

Implementation of IT Tools To Help Improve The Efficiency of Timetable Planning*

Dorota BORKOWSKA and Anna GLADYSZ

Rzeszow University of Technology, Poland

Correspondence should be addressed to: Dorota BORKOWSKA, db@prz.edu.pl

* Presented at the 45th IBIMA International Conference, 25-26 June 2025, Cordoba, Spain

Abstract

Scheduling is a multidimensional challenge for many organisations due to the numerous constraints that need to be considered during planning. The problem becomes even more complex when the data for the scheduling process is scattered, leading to time-consuming and labor-intensive manual work. This paper highlights the possibility of automating the scheduling process through the use of IT tools. An implementation of a FET programme is proposed that automates the scheduling process, minimising the risk of conflicts and ensuring optimal use of resources. The study presents solutions aimed at improving the planning of academic schedules by eliminating or reducing certain scheduling challenges. The appropriate selection of IT tools enables the optimisation of timetabling, resulting in greater organisational efficiency for the university, time savings, and improved working conditions for all schedule users.

Keywords: Timetabling, scheduling, optimisation, streamlining, FET

Introduction

Scheduling is a common challenge faced by many companies and institutions. Proper organisation, control, work optimisation or resource allocation significantly impact work quality and efficiency. Preparing a schedule taking into account employee or resource constraints, aiming to optimally allocate tasks with the best possible use of available instruments is an extremely labour-intensive and complex task. Due to the fact that many optimisation problems are still solved manually, the multidimensional nature of scheduling also makes it a time-consuming process involving many people.

The problem of scheduling also concerns the academic environment. Each university, through experience, develops its own standard practices for schedule preparation. When creating a timetable, numerous constraints and solution objectives must be considered, making it impossible to establish cyclical schedules. Factors such as fluctuations in student enrollment, affecting the number of class groups, the introduction of new study programmes, and the creation of additional specialisations within existing programmes are among the primary reasons requiring schedules to be generated from scratch. The limited time available for preparing a new schedule determines the need to

enhance the efficiency of the planning process. There are various ways to improve the effectiveness of academic scheduling. These methods rely on tools that support data preparation, timetable planning, and schedule publication.

There are many IT tools available on the market to support the work of the planner. When choosing the right software, the planner is primarily guided by time savings, accessibility, the ongoing development of the software, the richness of the operating instructions, the available budget and the ability to integrate with other IT systems. The complexity and functionality of planning software mainly depends on the local needs and the nature of a country's universities. For example, most Polish universities use a comprehensive IT tool for the management of student affairs, USOS - the University System of Student Services (USOS). Associated with this system are a number of web services supporting many aspects of the university's operation, which involves a significant reduction in the burden on administrative staff. These services enable, among other things, the display of timetables. Therefore, the most popular software on the Polish market is that which allows the synchronisation of timetables with the USOS database. Such programmes include Planista 3 (Planista), Plansoft.org (Plansoft) or UNIplaner (Uniplaner), which are designed to support the creation of timetables based on data defined in USOS. These programmes require a continuous connection to an appropriately configured database, and all operations performed in them are immediately reflected in USOS.

The implementation of appropriate tools, particularly IT solutions, brings significant benefits to both planners and the entire institution. With the right selection of IT tools, it is possible to automatically generate balanced timetables for both students and staff, ensuring efficiency and optimal resource allocation.

In addition to commercial solutions, there are a number of open source programmes available on the market, which are mainly distinguished by their simplicity, intuitiveness, ease of use, speed of operation and accessibility. One such programme is FET - Free Timetabling Software (FET - Free Timetabling Software), which supports automatic timetabling by combining all the advantages of an effective application. The format in which the individual timetable versions are saved in FET to an XML file provides the user with extensive possibilities for editing or extending the timetable and, most importantly, for importing or exporting data to study management systems.

The paper proposes the use of effective IT tools to support scheduling processes in the FET programme. It indicates solutions that can replace the manual, labor-intensive and error-prone process of creating a class timetables. A comprehensive approach to scheduling including the automation of input and output data processing, leads to increased work efficiency, as well as time and resource savings.

Planning Assumptions

The scheduling problem, referred to in the literature as Timetabling, is classified as an NP-hard optimisation problem (Thomas H Cormen 2009, Rustauletov 2020, Garey and Johnson 1983, Betaouaf et al. 2021). It involves a large number of variables and numerous constraints, many of which may be conflicting, hence the need for their organisation and selection. Additionally, it is a two-dimensional problem, as each activity is assigned both a start time and a teaching room. Furthermore, each schedule has its own complexity, depending on the specific characteristic of the organisation for which it is developed.

When considering the scheduling problem in higher education institutions, its complexity primarily depends on the size of the university measured by the number of students, which in Polish universities ranges from a few hundred to several tens of thousands. While at smaller institutions, a single team of planners may be sufficient to create schedules for the entire organisation, in the case of the largest universities, this task requires the involvement of multiple teams. These individual teams are then responsible for scheduling for specific faculties within the university.

The task of the planning team is to create a timetable that enables academic teachers to teach effectively and allowing students to acquire and expand their knowledge and skills effectively. This process requires a thorough understanding of the institution's regulations, effective communication between university departments, and reliable planning tools. The integration of these factors directly impacts the quality of the timetable, which in turn influences the work culture and collaboration within the academic environment.

Creating a university timetable is a complex task due to the multitude of various constraints that must be considered and strictly observed during its preparation. Different room capacities, specialised rooms, availability of academic teachers, elective subjects, variable frequency of courses, non-standard duration of classes, are just some of the constraints that need to be taken into account when building a plan. In order to maintain control over all plan parameters, it is worth reaching for specialised tools that automate the planning process and help prevent human errors (Joy et al. 2023).

The paper attempts to automate the scheduling process using the Faculty of Management at Rzeszów University of Technology (PRz), a medium-sized higher education institution, as a case study. The focus is on preparing a semester schedule for full-time students. The faculty offers six undergraduate and four graduate programmes. The example was based on real data with a total of 1238 scheduled classes (on a weekly or biweekly basis) for 23 years, including 82 exercise groups and 152 laboratory groups. The faculty follows a structured division of classes into lectures, exercise, and laboratory, which determines the student distribution. Lectures are attended by all students within a given study programme or specialisation. Exercises are limited to groups of no more than 30 students, while laboratory groups are formed as a half of the exercise group. These predefined class groupings directly impact room assignments in terms of capacity and equipment. Additionally, the maximum number of students in any given study programme cannot exceed the seating capacity of the largest lecture hall.

