MŰEGYETEM 1782

Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

Implementation of the optimization of multi-variable target functions through an example timetabling application

BSc. Thesis

Author
Viktória Nemkin

Supervisors
dr. Márk Asztalos, András Kárpinszky

December 7, 2018

Contents

HALLGATÓI NYILATKOZAT

Alulírott *Viktória Nemkin*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, December 7, 2018

Viktória Nemkin
hallgató

Kivonat

Az órarendtervezés elmélete rendkívül kidolgozott. Rengeteg algoritmus és matematikai elmélet létezik melyek felhasználásával órarendet lehet tervezni elméletben. Sajnos a gyakorlatban viszont az órarendeket legtöbbször kézzel készítik. Ez igaz a Budapesti Műszaki és Gazdaságtudományi Egyetemre is, ahol a minden kar saját órarendkészítő munkatársa foglalkozik ezzel a feladattal. Ez rengeteg időt és energiabefektetést igényel a részükről.

A szakdolgozatom témája az egyetemi órarendkészítés automatizálása. Mivel ez egy komplex feladat, rengeteg különféle igényt kell kielégítenie egy ideális órarendnek, ezért olyan módszert kell választani amely segítségével képesek vagyunk általánosan, matematikailag megfogalmazni ezt a problémát és olyan megoldást adni rá, mely a későbbiekben tovább bővíthető az újonnan felmerülő igényeknek megfelelően.

A választott módszer a kényszerprogramozás. A kényszerproblémák matematikai megfogalmazást adnak egy feladathoz, mely változókból, a változók értékkészletéből és egyszerű műveletekkel megadott kényszerekből, egyenletekből állnak. Ezek segítségével leírhatók tetszőlegesen bonyolult feltételrendszerek. Egy ilyen eszközkészlet segítségével programozottan lehet megoldásokat generálni a probléma leírása alapján.

Abstract

The theory timetable design is widely elaborated. There are plenty of algorithms and mathematical principles that can be used to create timetables in theory. Unfortunately, in practice, timetables are mostly hand-made. This is true for the Budapest University of Technology and Economics, where the faculties employ their timetable creators who deal with this task completely manually.

The subject of my dissertation is the automation of university timetable creation. This is a complex task since the ideal timetable has to meet several different requirements and needs, so we need to mathematically formulate this problem to be able to provide a solution that can be further expanded to meet emerging needs.

The method chosen is constraint programming. Constraint programming problems provide a mathematical description for a task consisting of variables, domains and mathematical formulae of constraints. They can be used to describe arbitrarily complex conditions. Using such a toolkit, one can programmatically generate solutions based on the description of the problem.

Introduction

Timetable planning for universities is a complex procedure due to the significant number of variables and conditions that have to be taken into account. In theory, there are many algorithms and methods to solve this problem, but in practice, most people use last year's timetable and manually change where needed.

Similarly, at our university, this process is done entirely by hand. Software is only used to validate the result and check for collisions. Each faculty has a designated timetable creator who devises their schedule every semester. For the Faculty of Electrical Engineering and Informatics, this means manually creating 9 majors' schedules for roughly 5000 students every single term. It is a tedious and repetitive task with no sense of reward or accomplishment, and it requires a substantial amount of time to complete.

This is why I decided to automate it.

I aim to create a software solution designed specifically for the timetabling needs of the Faculty of Electrical Engineering and Informatics of the Budapest University of Technology and Economics. This software shall be capable of loading course data, a list of teachers and rooms available and then producing a timetable that meets the specific requirements for the given semester.

The first chapter analyses the problem in detail, describes the needs of our faculty, then declares a subset of those as requirements to meet during the semester. The second chapter talks about the current market of timetabling software and why these are inept for us. The third part addresses the mathematical background required for solving the task. The fourth chapter describes the architecture of the software. The fifth part discusses the problems that arose during the semester and the solutions for them. The sixth chapter examines the results.

Chapter 1

Requirement specification

I have met with Erzsébet Győri, the timetable creator of our faculty and we discussed how she currently creates the timetables and what specific requirements and requests arise during the process and what my system shall be capable of.

Below is a summary of these.

- The user shall be able to record the buildings and rooms in the system.
- The user shall be able to record the departments, their faculty members and which classes and course types they can teach.
- The user shall be able to record the years of the students.
- The user shall be able to record the classes for every year and the courses from the given classes.
- Every practice session shall be able to be scheduled independently.
- For every timeslot, the maximum amount of classes shall not have more capacity than the entire year of students.
- During breaks students shall be able to comfortably walk from the current room to the next in the available time. Commutes from building K to building I and back shall be reduced or eliminated.
- For practice sessions of a given class, all of them shall be either before or preferably after the seminar for the given week.
- Some practice sessions have an assigned teacher, and some don't (for student demonstrators). When the teacher is assigned the class shall not conflict with another class of the same teacher.
- Some faculties have a low number of student demonstrators so they can't handle a large number of parallel practice sessions.

