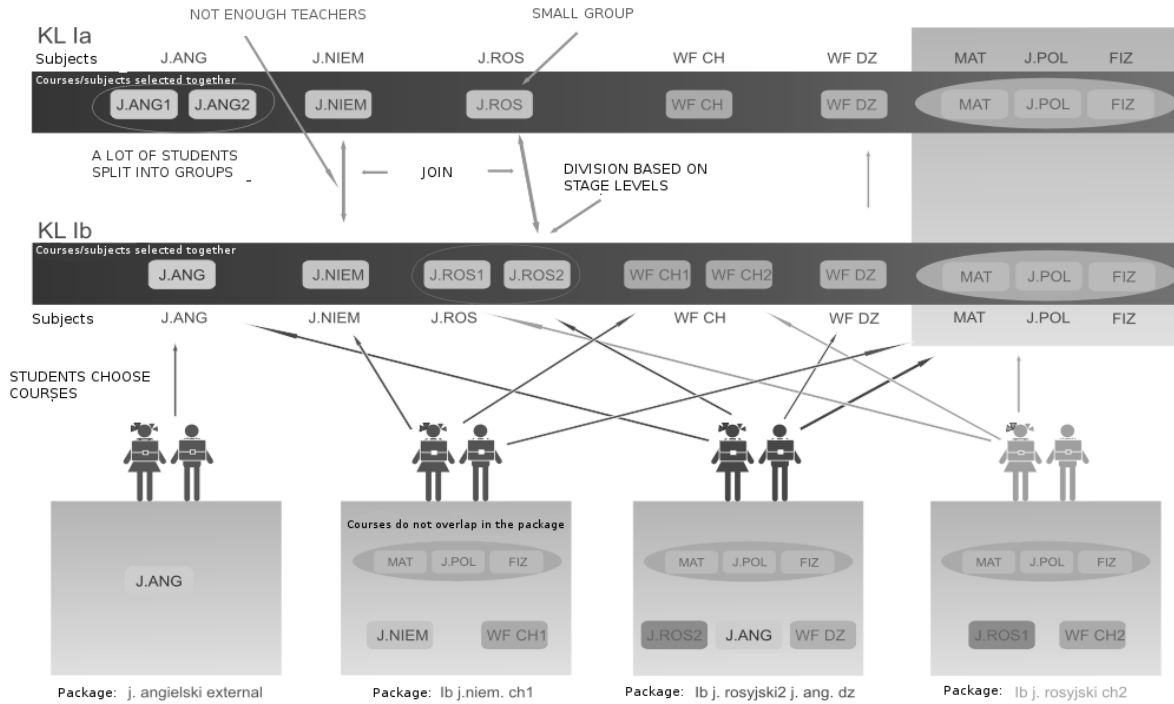


Figure 2: Defining the packages of courses

Below are some examples of correlating and separating of lessons in the context of groups defining in such difficult scenarios when some information is not available.

Example 1. Language groups

We need a timetable but the teachers still have not decided about students divisions into groups e.g. based on stage level. (see Figure 3)

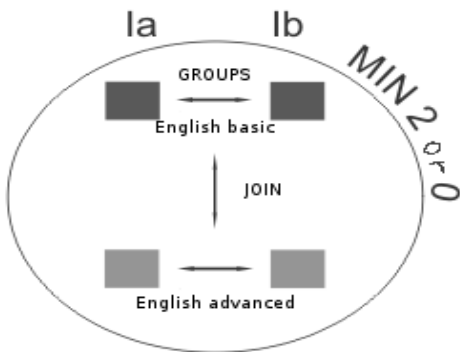


Figure 3- Language groups

Example 2. Elective line

We need a timetable but students have not selected optional courses yet - after selecting courses, we do not want gaps (3 subjects to be selected divided by 3 lessons in 1 lesson hour) (see Figure 4).