Full-time teaching takes place from Monday to Friday between 7.00 a.m. to 8.45 p.m. The teaching unit is a teaching hour, which lasts 45 minutes. Every two teaching hours forming a standard teaching block, a 15-minute break must be scheduled. For classes with a non-standard number of teaching hours, the timing of breaks is determined individually. Each single teaching day consists of the following set of teaching blocks:

- I teaching block: 7:00 – 8:30
- II teaching block: 8:45 – 10:15
- III teaching block: 10:30 – 12:00
- IV teaching block: 12:15 – 13:45
- V teaching block: 14:00 – 15:30
- VI teaching block: 15:45 – 17:15
- VII teaching block: 17:30 – 19:00
- VIII teaching block: 19:15 – 20:45

In addition, the following guidelines from the orders of the PRz Rector and the Dean of the Faculty of Management were taken into account during the planning process:

- Full-time teachers and students follow a five-day workweek, from Monday to Friday.
- A teacher's daily workload must not exceed eight teaching hours.
- All classes are conducted on campus; online classes are not permitted in the full-time study mode.
- Classes, such as physical education and foreign languages, are selected by students from the offerings provided by interdepartmental units responsible for these subjects. Foreign language classes for a given course are scheduled at the same time for all language groups. Physical education sessions are organised in a way that allows students to choose their preferred time slots and types of activities.

For the given problem, the solution search space can be calculated using the following formula (Cormen 2009, Krzeszowska et al. 2010):

$$(1) \quad (p \times r)^l$$

where p presents the number of teaching blocks in the weekly schedule, r denotes the number of available rooms, and l stands for the number of classes to be scheduled.

Table 1: Total number of classes

Type of classes	Number of classes							
	Lectures		Exercises			Laboratory		
			C	P	S	L	P	S
The number of class hours [h]	30	15	30	15		30		15
The number of classes	53	163	210	382		177		184

Classes marked as 30 hours in Table 1 take place every week throughout the academic semester, while 15-hour classes are held every two weeks. The academic semester consists of 15 weeks, which are divided for convenience into odd weeks (A) and even weeks (B). To ensure an equal distribution of hours across the semester for 15-hour courses in both weeks A and B, classes from both weeks are scheduled for 45 minutes each in the final (15th) week. It is assumed that all classes, including lectures, tutorials, and laboratory sessions, run throughout the entire semester within standard time slots.

Based on the assumption that the academic week consists of 40 time slots ($5 \text{ days} \times 8 \text{ teaching blocks}$) and that the faculty has 18 rooms available for tutorial sessions, the following was obtained:

$$(p \times r)^l = (40 \times 18)^{401} = 720^{401} > 6 \times 10^{1145}$$

where $l = 210 + \frac{1}{2} \cdot 382$, which gives a vast space of all possible solutions.

Preparations for Scheduling

When beginning the scheduling process, the first step is to analyse which courses need to be included in the timetable, considering the size of student groups and years of study. The study programme defines the subjects assigned to each year of study, including specialised courses. By incorporating additional information on the number of groups and specialisations, a comprehensive list of classes that must be scheduled is obtained. The number of student groups is often difficult to predict, particularly for first-year students, as it largely depends on the results of the admission process. Additionally, the number of groups in a given year is influenced by students' choice of specialisation, which determines a set of specialised courses. If different specialisations share common subjects, these must be scheduled together for all relevant groups. Therefore, each academic year is assigned a specific set of courses, and each course generates a series of classes that need to be scheduled, depending on the number of groups.

The collected data on the number of lecture, exercise, and laboratory groups for a given academic year and study programme also consider the specialisations within the education cycle. For example, in the third year of the Finance and Accounting programme, students are divided into two specialisations (Table 2). In addition to specialisation-specific courses, all students in the year of study attend common subjects delivered through lectures, exercises, and laboratory sessions. Furthermore, specialised courses also include all three forms of activity. Lecture groups within specialisations vary significantly in size. Additionally, in the second specialisation, an uneven distribution of laboratory groups is observed. A key priority is the proper numbering of student groups within of a given year. For instance, students in the third laboratory group of the second specialisation should simultaneously be assigned to the ninth laboratory group within the entire year of study.

Table 2. Example of division into groups and their numbers

Finance and Accounting programme - third year of study										Number of students
Common subjects	C1		C2		C3		C4		C5	119
	L1	L2	L3	L4	L5	L6	L7	L8	L9	
spec. 1	C1		C2		C3					79
	L1	L2	L3	L4	L5	L6				
	C1		C2		C3					
	L1	L2	L3	L4	L5	L6				
spec. 2	C1		C2							40
	L1	L2	L3							
	C4		C5							
	L7	L8	L9							

The next step is to verify whether the faculty has a sufficient number of seats in rooms for all teaching blocks.

Table 3. Percentage utilisation of educational facilities in the analysed faculty

Teaching block	Monday (%)	Tuesday (%)	Friday (%)	Thursday (%)	Friday (%)	Total (%)
7.00 – 8.30	11	24	36	33	26	26
8.45 – 10.15	74	72	89	73	94	80
10.30 – 12.00	97	96	81	98	94	93
12.15 – 13.45	95	98	100	98	100	98
14.00 – 15.30	83	80	95	97	82	87
15.45 – 17.15	79	85	70	55	6	59
17.30 – 19.00	63	59	58	40	3	44
19.15 – 20.45	4	4	10	5	0	5

Table 3 presents the percentage utilisation of lecture halls and rooms during the analysed semester at the faculty. The study does not include laboratory classes, as they take place in both computer labs and specialised laboratories designated for specific laboratory courses.

Maximising the utilisation of room resources to minimize costs often prevents the generation of an optimal schedule. Determining an appropriate level of resource utilisation as a reasonable compromise between costs and the requirements of schedule users (both instructors and students) is challenging. An analysis of the class schedule at the faculty indicates that an occupancy rate of 80% or higher throughout the entire week creates significant difficulties for the scheduler. The limited availability of rooms during the most desirable hours (from 8:45 AM to 3:30 PM) leads to gaps in classes for both teachers and students, as well as an inability to fully meet all user expectations.

The quality of the class schedule is primarily shaped by how well it meets the preferences of both students and teachers. Therefore, the next stage of schedule preparation involves understanding user expectations. Determining

the preferences of student groups is significantly more challenging and generally focuses on minimising gaps between classes and incorporating free days into the schedule.

In order to better understand teachers' expectations, it is advisable to introduce well-designed forms that allow them to provide suggestions regarding class scheduling. Preparing these forms in an electronic format enables the automation of data processing, facilitating the extraction of essential information for further planning.

FET Application

The purpose of the study was to present solutions that automate the process of creating an academic timetable. As a tool for scheduling classes, the FET programme - Free Timetabling Software was chosen, which allows the introduction of various constraints, narrowing the space of possible solutions. FET supports automatic scheduling in schools and universities for structures organised into sets: years, groups and subgroups (Rustaufetov 2020, Das et al. 2021). FET uses an efficient heuristic algorithm based on recursive swapping of classes, simulating a manual schedule generation procedure. The programme places each class in a time slot, respecting the specified constraints (Muhamad et al. 2018, Tosi 2023).

Appropriate preparation of input data and the imposition of constraints that take into account any recommendations regarding student groups, lecturers, rooms and time allows the automatic placement of classes in time slots.

The general scheme of FET implementation (Fahmy et al. 2014) was presented by means of four phases. These include the process of data collection, data input, activity creation, and timetable generation (Fig. 1).