- Every department and faculty member shall be able to specify when they are available to teach and be given a schedule accordingly.
- For all courses, the departments shall be able to specify when they can be scheduled.
- Every faculty member shall be able to specify the locations (buildings) they would prefer to teach in.
- Some classes have multiple seminars and designated practice sessions depending on the chosen seminar. These practice sessions shall be scheduled depending on their corresponding seminars.
- We have rooms with capacities, described as XL (400 students), L (200 students), M (100 students), S (35 students), and XS (20 students) and special laboratory rooms. Each course shall have a suitable sized room assigned to it.
  - XL rooms are suitable for holding large seminars for the first few years of students.
  - L rooms are fitting for specialisation seminars for students in a higher year.
  - M rooms are good for smaller seminars, consultations or electives.
  - S rooms are for practice sessions.
  - XS rooms are suitable for a smaller number of students, like IMSC groups.
  - Laboratory rooms are either owned and assigned by the departments or made available by the HSZK (in building R).
- Some rooms are only available for a given period (Q-I, for example - is shared with GTK). The software shall be able to take room availabilities into account.
- The software shall be able to integrate with the Neptun system. It shall be able to download the list of courses for a given semester, schedule them and output the result in a suitable format for Neptun.
- Laboratory courses have a preferred order during the week. This is so they don't have to switch back and forth between the equipment.
- Teacher's schedules shall not have more than a specified number of hours of class for any given day.
- Exam periods shall be scheduled as well.
- For the first year students are arranged in groups. Each student group has an assigned timetable so they take their practice sessions and laboratory classes together.
- The schedule shall not have hole-hours for first-year students.
- Student groups have a lunch break from 12 pm to 1 pm.

- IMSC student groups shall be taken into account. They are separated and they, have their own timetable.
- There shall be separate software for the data entry (client) and the solver (server).
- The client shall be able to run on low resources on older Windows versions as well as Windows 10.
- The server shall run on Linux and make use of multiple available processors.
- The generated schedules shall have no conflicts for the years of students, the faculty members and the rooms.

The requirements above are complex and expansive. For the current semester I decided to focus on the following subset of these:

- The user shall be able to record the buildings and rooms in the system.
- The user shall be able to record the departments, their faculty members and which classes and course types they can teach.
- The user shall be able to record the years of the students.
- The user shall be able to record the classes for every year and the courses from the given classes.
- The user shall be able to record different types of classes with corresponding types of rooms (for example seminar rooms, practice session rooms and laboratory rooms).
- Every practice session shall be able to be scheduled independently.
- For every timeslot, there should be a specific amount of parallel courses.
- There shall be separate software for the data entry (client) and the solver (server).
- The client shall be able to run on low resources on older Windows versions as well as Windows 10.
- The server shall run on Linux and make use of multiple available processors.
- The generated schedules shall have no conflicts for the years of students, the faculty members and the rooms.

The most important requirement being the last one: generate schedules with no conflicts for all parties. This requirement will be the focus of my thesis.

Chapter 2

Market analysis

Few timetable creator products exist on the market. The most notable ones are Timetabler [6] and UniTime [7].

Timetabler is a corporate software which retails for 363.004 HUF. It is a single application that runs on Windows. This is unfortunate since we would prefer to separate the data entry and the solver from each other. Data entry will be done on a personal computer by a human, for an extended period with many interruptions. High-performance computing with expensive hardware is only required for the solver. This can run on a better quality server on campus.

Many advanced features are missing from this software that would be needed for our university. For example, you can't restrict locations for classes, can't minimise travel time between classes and there is no way to restrict the number of parallel practice sessions (where the amount of student demonstrators is limited for a department).

UniTime is an open source application available on GitHub, mainly written by Tomáš Müller from Purdue University in the United States. It is an excellent application, with many features, like a separate client for data entry and a server for running the optimiser. Sadly, the structure of higher education in the United States is very different from how it is in Hungary, and this application does not account for some basic needs our university has. Most notably they only try to minimise student class conflicts. However, we are required to eliminate conflicts in our sample curriculum for every semester. Moreover, students are considered individually and not in a year group.

Neither of these has the band system we have in place: exam timeslots, elective timeslots and timeslots for specialisations. There is no way to specify the order of classes. For example, practice sessions should follow seminars, and laboratory classes that require specialised equipment have a strict order, so the teachers don't have to shift equipment back and forth. It can't say there is no teacher assigned yet. It can't restrict the number of parallel practice sessions for departments with a small number of student demonstrators. Most notably neither of these communicates with Neptun, which is crucial for our application.

No software currently available on the market can provide every feature needed for our university and faculty. This is why we currently create the timetables manually.

I believe it makes sense to start a project to create a custom software tailored specifically for our needs and maybe in a few years it could actually be used to generate our schedules automatically.

Chapter 3

Theory

3.1 Operations research

Operations research [5] uses applied mathematics to give optimal solutions to complex decision-making problems maximising profitability, performance and yield or minimising loss, risk and cost of real-world objectives.

It originates from British military efforts during World War II, where it was described as ”a scientific method of providing executive departments with a quantitative basis for decisions regarding the operations under their control”, and it has grown to help a variety of industries.

Operations research focuses on practical real-world applications, for example [5]:

- Critical path analysis: identifying processes that affect the overall duration of the project.
- Floorplanning: designing factory settings to reduce the cost of manufacturing or computer chip layouts to reduce energy cost.
- Public transportation: determining the routes and schedules of buses so that as few buses are needed as possible.
- Supply chain management: managing the flow of raw materials and products based on uncertain demand for the finished products.

Timetable planning is a problem that can be solved using methods from operations research.

3.2 Constraint Satisfaction Problems

One of the techniques in operations research is Constraint Programming.

Formally, we define Constraint Satisfaction Problems as follows [4]:

A CSP is a triplet $\{X, D, C\}$, where:

$X = \{x_1, ..., x_n\}$	is a set of variables.
$D = \{d_1, ..., d_n\}$	is a set of domains, so that $\forall i$, $1 \leq i \leq n$, $x_i \in d_i$.
$C = \{c_1, ...c_m\}$	is a set of constraints on the variables in X.

The constraints used in constraint programming are of various kinds: those used in constraint satisfaction problems (e.g. "A or B is true"), linear inequalities (e.g. "x <= 5"), and others. Constraints are usually embedded within a programming language or provided via separate software libraries.