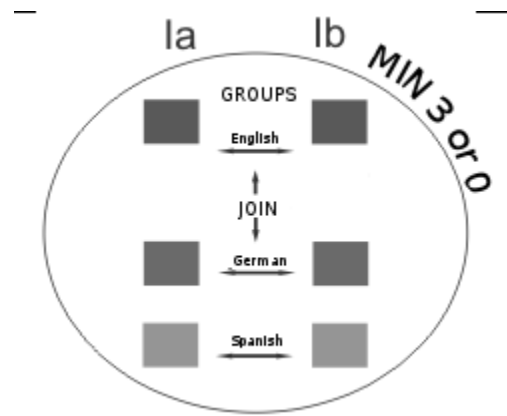


Figure 4- Elective line

We create a subject group - "Languages" and assign Ia

English, Ia German, Ia Russian, Ib English, Ib German, Ib Russian. Next, we need to set a minimum number of lessons in the group in each period to 2 but allowing an option of no lessons because otherwise the system would force 2 lessons to appear in each available period. Now we can combine subjects together, especially if we do not have enough teachers or the defined groups of students are small e.g. Ia English with Ib English etc.

**Example 3. Facilitating substitutions**

We need one teacher to be available for some possible substitutions if necessary (3 subjects to be taught divided by 2 lessons in 1 lesson hour) (see Figure 5).

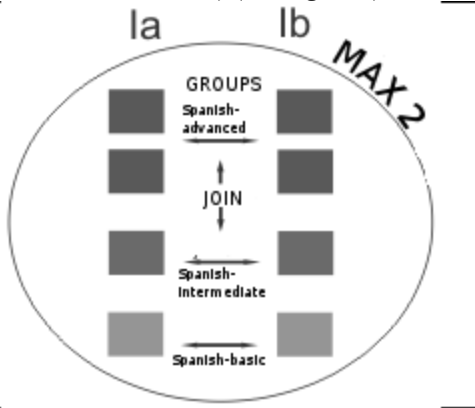


Figure 5 – Facilitating substitutions

As we can see on Figure 5 there is one Spanish teacher available in every period so in “emergency” situation i.e. a teacher absence the spare teacher can pick up his lesson.

**Example 4. Different number of lessons**

Because of a teachers' shortage we need to combine 2 lessons of IIIa PE with 1 lesson of Ia PE and IIa PE by means of subject groups (Figure 6).

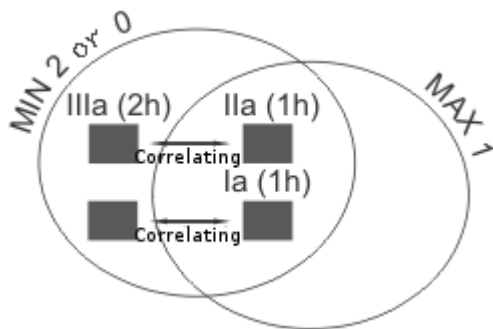


Figure 6 - Different number of lessons

In this case we need to ensure that the single lessons of Ia & IIa will be correlated only with 2 lessons of IIIa but not between themselves. We can achieve this by creating additional group of subjects containing lessons Ia & IIa and setting a maximum number of simultaneous lessons to 1. Also as the lessons are combined we will need only one classroom for the two classes /”no rooms” options set for IIIa only or for both Ia and IIa/.

**Constraint Priorities**

Next, constraint priorities as a main innovation in timetabling process will be present. For some info related to Figure 7 refer to the extended abstract prepared for this event. However a practical use will be demonstrated in the following part.

**Timetabling Flow**

For an average school the whole timetabling process should take a few hours but for demo purposes we will use a small piece of the real timetable. I first show a process of timetable defining, its simplicity, next coming through some parameters changing in order to get some modifications and finally showing possibilities of managing the content. At the very end some possible uses by non-school institutions will be presented.

Here is an exemplary flow for the demo:

- Create a timetable schema with days and times of lessons.
- Enter all classrooms data with their availabilities.
- Define daygroups. Daygroups help to manage a distribution of subjects among days of week.
- Define groups of subjects. Groups of subjects help to manage a distribution of subjects. Here we have defined distribution of pure science classes.
- Enter all teachers with their availabilities and preferences.
- Enter class/course/ data. This step needs to be repeated for every course which significantly differs from others – similar courses with almost the same subjects sets can be cloned later on.
- Define subjects of a course or courses just entered.
- Assign classrooms to subjects manually or generate it automatically thanks to entered sizes of courses & classrooms.
- Clone a course or courses defined before. This is deep cloning, which means the course is cloned with its subjects and assigned classrooms.
- Update subjects information for cloned courses.
- Split courses into groups. The reason for splitting can be a number of pupils.
- Update some subjects data for split groups. It could be