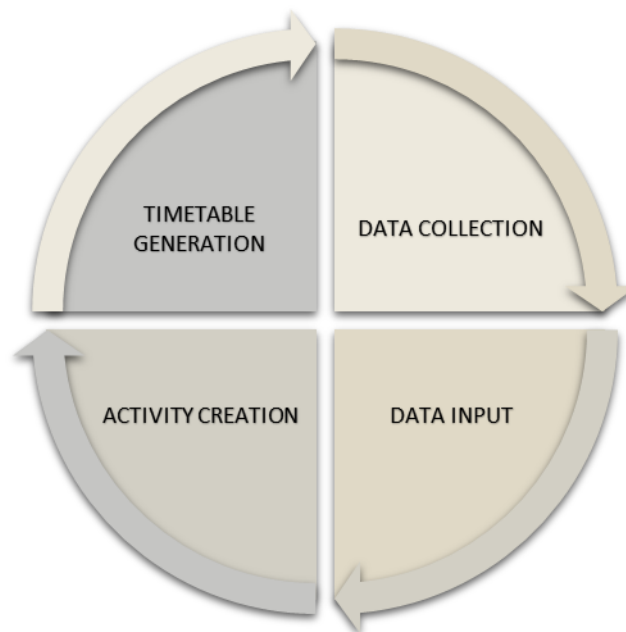


Fig. 1. Implementation Framework

The first phase involves collecting the necessary data to develop the schedule, such as the fields of study, the division of students into groups, the list of subjects, the list of teachers, or the list of buildings and rooms. The second phase involves entering the data into the FET programme with a list of the days and times classes are held during the day. The third phase involves creating a set of classes for all groups of students and teachers. A single activity contains the name of the subject, students/set of students, teachers/all teachers, duration or tags of the activity. The fourth phase involves generating and locking the timetable.

A very important aspect of scheduling classes is assigning appropriate rooms to them. This process is called the term Rooming. FET allows to determine the capacity of the rooms and the buildings in which they are located. In

addition, the programme allows to enter various room restrictions in order to achieve the most effective planning result. With these, it is possible to assign a set of specialised rooms to classes, e.g. only computer laboratories for IT classes.

Automation of the data input process in FET

The planning of semester classes at the Faculty of Management of the PRz is based on administrative data collected in tabular form, which includes a list of classes supplemented with all the necessary information on fields of study, specialisations, cycles of study, number of lecture, exercise and laboratory groups, names of activities, academic teachers and number of hours of individual activities. In addition, the planner must obtain from various sources data including a list of teaching rooms and the number of students assigned to each activity group. The data stored in tabular form allows the planner to efficiently explore it through the use of various IT tools. The second phase of FET implementation implemented manually is a very time-consuming task. Therefore, it is worth using tools that automate the data input process. The article presents a proposal for building a timetable from start to finish in an automated method in the FET programme, taking into account the assumptions made.

Saving the timetable in FET to XML format gives the user many possibilities for editing or expanding it. The file schema of each version of the timetable consists of structured blocks of code storing information on a separate category of data. Knowing the structure of the FET file (*.fet) makes it possible to enter all input data, eliminating the second phase of FET implementation (Fig. 1). Using basic programming skills, the planner can prepare code, run cyclically to generate the input file for the FET programme.

First, the days and time intervals of the classes should be defined. For classes divided into weeks A and B lasting from Monday to Friday, it is necessary to enter in the <Days_List> </Days_List> section 10 entries, two for each day of the week:

```
<Days_List>
<Number_of_Days>10</Number_of_Days>
<Day>
<Name>Monday A</Name>
<Long_Name></Long_Name>
</Day>
<Day>
<Name>Monday B</Name>
<Long_Name></Long_Name>
</Day>
...
</Days_List>
```

As a result, after loading the file into the FET programme, the *Data -> Basic -> Days and hours -> Days per week* window will display the applied data (Fig. 2).

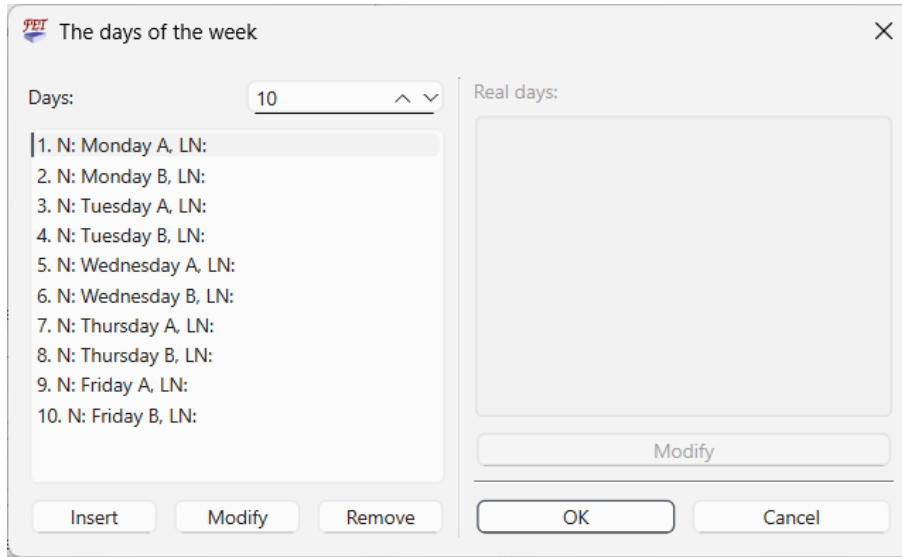


Fig. 2. Window *Data* → *Basic* → *Days and hours* → *Days per week*

The teaching hours are placed in the section `<Hours_List>` `</Hours_List>` (Fig. 3):

```
<Hours_List>
<Number_of_Hours>16</Number_of_Hours>
<Hour>
<Name>7:00-7:45</Name>
<Long_Name></Long_Name>
</Hour>
<Hour>
<Name>7:45-8:30</Name>
<Long_Name></Long_Name>
</Hour>
...
</Hours_List>
```