Constraints differ from the common primitives of imperative programming languages in that they do not specify a step or a sequence of steps to execute, but rather the properties of the solution to be found. This makes constraint programming a form of declarative programming. Using this approach we can mathematically describe the problem with variables and equations. Unlike other approaches, for example using the method of colouring a bipartite graph for timetable creation this method provides flexibility with the type of features we can build into the system. With constraints we can express a wide range of requirements and we can be sure if new requests come in we will be able to include them in the system. [3]

This is why constraint programming is a good choice for solving this problem.

Chapter 4

Realisation

In the following chapter I talk about the development process of the system.

4.1 Language, framework, toolkit

There are many programming languages, frameworks and libraries that I considered when choosing my toolkit.

I wanted the server to be able to run on Linux because it is open source and free, so there is no need to buy a license for the operating system of choice. I expect our university to have Linux servers that they can use to run the server or a virtual machine can be created anywhere that runs Linux, without the need to run a UI.

I chose C++ and Qt as a platform. I decided not to use Microsoft technologies, .NET and C# because even though they are opening to Linux, I believed C++ and Qt was a better, more supported option. I did not want to go with Java because C++ is much faster and more suited for algorithmically complex tasks. The solver could have used a declarative language like Prolog or Erlang, but I wanted to store the information in a database which is not very convenient in these languages. Qt is a suitable choice for the client since it supports multiple platforms seamlessly, so it can run on Windows and Linux as wanted.

I choose PostgreSQL as a database driver. I needed something that works seamlessly on Linux and can run separately from the server. This way the database can be installed on a computer with good network connections and large storage capabilities, independently of the server and the client. The database has to be online continuously so the client can upload new data to it while the server needs to be run only once.

Several libraries can be used for constraint programming. I chose Google's OR-Tools [1] [2] because it is completely open source and it has multiple algorithms implemented in it, like linear programming plus many graph algorithms which makes it a versatile tool for operations research.

4.2 Data model

I assumed that the data model would be suspect to several changes as I develop the solver. In retrospect, this was true.

I wanted to create a framework where I could quickly change my data model depending on my needs and iterate over the different versions. I needed the data to be persistent and to have matching database tables and queries in SQL for manipulation.

To achieve this, I created a json format to describe my data.

The following format describes the C++ and SQL equivalent for every type in my system.

```
{
    "id": {
        "cpp": "int",
        "sql": "SERIAL PRIMARY KEY",
        "default_value": "0"
    },
    "reference": {
        "cpp": "int",
        "sql": "INTEGER",
        "default_value": "0"
    },
    "int": {
        "cpp": "int",
        "sql": "INTEGER",
        "default_value": "0"
    },
    "double": {
        "cpp": "double",
        "sql": "DOUBLE PRECISION",
        "default_value": "0.0"
    },
    "text": {
        "cpp": "std::string",
        "sql": "TEXT",
        "default_value": "\"\""
    },
    "bool": {
        "cpp": "bool",
        "sql": "BOOLEAN",
        "default_value": "false"
    },
    "timestamp": {
        "cpp": "std::string",
        "sql": "TIMESTAMP",
        "default_value": "\"\""
    }
}
```

Then, using these types the json file below describes my data model.

```
{
    "default": {
        "members": [
            {
                "name": "modified_timestamp",
                "type": "timestamp"
```

```
            },
            {
                "name": "is_deleted",
                "type": "bool"
            }
        ]
    },
    "classes": [
        {
            "class": "timeslot",
            "members": [
                {
                    "name": "name",
                    "type": "text"
                }
            ],
            "references": []
        },
        {
            "class": "location",
            "members": [
                {
                    "name": "name",
                    "type": "text"
                }
            ],
            "references": []
        },
        {
            "class": "class_type",
            "members": [
                {
                    "name": "name",
                    "type": "text"
                }
            ],
            "references": []
        },
        {
            "class": "year",
            "members": [
                {
                    "name": "name",
                    "type": "text"
                }
            ],
            "references": []
        },
        {
            "class": "room",
            "members": [
                {
                    "name": "name",
                    "type": "text"
                },
                {
                    "name": "size_type",
                    "type": "text"
                }
            ],
```

```
            "references": [
                {
                    "class": "location"
                },
                {
                    "class": "class_type"
                }
            ]
        },
        {
            "class": "department",
            "members": [
                {
                    "name": "name",
                    "type": "text"
                },
                {
                    "name": "short_name",
                    "type": "text"
                }
            ],
            "references": []
        },
        {
            "class": "course",
            "members": [
                {
                    "name": "name",
                    "type": "text"
                }
            ],
            "references": [
                {
                    "class": "year"
                },
                {
                    "class": "department"
                }
            ]
        },
        {
            "class": "class",
            "members": [
                {
                    "name": "name",
                    "type": "text"
                }
            ],
            "references": [
                {
                    "class": "class_type"
                },
                {
                    "class": "course"
                }
            ]
        },
        {
            "class": "faculty_member",
            "members": [
```

```
                {
                    "name": "name",
                    "type": "text"
                }
            ],
            "references": [
                {
                    "class": "department"
                }
            ]
        },
        {
            "class": "license",
            "members": [],
            "references": [
                {
                    "class": "course"
                },
                {
                    "class": "class_type"
                },
                {
                    "class": "faculty_member"
                }
            ]
        }
    ]
}
```