PRIORITY	RELATED TO	REQUIREMENTS	ELEMENTS	ELEMENTS BEFORE CONSTRAINTS
1	Subjects	Number of lessons per week	All subjects	No
2	Classes	No gaps	All classes	Yes
3	Subjects	Classrooms arrangement	English; Chemistry	No
4	Classes	Availability; Starting time of lessons	All classes	No
5	Teachers	Gaps constraints; Availability	John Smith; Ann Brown	Yes
6	Teachers	Availability	All teachers	No
7	Subjects	All	All subjects	No
8	Classes	All	All classes	No
9	Teachers	All	All teachers	No
10	Groups of subjects	All	All Groups	No
11	Students	All	All students	No
12	Resources	All	All resources	No

Figure 7: Example of constraint priorities defining

necessary e.g. to assign pure science individually for each group in order to force an even distribution within one class.

- Add new optional courses. It is common to enter some courses for boys/girls here e.g. physical education or some language courses which require some stage level from attendees. We enter only the ones which cannot be cloned. Sometimes there are courses defined here on school level.
- Enter subjects data for the optional courses with classrooms assignment information.
- Clone some similar courses.

- Combine some subjects together if necessary e.g. too small groups, not enough teachers.
- Identify groups and divisions and generate packages of courses.
- Manually add packages of courses on school level.
- Update the packages attributes. Most likely we need to set some attributes as no gaps mode, starting lesson time or min, max number of lessons. This can be done by editing individual items or by using the mass change option.
- Enter students or individual packages. Turn on registration for courses if necessary. Afterwards it needs to be turned off.
- Set the constraint priorities. We can define constraints priorities for removing constraints in case of inconsistency or during optimization process. It helps to get a solution as close to our demands as we need. For example we can demand no gaps for a few teachers as a priority constraint. Finally, the timetable generator informs us about all necessary modifications it has done.
- Set scheduler working parameters and click the "Generate" button. We can define several options here. The most important ones are:
  - optimization process touches constraints with the minor priority starting from* - this option lets preserve major constraints from being removed;
  - use results from previous run* - before running the generator we can decide to continue previous calculations selecting "Continue previous run";
  - generate each day separately* – to speed up the generator

THE BEST SCHOOL TIMETABLE GENERATOR & BROWSER ONLINE

Classes /courses/

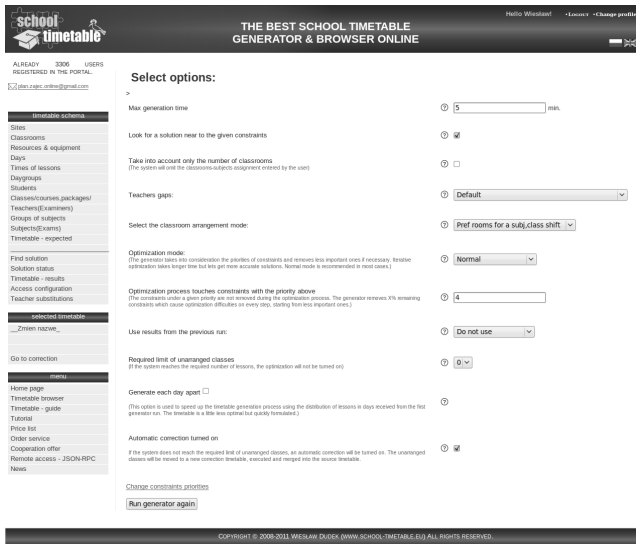
Name	CLASS	Division	GROUP
1a	1a		
1b	1b		
g	g		
g	g		
IT			
Phys. Education	1a	sex	boys
Phys. Education	1a	sex	girls
Phys. Education	1b	sex	boys
Phys. Education	1b	sex	girls

Classes /packages of courses/

NAME OF THE PACKAGE OF COURSES (OR STUDENT'S NAME)
1a-boys
1a-girls
1b-boys
1b-girls
IT
IT
Lets Do

Copyright © 2008-2011 Wekiv Duda (www.school-timetable.eu). All rights reserved.

or to execute only selected days  
*automatic correction turned on* – if necessary the second run will be triggered



After a “ambitious run” the second „good enough run” is triggered if the correction was turned on. The second run is triggered automatically if necessary and it typically takes several seconds.