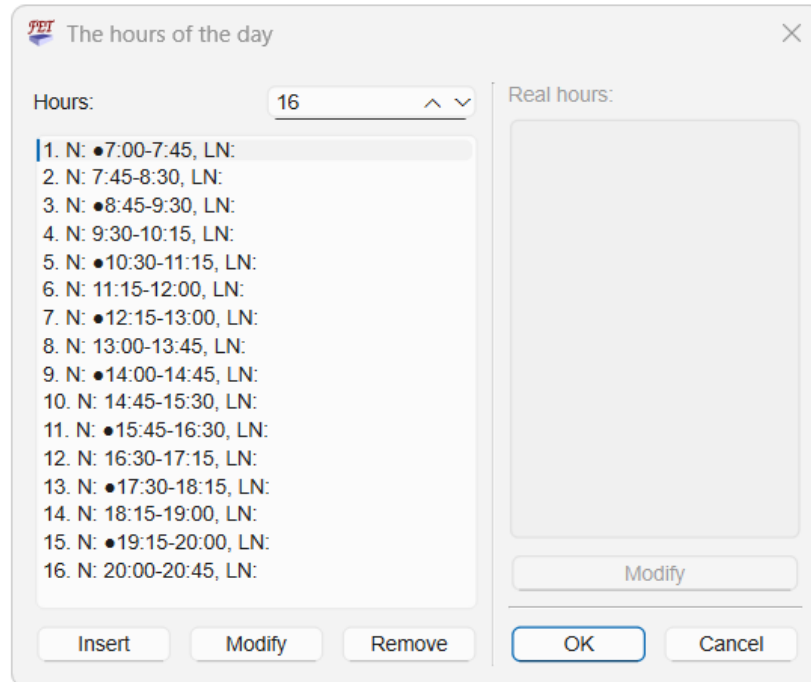


Fig. 3. Window *Data* → *Basic* → *Days and hours* → *Hours (period) per day*

The list of subjects is in the section <Subjects_List> </Subjects_List> (Fig. 4):

```

<Subjects_List>
<Subject>
<Name>Econometrics (ZDL4) - L30</Name>
<Long_Name></Long_Name>
<Code></Code>
<Comments>ZZ0-DL&gt;E</Comments>
</Subject>
...
</Subjects_List>

```

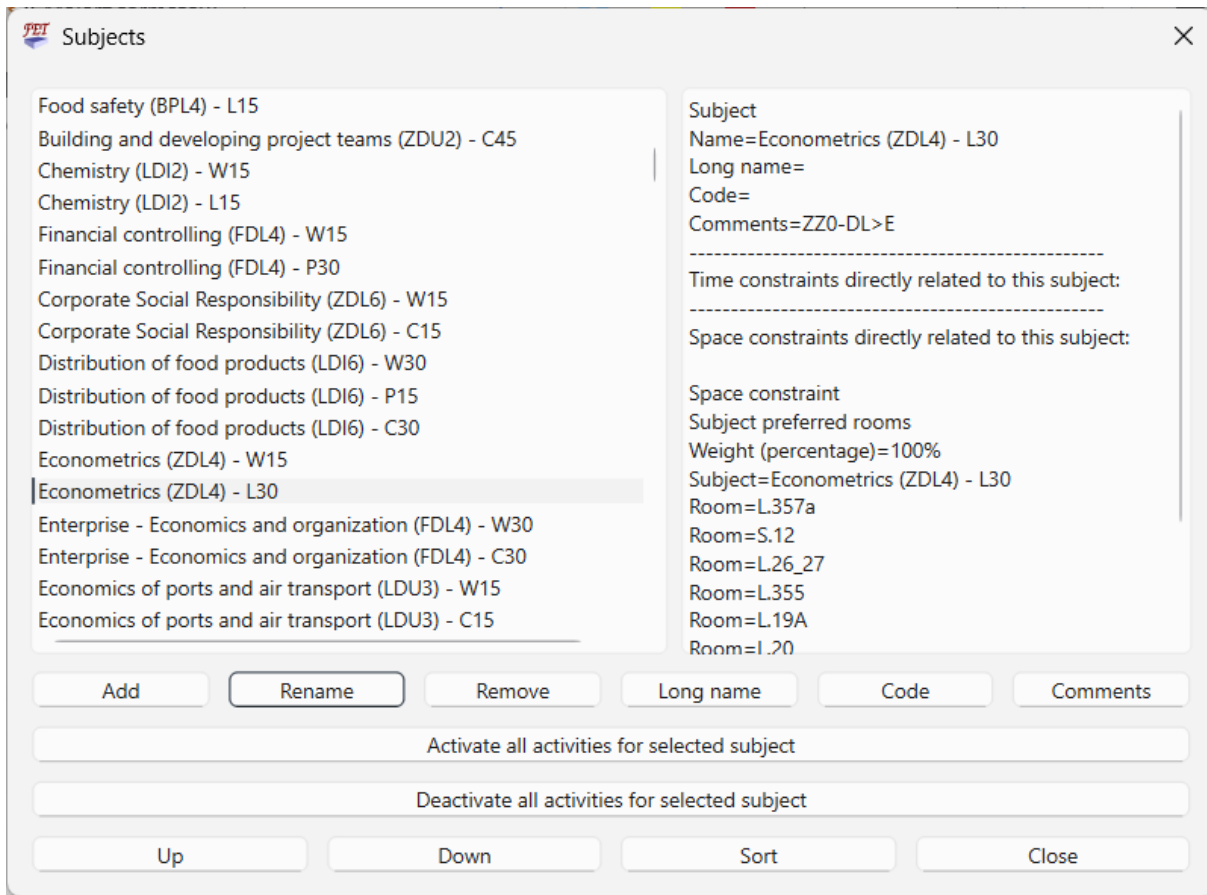


Fig. 4. Window *Data* → *Subject*

Section `<Activity_Tags_List>` is used to collect activity tags (Fig. 5):

```

<Activity_Tags_List>
  <Activity_Tag>
    <Name>Labolatory</Name>
    <Long_Name></Long_Name>
    <Code></Code>
    <Printable>true</Printable>
    <Comments></Comments>
  </Activity_Tag>
  ...
</Activity_Tags_List>