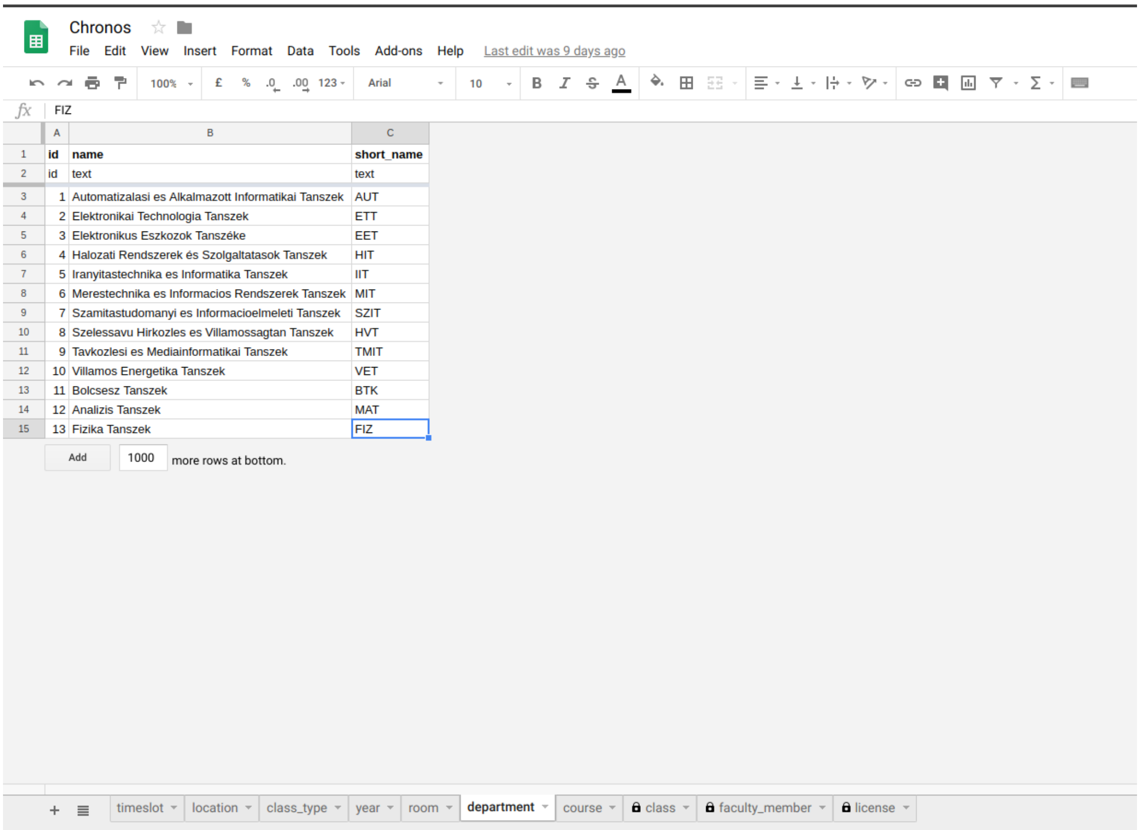
Every class has an id member which is the primary key in the database table for that class. Then, the default section describes the members every class has, namely the last modification timestamp and a delete toggle for soft deletion.

The next section describes my classes. The timeslot is responsible for storing the different class periods (for example, Monday 8 am to 10 am, or Friday 10 am to 12 pm). The location class stores the buildings. The class type stores if the class is a seminar or practice session or a laboratory class. Year describes the different student years in the computer engineering major. Room stores the rooms on campus, which references the location the room is in and the type of class the room can hold. Department stores the faculty's departments. Course stores the different courses that are offered for the major (for example, Probability Theory or Theory of Algorithms). Class stores all the classes that are available from a specific course. The faculty member stores the teachers, referencing the department they are part of. The license indicates if a faculty member can teach a course, of the given class type.

4.3 Automatic code generation

Using these json descriptors, I wrote a Python script that generates the corresponding C++ classes and the database class responsible for interfacing to the PostgreSQL database. This way every time the data model changed throughout the development of the solver algorithm I was able to quickly change the json descriptor and generate a persistent data model for it without spending days changing the code manually.

Then I created a Google Drive Sheet for the data model, seen in Figure 4.1.



	A	B	C
1	id	name	short_name
2	id	text	text
3	1	Automatizalasi es Alkalmazott Informatikai Tanszek	AUT
4	2	Elektronikai Technologia Tanszek	ETT
5	3	Elektronikus Eszkozok Tanszéke	EET
6	4	Halozati Rendszerek és Szolgaltatasok Tanszek	HIT
7	5	Iranyitastechnika es Informatika Tanszek	IIT
8	6	Merestechnika es Informacios Rendszerek Tanszek	MIT
9	7	Szamitastudomanyi es Informacioelmeleti Tanszek	SZIT
10	8	Szelessavu Hirkozles es Villamossagtan Tanszek	HVT
11	9	Tavkozlesi es Mediainformatikai Tanszek	TMIT
12	10	Villamos Energetika Tanszek	VET
13	11	Bolcsesz Tanszek	BTK
14	12	Analizis Tanszek	MAT
15	13	Fizika Tanszek	FIZ

Figure 4.1: *Google Drive Sheet*

The worksheets represent the classes in the data model. The first row lists the names of the variables, the second row the types of them. After that, every row is one object from the class or one row in the database. I wrote a Google Apps Script that will generate the descriptor jsons and an SQL script that creates the database structure and initialises the tables with the data from the worksheets.

Now, every time I needed to change the data model or the specific object I was able to open this spreadsheet, quickly edit it the way I wanted to, then run a few generator scripts and have my database ready and code modified to match it.

This quick prototyping method allowed me to make changes faster throughout the semester and played a huge role in completing my thesis on time.

4.4 Mathematical description of the problem

In the following section I will describe the Constraint Satisfaction Problem that I used to solve the timetabling problem.

Let c_c denote the number of classes, t_c the number of timeslots r_c the number of rooms, f_c the number of faculties and p_c the number of allowed parallel non-seminar classes.

4.4.1 Variables

The variables are assigned in the following way.

T_i is the timeslot assigned to the ith class where $1 \le i \le c_c$.
R_i is the room assigned to the ith class where $1 \le i \le c_c$.
F_i is the faculty member assigned to the ith class where $1 \le i \le c_c$.