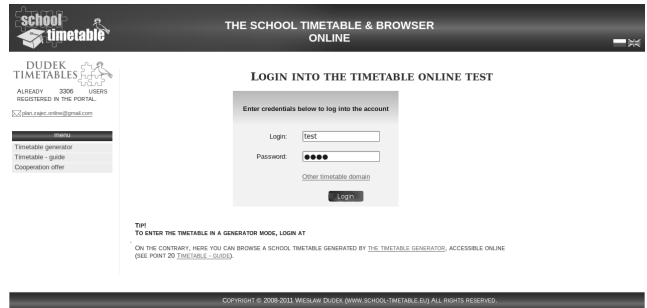
- The solution is ready to be browsed. Check the actual inconsistencies found and the constraints removed by optimization process.



- Modify or share the timetable. We can do another correction, modify or make the timetable accessible online. Also some combinations of turning on/off the

correction process and optimization process can be tested here. Moreover, we can test some exporting capabilities and a final correcting of classrooms arrangement.

- Log into the shared timetable/ manage the content. We can check the timetable as well as manage substitutions or replacements online.



- We could enter a few working sites for school and set the site for subjects & classrooms. We can define separate times of lessons for the sites and change the sites inter-moving times (Figure 8).

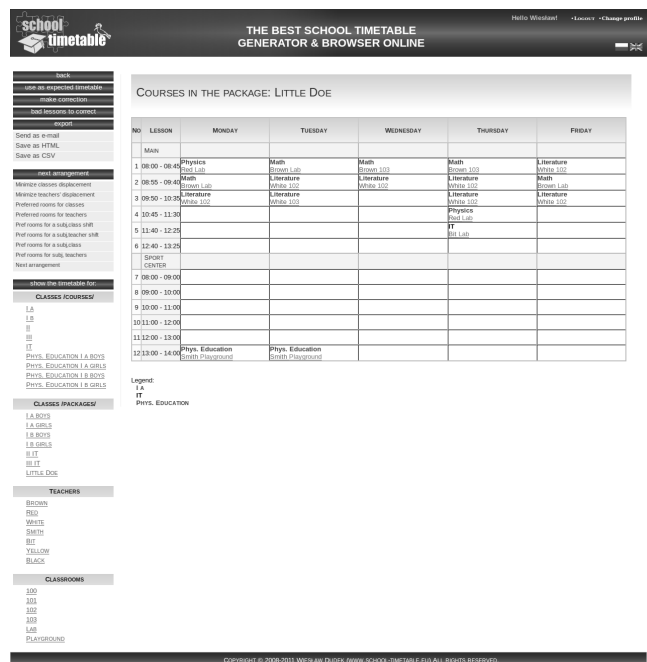


Figure 8: Timetable with sites defined.

## Other Applications

The system is capable to generate timetables for other institutions beside schools e.g. work-schedules for some companies.

Here is an example of using the scheduler to generate work-schedules and shifts (Figure 9).

Classrooms were not used there so each subject /task/ was defined with the "Without classrooms arrangement"