```

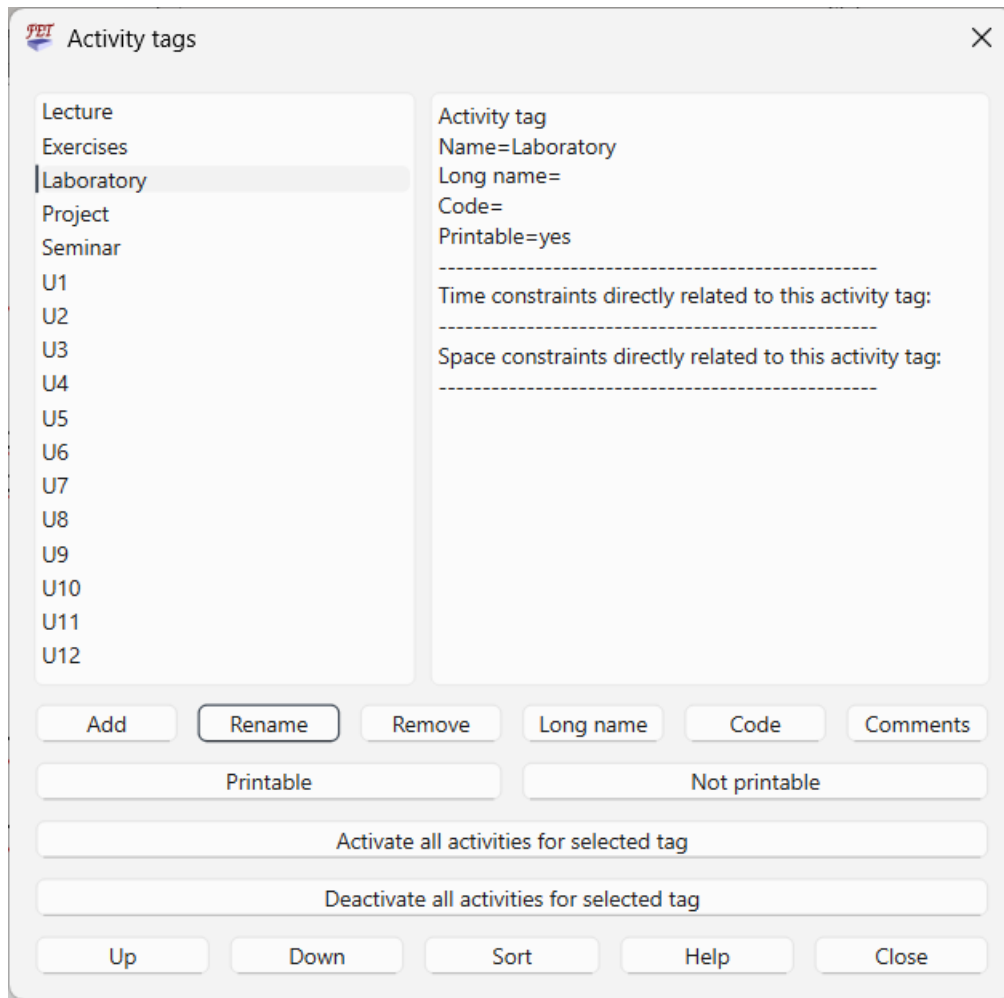


Fig. 5. Tags inserted automatically and by the user in the window *Data -> Tags*

Data of university teachers should be placed in the section `<Teachers_List> </Teachers_List>` (Fig. 6):

```

<Teachers_List>
<Teacher>
<Name>10-University Teacher (ZF)</Name>
<Long_Name></Long_Name>
<Code></Code>
<Target_Number_of_Hours>0</Target_Number_of_Hours>
<Qualified_Subjects>
</Qualified_Subjects>
<Comments></Comments>
</Teacher>
...
</Teachers_List>

```

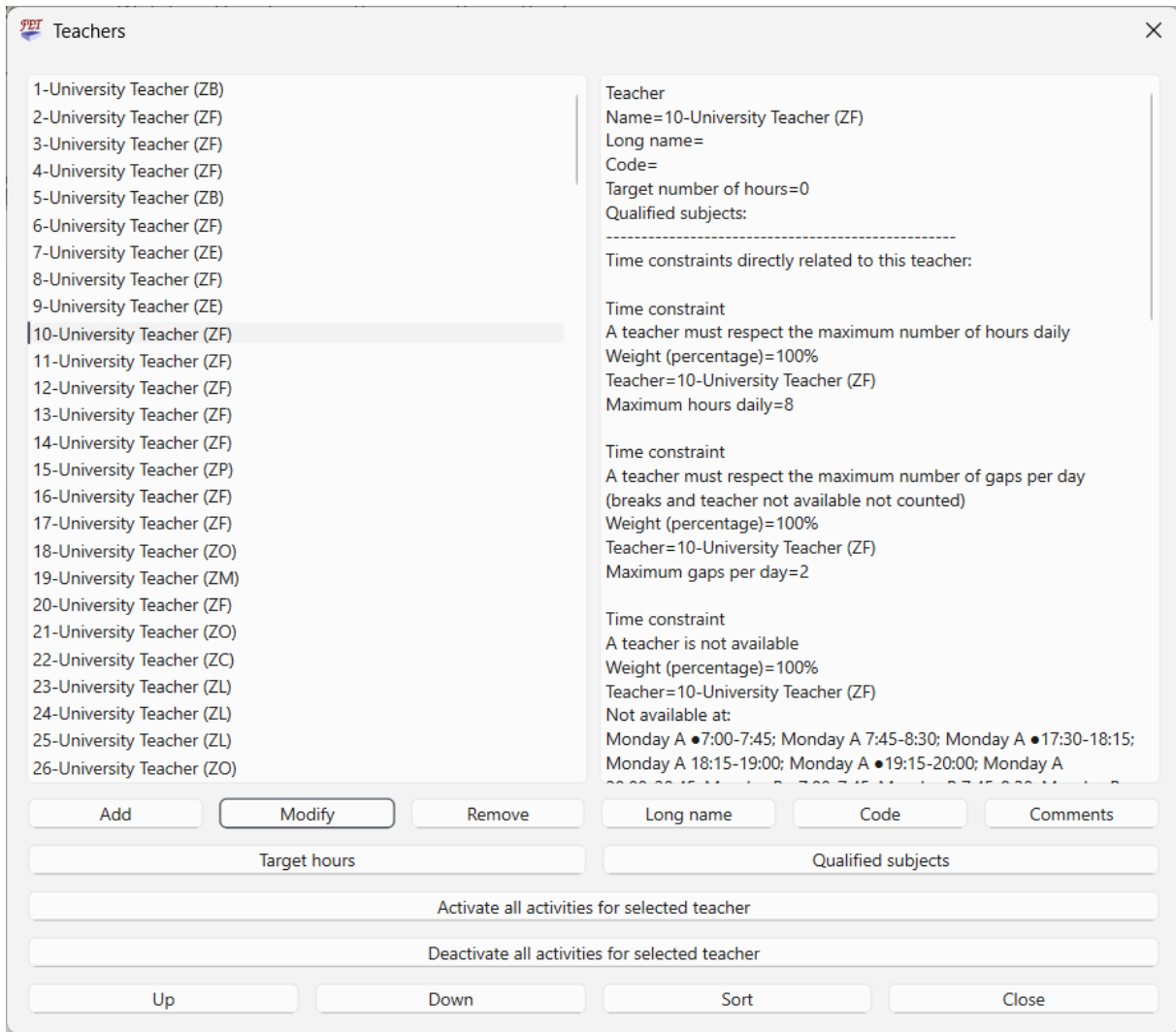


Fig. 6. University teachers with restrictions introduced in the window *Data -> Teachers*

However, student data is stored in the section `<Students_List>` `</Students_List>`. The planner can enter a set of years, groups, and subgroups for individual years, which, after loading the FET input file, can be viewed in the following windows:

Students -> Years (forms, classes) -> Years(forms, classes) of students

Students -> Groups -> FET – groups of students

Students -> Subgroups -> Subgroups of students.