Then, there are special uniqueness variables. These variables help us to describe unique constraints in the system.

U_{TR_i} is a uniqueness variable on the timeslot and room pair assigned to the ith class where $1 \le i \le c_c$. These variables make sure rooms are not overbooked in any timeslot.

U_{TF_i} is a uniqueness variable on the timeslot and faculty member pair assigned to the ith class where $1 \le i \le c_c$. These variables make sure the faculty members don't have to teach two classes at the same time.

$U_{TYP_{i,j}}$ is a uniqueness variable on the timeslot, year and parallelness triplet assigned to the ith class where $1 \le i \le c_c$ and jth parallel timeslot where $1 \le j \le p_c$. These variables make sure that no year of students has to go to two classes at the same time.

P_i is a parallelness variable for non-seminar classes. This allows the years of students to have parallel non-seminar classes in their timetable.

4.4.2 Domains

The domains of the variables are as follows.

$T_i \in (1, t_c)$
$R_i \in (1, r_c) \cap \{$"where the room type matches the course type"$\}$
$F_i \in (1, f_c) \cap \{$"which class the faculty member is licensed to teach"$\}$

$U_{TR_i} \in (1, (t_c + 1)(r_c + 1))$, $1 \le i \le c_c$
$U_{TF_i} \in (1, (t_c + 1)(f_c + 1))$, $1 \le i \le c_c$
$U_{TYP_{i,j}} \in (1, (t_c + 1)(y_c + 1)(p_c + 1))$, $1 \le i \le c_c$, $1 \le j \le c_c$

4.4.3 Constraints

The constraints are below.

The uniqueness variables are number pairs represented as one number in a t_c base-number system.

$U_{TR_i} = R_i(t_c + 1) + T_i$
$U_{TF_i} = F_i(t_c + 1) + T_i$

The U_{TYP_i} variable uses the same representation trick but for a triplet. When the ith class is a non-seminar class:
$U_{TYP_{i,1}} = Y_i(p_c + 1)(t_c + 1) + P_i(t_c + 1) + T_i$

When the ith class is a seminar class we have to occupy ever parallel timeslot. This is done by creating p_c number of uniqueness variables for seminars.
$U_{TYP_{i,j}} = Y_i(p_c + 1)(t_c + 1) + j(t_c + 1) + T_i$, $1 \leq j \leq p_c$

The following constraints make sure that the pairs and triplets are unique.

AllDiff(U_{TR_i}) $\forall i$
AllDiff(U_{TF_i}) $\forall i$
AllDiff($U_{TY_{i,j}}$) $\forall i, j$

4.4.4 Optimalization

To optimalize the devised timetable, I wanted to minimize the number of hole-hours for the students. To achieve this, we have to mathematically describe this using the variables in the Constraint Satisfaction Problem description above.

Theorem Let $S_1, S_2, ..., S_{y_c}$ be the sets of the classes for a given year.
Minimizing $H_k = \sum_{i \in S_k} \sum_{j \in S_k} |T_i - T_j|$ will minimize the number of hole-hours in the kth year's timetable.

Lemma If the kth year's timetalbe with value $H_k = h$ contains a hole-hour there is always a timetable with value $H_k = h^*$ so that $h^* < h$.

Proof Let's move the chronologically first class, $min_f T_f$ in the timetable to the hole-hour. Then let's look at the changes in the H_k value after this move. The Y_i, Y_j pairs where $i \neq f$ and $j \neq f$ did not change, so their contribution to the sum did not change either.

The only pairs that changed are the ones where one of them was T_f.

Since T_f was the first class, the value of T_f could only increase when moved to the hole-hour.

Let T_s denote the chronologically second class in the timetable. If we increase T_f's value until $T_f = T_s$ the sum of every (T_f, T_x) pair's distances decreased, since $T_s \leq T_x$.

Then, let's denote the chornologically thrid class as T_t. If we increase T_f's value until $T_f = T_t$ there are two cases. The first case is with T_s. Since T_f is getting further away from T_s, their distance is increasing. However, if we pair T_s up with the chronologically last class, T_l, the sum of their distances from T_f stays constant. The other case is with every other T_x, where the distance is decreasing.

We can continue this logic until a) we reach a hole-hour and put T_f there, or b) until we leave the first half of the timetable.

In case a) we have successfully proven that the sum of all pairs's distances decreased by putting T_f in the hole-hour. In case b) we know that hole-hour exists in the second half of the timetable. Here, we need to do the same process but instead of the chronologically first class we use the chronologically last and decrease its value, so from this viewpoint the hole-hour will be in the first half of the timetable.

This proves the correctness of the lemma.

Now, if for every timetable that contains a hole-hour we have proven that the H_k value can't be minimal. This means that only timetables with no hole-hours can minimize the H_k value, so minimizing for this results in hole-hour-less timetables.

This is the constraint I used for optimizing the timetables in the solver.

4.5 Client

The client is a simple Qt application, with minimal GUI.

The first picture shows the login screen of the application. I used PostgreSQL's user authentication. The database is installed on another computer, with IP 10.240.2.125.

The second picture shows the simple, basic GUI for the application, written in Qt.

It can display the data, but currently it will not upload it back to the database.