ELA

No	LESSON	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
	SLOWACKIEGO 1																																
1	00:00 - 01:00						U	U																									
2	01:00 - 02:00						U	U																									
3	02:00 - 03:00						U	U																									
4	03:00 - 04:00						U	U																									
5	04:00 - 05:00						U	U																									
6	05:00 - 06:00						U	U																									
7	06:00 - 07:00						U	U																									
8	07:00 - 08:00						U	U																									
9	08:00 - 09:00						U	U																									
10	09:00 - 10:00				PAP+ZEB ZEB		U	U																									ZR
11	10:00 - 11:00				PAP+ZEB ZEB		U	U																									ZR
12	11:00 - 12:00				PAP+ZEB ZEB		U	U																									ZR
13	12:00 - 13:00				PAP+ZEB ZEB		U	U																									ZR
14	13:00 - 14:00				PAP+ZEB ZEB PAPU		U	U																									ZR
15	14:00 - 15:00				PAP+ZEB ZEB PAPU		U	U										ZW		ZW		ZW					ZW	ZW				ZR	
16	15:00 - 16:00				PAP+ZEB ZEB PAPU		U	U				ZW						ZW		ZW		ZW					ZW	ZW				ZR	
17	16:00 - 17:00				PAP+ZEB ZEB PAPU		U	U				ZW						ZW		ZW		ZW					ZW	ZW				ZR	
18	17:00 - 18:00				PAPU		U	U				ZW						ZW		ZW		ZW					ZW	ZW				ZR	
19	18:00 - 19:00				PAPU		U	U				ZW						ZW		ZW		ZW					ZW	ZW				ZR	
20	19:00 - 20:00				PAPU		U	U				ZW						ZW		ZW		ZW					ZW	ZW				ZR	
21	20:00 - 21:00				PAPU		U	U				ZW						ZW		ZW		ZW					ZW	ZW				ZR	
22	21:00 - 22:00				U		U	U				ZW						ZW		ZW		ZW					ZW	ZW				ZR	
23	22:00 - 23:00				U		U	U																									
24	23:00 - 23:59				U		U	U																									
25	10:00 - 18:00				SZK																												

Generated by:



Figure 9: Work-schedule

option (but we could easily use them if some resource was needed for work and shift or we can use "Resources" for the same purpose as well).

Teachers were defined with the "virtual" option but in some cases we could be in the need to define managers to supervise shifts. For such "virtual" teachers, the tasks (subjects) can overlap. The "none" named teacher will not be displayed on the shared timetable.

Employees can be defined as courses or packages of courses. In this example, we decided to use the first option. The second option could be useful if we wanted to let the employees register for work shifts. Also a "virtual" class named "none" was defined. The class contains all tasks (subjects) for employees to execute.

Here the tasks were correlated with John's (Jan) morning shift to be done by him by defining the "Taski Jana"(John's tasks) group with the subjects: "ZR" (i.e. John's morning shift), "ZAD1", "ZAD2". Tasks "ZAD1", "ZAD2" were defined as separated by defining the "Separated tasks"

subject group. The working calendar for a month with 31 days & 24 hours was entered. For each employee (here: course) who can be on a morning shift, a ZR subject was entered, the same for a night shift ZW and a "mid-shift" ZM with an 8-hour length and 0 hours a day. The number of hours set to zero lets define quantity on another level - in this case on the level of all possible ZR,ZW,ZM (by defining the group of subjects) and the number of working hours to be set by the employees.

To define the number of working hours, the "Zmiana poranna" (morning shift) group was created with ZR subjects assigned to all employees. Next min and max number of lessons a day were set to 8 (i.e. one shift a day) and 1 simultaneous shift a day was allowed. To allow 2 employees per shift, we can set the "number of simultaneous lessons" to 2 and min, max 16 hours a day. The number of working hours could be set by defining a group of days "Miesiac" (month) and setting min & max number of lessons in the group for each employee.

By defining "Robocze ..." (Working ...) groups, we could set max number of hours a day for an employee to 8 and required by the Polish labour law 11 h of rest between shifts - parameter "Min distance in hours" set to 19 because the distance is counted from the beginning of a shift. For each task (subject) the starting time was set to be flexible in case of some unavailability of employees or resources e.g. ZR can start from 8:00 or 9:00. The employee Ann (Ania) wanted to avoid ZW shifts so in a group of days "Miesiąc" had to be set max 0 hours for ZW and for similar reason max 8 hours were set for ZM. For other employees, working hours for some shifts were limited. To balance working hours for each employee, the "Max distance in hours" option was set to 80 to ensure the shift distance will not be longer than 80-8 hours (8 is the length of a shift). To execute a task: PAP+ Ela needs to finish some trainings, which should be done before the scheduled task (subjects precedence feature used).

## References

Demo samples: <http://www.school-timetable.eu/access/demo>  
A service guide: <http://www.school-timetable.eu/guide/guide>  
A service tutorial: <http://www.school-timetable.eu/access/tutorial>  
More info: <http://www.school-timetable.eu>