An example block of code including student data is as follows:

```

<Students_List>
<Year>
<Name>2ZDL-W1</Name>
<Long_Name></Long_Name>
<Code></Code>
<Number_of_Students>125</Number_of_Students>
<Comments></Comments>
<Number_of_Categories>0</Number_of_Categories>

```

```

<Separator> </Separator>
<Group>
  <Name>2ZDL-C4</Name>
  <Long_Name></Long_Name>
  <Code></Code>
  <Number_of_Students>22</Number_of_Students>
  <Comments></Comments>
  <Subgroup>
    <Name>2ZDL-L8</Name>
    <Long_Name></Long_Name>
    <Code></Code>
    <Number_of_Students>15</Number_of_Students>
    <Comments></Comments>
  </Subgroup>
</Group>
</Year>
...
</Students_List>

```

The section <Buildings_List> </Buildings_List> stores the data of buildings, while the section <Rooms_List> </Rooms_List> stores the data of teaching rooms where classes are held:

```

<Buildings_List>
<Building>
  <Name>L</Name>
  <Long_Name></Long_Name>
  <Code></Code>
  <Comments></Comments>
</Building>
...
</Buildings_List>

<Rooms_List>
<Room>
  <Name>L.20A</Name>
  <Long_Name></Long_Name>
  <Code></Code>
  <Building>L</Building>
  <Capacity>15</Capacity>
  <Virtual>false</Virtual>
  <Comments>L-31.20A</Comments>
</Room>
...
</Rooms_List>

```

The information entered in these sections can be viewed in the Data -> Space -> Buildings / Rooms windows of the FET programme.

In addition, at the stage of preparing the FET input file, it is useful to automatically assign the appropriate room sets for each subject. Using the activity designations from Table 1, one can easily assign rooms with the appropriate capacity or purpose to dedicated subjects. These conditions should be placed in the section <Space_Constraints_List> </Space_Constraints_List>. For example, the assignment of a set of laboratory rooms available to the PRz Faculty of Management for laboratory classes is implemented by the following block of code:

```

<ConstraintSubjectPreferredRooms>

```

```

<Weight_Percentage>100</Weight_Percentage>
<Subject>Econometrics (ZDL4) - L30</Subject>
<Number_of_Preferred_Rooms>9</Number_of_Preferred_Rooms>
<Preferred_Room>L.357a</Preferred_Room>
<Preferred_Room>S.12</Preferred_Room>
<Preferred_Room>L.26_27</Preferred_Room>
<Preferred_Room>L.355</Preferred_Room>
<Preferred_Room>L.19A</Preferred_Room>
<Preferred_Room>L.20</Preferred_Room>
<Preferred_Room>L.20A</Preferred_Room>
<Preferred_Room>L.10A</Preferred_Room>
<Preferred_Room>L.354D</Preferred_Room>
<Active>>true</Active>
<Comments></Comments>
</ConstraintSubjectPreferredRooms>

```

In the next step, there is need to enter all the activities that should be included in the timetable in the specified time period. Manually entering 1169 (Table 1) differing classes is an extremely labor-intensive and error-prone operation. Therefore, automating this process is a most desirable task. The activity data is placed in the Activities_List section, which consolidates the information entered so far, including, but not limited to, the teacher, the subject name, the tag, the set of students, the frequency of the activity, and the duration of the activity (Fig. 7):

```

<Activities_List>
<Activity>
<Teacher>31-University Teacher (ZF)</Teacher>
<Subject>Financial controlling (FDL4) - P30</Subject>
<Activity_Tag>Project</Activity_Tag>
<Activity_Tag>U1</Activity_Tag>
<Students>2FDL-L1</Students>
<Duration>2</Duration>
<Total_Duration>4</Total_Duration>
<Id>138</Id>
<Activity_Group_Id>138</Activity_Group_Id>
<Active>>true</Active>
<Comments></Comments>
</Activity>
...
</Activities_List>

```

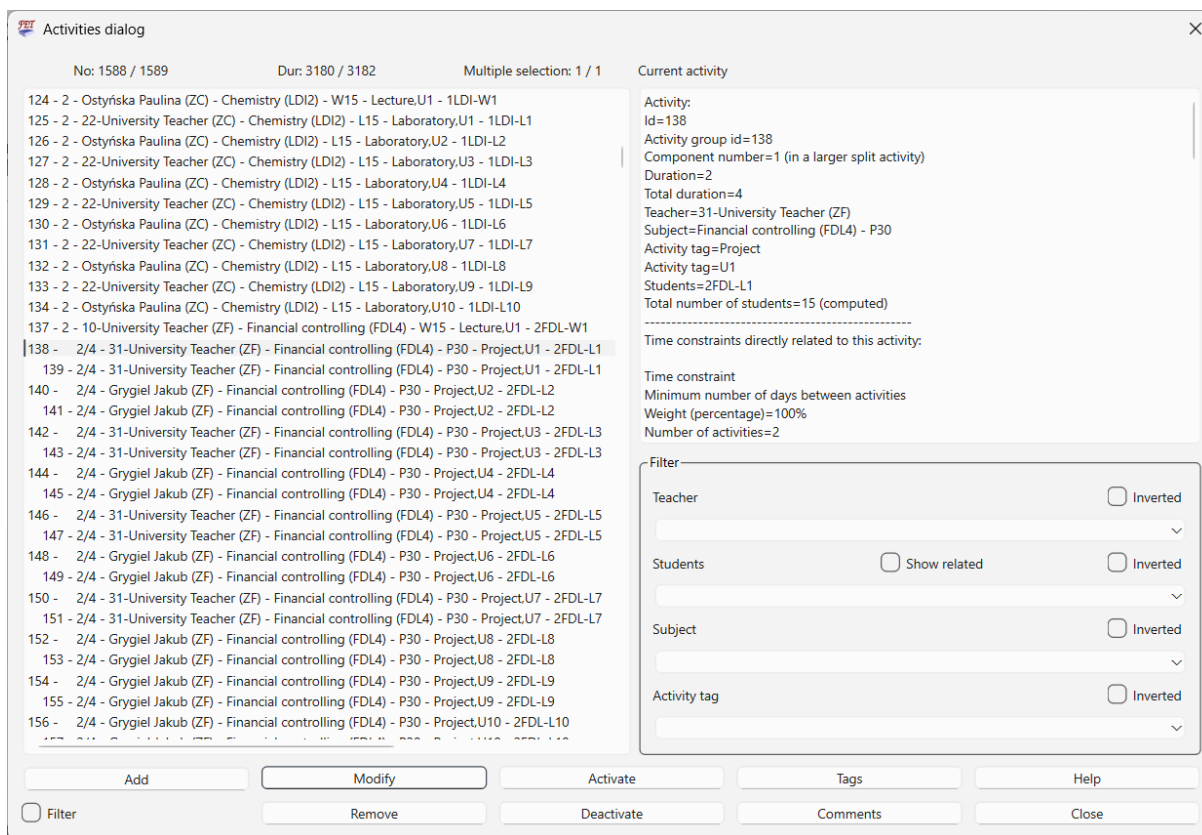


Fig. 7. Window conditions activities *Data - > Activities*

Double-click on the selected activity in the window (Fig. 7), the Modify activity window (Fig. 8) allows to display and edit information from the section <Activities_List> </Activities_List>.