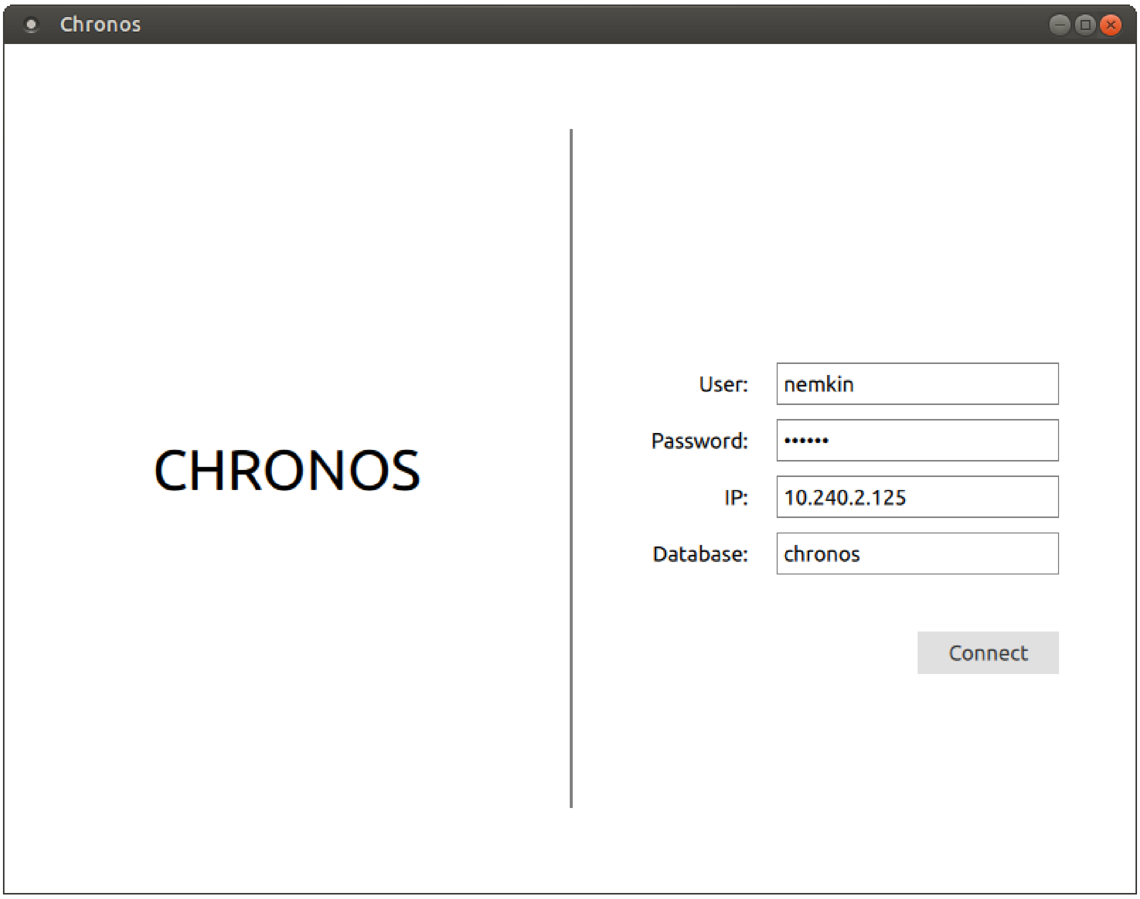


Figure 4.2: *Login Screen*

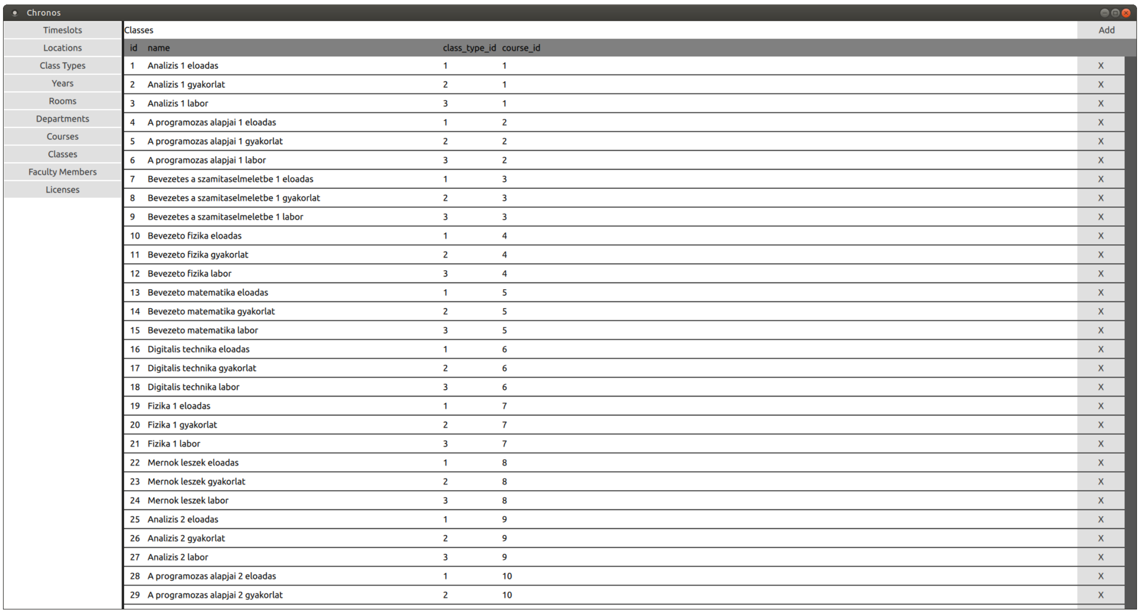


Figure 4.3: *Inside*

Chapter 5

Results

5.1 Hardware

I used special hardware for testing the solver's capabilities.