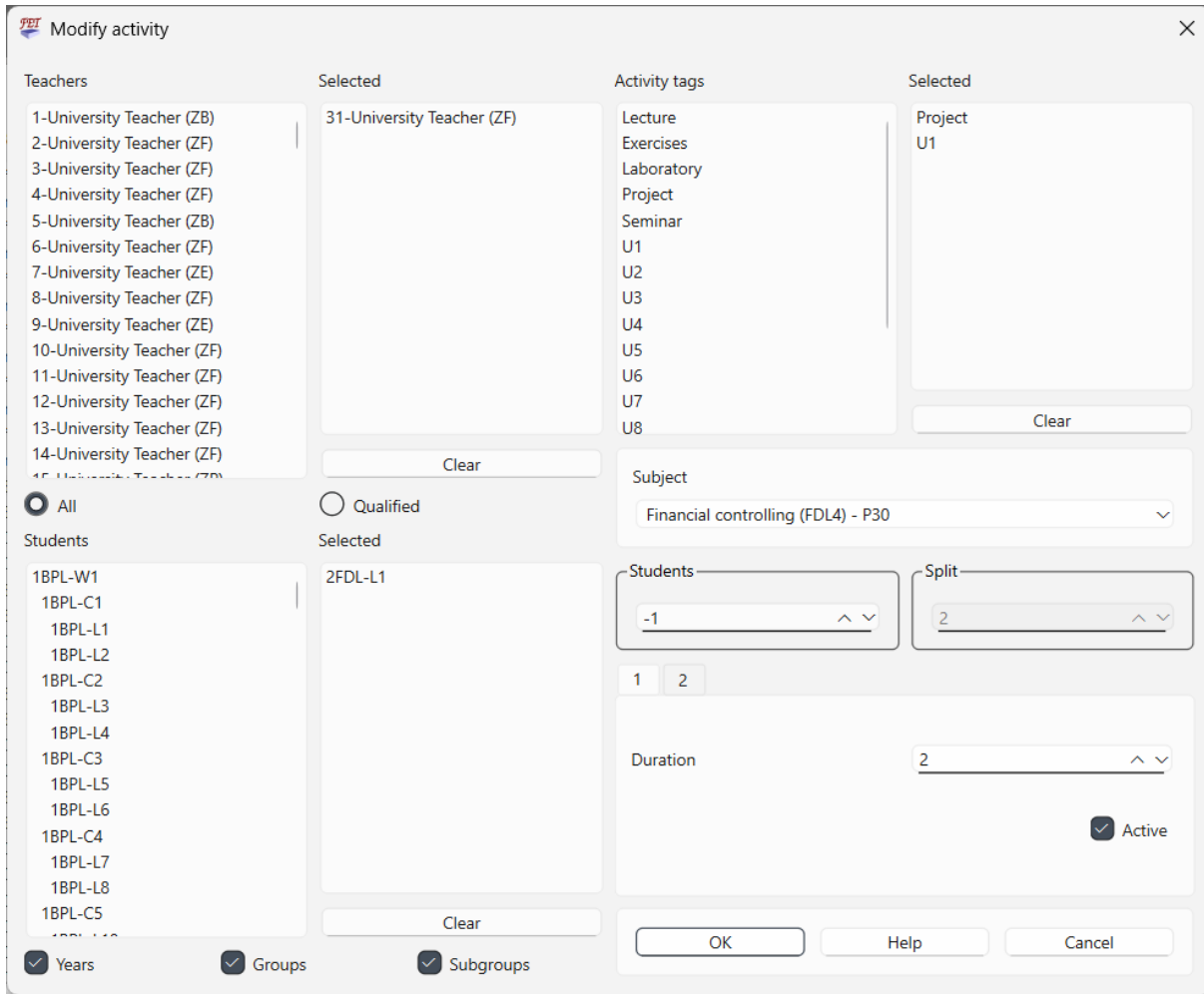


Fig. 8. Window *Modify Activity*

For classes of 30 h - held each week, the Split field is set to 2. Then for the Duration option there are two tabs, for week A and B, respectively. If classes are to be scheduled in equal intervals in both weeks - 2 teaching hours each, the Duration parameter should take the value of 2. With such settings, the *Data - > Activities* window displays two entries for classes with consecutive numbers (Fig. 7).

The creation of tools that allow the cyclic generation of an input *.fet file with entered lists of days, hours, subjects, lecturers, students, buildings, rooms, and especially activities, significantly reduces the labor-intensive input data phase, and completely eliminates the third phase of FET implementation (Fig. 1). Above all, automation of the aforementioned processes makes it possible to obtain data free of human error.

The nature of the organisation and its governing orders generate sets of constraints that should be included in the schedule. Conditions that are uniform for entire groups of academics, students or individual resources, the planner is able to capture using a programming language, thus expanding the *.fet input file.

For example, a restriction on the maximum daily teaching load for each employee can be entered in the section `<Time_Constraints_List> </Time_Constraints_List>` containing a list of various time constraints. The working day of a *10-University Teacher* is limited to 8 teaching hours as follows:

```

<ConstraintTeacherMaxHoursDaily>
<Weight_Percentage>100</Weight_Percentage>
<Teacher>10-University Teacher (ZF)</Teacher>
<Maximum_Hours_Daily>8</Maximum_Hours_Daily>
<Active>>true</Active>
<Comments></Comments>
</ConstraintTeacherMaxHoursDaily>

```

Duplicating the given code fragment for all employees guarantees significant time savings for the planner.

In addition to the necessary conditions to be included in the timetable, the planner has the option to expand the *.fet input file with optional conditions. The FET programme allows the input of a variety of time conditions, with the help of which balanced schedules can be obtained for all groups of students and employees. When trying to minimize the number of breaks between activities, the planner has conditions available: *Max gaps per day for a students set*, *Max gaps per week for a students set*, *Max gaps per day for a teacher*, *Max gaps per week for a teacher*, implemented with the example codes:

```

<ConstraintStudentsSetMaxGapsPerDay>
<Weight_Percentage>100</Weight_Percentage>
<Max_Gaps>2</Max_Gaps>
<Students>2FDL-C1</Students>
<Active>>false</Active>
<Comments></Comments>
</ConstraintStudentsSetMaxGapsPerDay>

```

```

<ConstraintStudentsSetMaxGapsPerWeek>
<Weight_Percentage>100</Weight_Percentage>
<Max_Gaps>6</Max_Gaps>
<Students>2FDL-C1</Students>
<Active>>false</Active>
<Comments></Comments>
</ConstraintStudentsSetMaxGapsPerWeek>

```

```

<ConstraintTeacherMaxGapsPerDay>
<Weight_Percentage>100</Weight_Percentage>
<Teacher>10-University Teacher (ZF)</Teacher>
<Max_Gaps>2</Max_Gaps>
<Active>>true</Active>
<Comments></Comments>
</ConstraintTeacherMaxGapsPerDay>

```

```

<ConstraintTeacherMaxGapsPerWeek>
<Weight_Percentage>100</Weight_Percentage>
<Teacher>10-University Teacher (ZF)</Teacher>
<Max_Gaps>4</Max_Gaps>
<Active>>true</Active>
<Comments></Comments>
</ConstraintTeacherMaxGapsPerWeek>

```

Automatic generation of the *.fet input file with the discussed constraints significantly improves the task of the planner. However, the types of conditions, as well as the number of conditions, should be imposed with much care and sensitivity, so as not to lead to a blockage in the operation of the FET programme. Imposing too many conditions that are difficult to implement, or introducing mutually exclusive constraints in the case of larger

universities, results in a significant increase in the running time of the programme, or interruption of its operation. Therefore, it is recommended to generate an input file for the FET programme, which will include only the necessary restrictions resulting from the university orders. Other conditions are proposed to be entered manually, for individual plan users, in the FET application window.