Here is the specification of the hardware:

The CPU has 32 cores, which allowed me to test for many multithreaded settings.

```
Architecture:           x86_64
CPU op-mode(s):         32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 32
On-line CPU(s) list:    0-31
Thread(s) per core:     2
Core(s) per socket:     8
Socket(s):              2
NUMA node(s):           2
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  45
Model name:             Intel(R) Xeon(R) CPU E5-2660 0 @ 2.20GHz
Stepping:               7
CPU MHz:                1200.305
CPU max MHz:            3000,0000
CPU min MHz:            1200,0000
BogoMIPS:               4389.29
Virtualization:         VT-x
L1d cache:              32K
L1i cache:              32K
L2 cache:               256K
L3 cache:               20480K
NUMA node0 CPU(s):      0-7,16-23
NUMA node1 CPU(s):      8-15,24-31
Flags:                  fpu vme de pse tsc msr pae mce cx8 apic sep
                        mtrr pge mca cmov pat pse36 clflush dts acpi mmx
                        fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp
                        lm constant_tsc arch_perfmon pebs bts rep_good nopl
                        xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64
                        monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm
                        pcid dca sse4_1 sse4_2 x2apic popcnt tsc_deadline_timer
                        aes xsave avx lahf_lm epb kaiser tpr_shadow vnmi
```

```
                        flexpriority ept vpid xsaveopt dtherm ida arat pln pts
```

It also had a large amount of memory available for use.

```
              total        used        free      shared  buff/cache   available
Mem:           125G        789M        120G        147M        4,1G        123G
Swap:            0B          0B          0B
```

5.2 Analysis

I ran the solver for 2 hours on 1,2,4,8,16,32 and 64 threads sequentially.

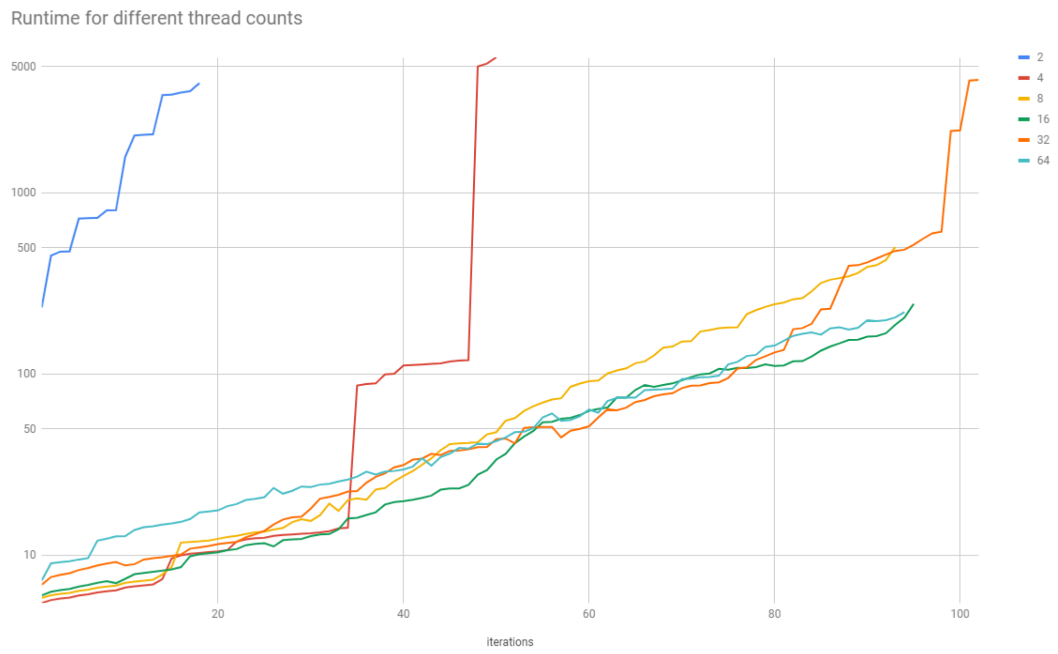


Figure 5.1: *Runtime*

In the graph above we can see the runtime for the different amount of threads. The X axis represents the number of iterations the solver was able to achieve. The Y axis is the runtime needed for that amount of iterations.

I did run the solver on 1 thread, but it was not able to return a feasible solution, so it is not shown on the graphs.

We can clearly see that 2 threads is really slow, it needs around 5000 seconds for 20 iterations. 4 threads looks better, it was able to run 40 iterations in the same amount of time. 8, 16 and 64 threads achieved the same results. 64 threads is interesting, since it is double the amount of cores in the computer. This clearly shows that when we use more threads than available computing units the context switch takes up a large amount of time, slowing down the entire application.

As we can see, 32 threads performed the best, achieving the most iterations, more than 100 in the given amount of time.