The FET programme has a wide range of time and place conditions to optimise the time and use of the organisational resources, while ensuring that students and employees are evenly distributed while complying with existing ordinances. Providing days off during the week for students as well as teachers, limiting movement between buildings, minimum number of hours of classes per day, are just some of those that make the FET programme a powerful and user-friendly tool to support the work of the planner.

Conclusions

The task of constructing a university timetable constitutes a serious challenge for both the planner and the organisation as a whole. The impossibility of preparing cyclic plans and the limited time available for its implementation requires a huge amount of effort and work. The complicated nature of the planning task forces the need to automate it. There are many IT tools available on the market dedicated to various planning activities. Their choice depends on the preferences of the planner and the needs of the organisation. Simplicity, intuitiveness, ease of use, speed of operation, accessibility are some of the most important features that effective software should have. One such programme is FET, which supports automatic scheduling in schools and universities, combining all the mentioned advantages of an effective application. In addition, the ability to integrate FET with other systems allows to completely improve the planning process. Saving the timetable in FET to XML format gives the user great possibilities for editing with the use of software tools. First of all, it allows automatic entry of all input data.

The paper proposes solutions that result in improvement of scheduling process by eliminating and minimizing some planning problems. Knowing the structure of the FET input file, the planner can prepare the code, run cyclically to generate it. Moreover, from the final version of the plan it is possible to retrieve the relevant information which is the input for the subsequent tools for presenting the plan. Integration of the processes of data input, activity creation, timetable generation, preparation and import of data into the information systems providing the final version of the plan significantly optimises the process of timetable creation. Automating the discussed timetable-building procedures makes it possible to quickly and efficiently generate a timetable at the beginning of each semester, which significantly facilitates and speeds up the work of the planner.

The conducted considerations allow for the formulation of several conclusions:

1. The use of input data generation algorithms, the FET scheduling programme, and the automation of the publication of the finished timetable in the information system improves the process of creating academic timetables significantly, saving time and eliminating human errors.
2. The implementation of input data processing algorithms that also include necessary constraints results in a significant reduction in the manual input process and completely eliminates the third phase of FET implementation - activity creation.
3. The proposed solutions make it possible to achieve balanced timetables, characterised by an even distribution of activities, thus affecting the quality of work of both teachers and students.
4. The FET programme ensures efficient allocation of time and space, combining the functions of scheduling classes and monitoring the use of teaching rooms.
5. The FET programme allows full use of available teaching rooms, resulting in resource savings.
6. The use of information technology makes the task of planning easier to manage and more efficient.

References

- Betaouaf, T.H., Boukli-Hacene, R. and Cherier, M.A. (2021) 'Application of automated system for university course timetable scheduling: an Algerian case study', *International Journal of Knowledge Engineering and Data Mining* [Preprint]. Available at: <https://www.inderscienceonline.com/doi/10.1504/IJKEDM.2021.119884> (Accessed: 15 April 2025).
- Das, E. *et al.* (2021) 'University timetabling optimization using FET with a sustainable and smart approach towards subgrouping process', *SGS - Engineering & Sciences*, 1(01). Available at: <https://spast.org/techrep/article/view/1540> (Accessed: 14 April 2025).
- Fahmy, S. *et al.* (2014) 'Open-source academic timetabling system: FET implementation at TATI university college', *Aust. J. Basic Appl. Sci*, 8(4), pp. 86–91.
- *FET - Free Timetabling Software* (no date). Available at: <https://lalescu.ro/liviu/fet/> (Accessed: 14 April 2025).
- Garey, M.R. and Johnson, D.S. (1983) 'Computers and Intractability. A Guide to the Theory of Np-Completeness', *Journal of Symbolic Logic*, 48(2), pp. 498–500. Available at: <https://doi.org/10.2307/2273574>.
- Joy, J. *et al.* (2023) 'Innovative Constraints Formulation in Timetable Planning for Efficient Resource Allocation in Academic Institutions', in *Sustaining Higher Education Through Resource Allocation, Learning Design Models, and Academic Development*. IGI Global Scientific Publishing, pp. 228–249. Available at: <https://doi.org/10.4018/978-1-6684-7059-6.ch010>.
- Krzeszowska, B., Trzaskalik, T. and Nocoń, E. (2010) 'Zastosowanie algorytmów genetycznych do układania planów lekcji na przykładzie Szkoły Podstawowej nr 5 w Olkuszu', *Uniwersytet Ekonomiczny we Wrocławiu* [Preprint]. Available at: <https://www.dbc.wroc.pl/dlibra/publication/160022/edition/121726> (Accessed: 14 April 2025).
- Muhamad, W.Z.A.W. *et al.* (2018) 'Solving university course timetabling problems using FET software', *AIP Conference Proceedings*, 2013(1), p. 020052. Available at: <https://doi.org/10.1063/1.5054251>.
- *Planista*, Serwisy | Portal USOS (no date). Available at: <https://www.usos.edu.pl/serwisy> (Accessed: 15 April 2025).
- *Plansoft*, Systemy rezerwacji sal, planowania rozkładów zajęć, wizyt i zasobów (no date). Available at: <https://www.plansoft.org/> (Accessed: 15 April 2025).
- Rustauletov, B. (2020) 'The Automation of Course Scheduling in Higher Education Institutions: Mathematical methods and software products', *Proceedings of International Young Scholars Workshop*, 9. Available at: <https://doi.org/10.47344/iysw.v9i0.169>.
- Thomas H Cormen (Author), C.E.L. (Author) (no date) *Introduction To Algorithms Third Edition (2009)*. Available at: <http://archive.org/details/introduction-to-algorithms-third-edition-2009> (Accessed: 14 April 2025).
- Tosi L. G. (2023), 'Passo a passo para alocação de disciplinas usando o Programa Free Timetabling Software - FET - no contexto do CCE-UFSC', *TCC Secretariado Executivo, Florianópolis, SC*.
- *UNIplaner - dla uczelni - Sages: wdrożenia systemów dla uczelni: Omega-PSIR, USOS, Uniplaner, WCAG AI, Chatbot oraz innych opartych na Sztucznej Inteligencji (no date) Sages*. Available at: <https://www.sages.pl/uniplaner> (Accessed: 15 April 2025).
- *USOS*, Strona główna | Portal USOS (no date). Available at: <https://www.usos.edu.pl/> (Accessed: 15 April 2025).