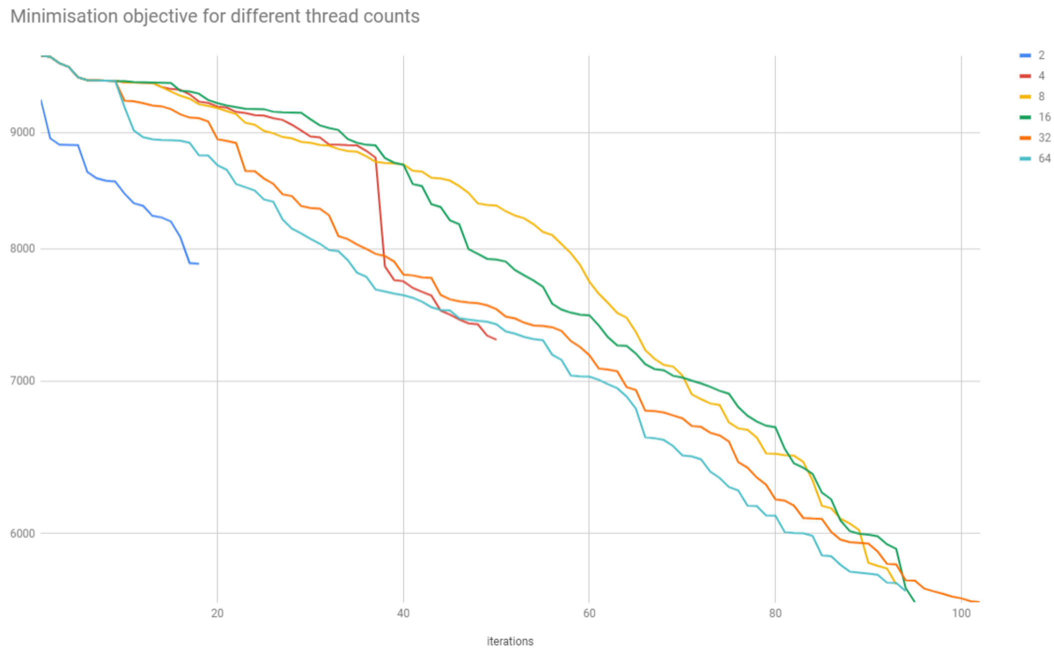


Figure 5.2: *Objective*

In the graph above we can see the objective achieved for every iteration the solver was able to run. The X axis represents the number of iterations and the Y axis the objective achieved.

As we can see, 2 threads achieved nothing. It was too slow, did not run enough iterations to find better solutions. 4 threads performed better, but still was far from ideal. 8, 16 and 64 threads look the same, 32 performing slightly better, with a better objective as its result.

5.3 Summary

The analysis above demonstrates that the solver benefits from a large number of parallel threads, which means performance and runtime can be boosted by adding more cores to the system. It also shows a basic principle in computing: more threads than cores will result in slower runtimes for all threads.

Chapter 6

Final words

The current state of the application is a good start. It is a proof of concept that timetable planning for Budapest University of Technology and Economics could be and should be automated.

In the current state, it will not be able to suit the needs of the faculty. The solver does not schedule exams, it does not take specialisation classes or electives into account. The rooms are only distributed depending on their type (large seminar halls, practice session rooms, laboratories) but there is no way to specify the capacity to account for every student. The client not usable to a non-expert in the current state. It can't account for the requests from the faculty members, such as building preference and their schedules.

In the future, I plan on adding more functionality to this software as part of my master's degree thesis.

Acknowledgement

I would like to thank Erzsébet Győri, the timetable creator for the Faculty of Electrical Engineering and Informatics at Budapest University of Technology and Economics for our extensive discussion on the topic; my supervisors dr. Márk Asztalos and András Kárpinszky for providing me with guidance and my father, Róbert Nemkin for lending me his server to test my application and giving me insight on the topic.

Bibliography

[1] Google Inc. Or-tools documentation. https://developers.google.com/optimization/.

[2] Google Inc. Or-tools on github. https://github.com/google/or-tools.

[3] Wikimedia Foundation Inc. Constraint programming. https://en.wikipedia.org/wiki/Constraint_programming.

[4] Wikimedia Foundation Inc. Constraint satisfaction problem. https://en.wikipedia.org/wiki/Constraint_satisfaction_problem.

[5] Wikimedia Foundation Inc. Operations research. https://en.wikipedia.org/wiki/Operations_research.

[6] October ReSolutions Ltd. Timetabler. https://www.timetabler.com/.

[7] Purdue University Tomáš Müller. Unitime. https://www.unitime.org/.

Appendix

F.1 Example timetable result

```
year
id:                 1
modified_timestamp: 2018-12-06 02:57:52.725823
is_deleted:         0
name:               1. felev
```

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
Fizika 1 eloadas E1B FIZ eloadastarto 1 1. felev	Analizis 1 labor R4J MAT labortarto 1 1. felev A programozas alapjai R4C EET labortarto 1 1. felev Bevezeto fizika labor R4A FIZ labortarto 1 1. felev Digitalis technika lab R4I MIT labortarto 1 1. felev Fizika 1 gyakorlat QBF08 FIZ gyakorlattarto 1 1. felev			
A programozas alapjai QBF08 EET gyakorlattarto 1 1. felev	Analizis 1 gyakorlat E401 MAT gyakorlattarto 2 1. felev Bevezetes a szamitasel QBF09 SZIT gyakorlattarto 1 1. felev Bevezetes a szamitasel R4A SZIT labortarto 1 1. felev Bevezeto fizika gyakor E402 FIZ gyakorlattarto 1 1. felev Bevezeto matematika gy QBF12 MAT gyakorlattarto 1 1. felev			
A programozas alapjai IB028 EET eloadastarto 1 1. felev	Bevezetes a szamitasel E1B SZIT eloadastarto 1 1. felev			
Bevezeto fizika eloada IB028 FIZ eloadastarto 1 1. felev	Bevezeto matematika el IB028 MAT eloadastarto 1 1. felev			

```
|                        |                        |                        |                        |                        |
|                        |                        |                        |                        |                        |
+------------------------+------------------------+------------------------+------------------------+------------------------+
|                        |                        |                        |                        |                        |
|  Mernok leszek eloadas |   Analizis 1 eloadas   |                        |                        |                        |
|          E1B           |          Q-I           |                        |                        |                        |
|   BTK eloadastarto 1   |   MAT eloadastarto 1   |                        |                        |                        |
|        1. felev        |        1. felev        |                        |                        |                        |
|                        |                        |                        |                        |                        |
|                        |                        |                        |                        |                        |
+------------------------+------------------------+------------------------+------------------------+------------------------+
|                        |                        |                        |                        |                        |
| Bevezeto matematika la | Digitalis technika elo |                        |                        |                        |
|          R4A           |          E1B           |                        |                        |                        |
|    MAT labortarto 1    |   MIT eloadastarto 1   |                        |                        |                        |
|        1. felev        |        1. felev        |                        |                        |                        |
|                        |                        |                        |                        |                        |
| Digitalis technika gya |                        |                        |                        |                        |
|         QBF08          |                        |                        |                        |                        |
| MIT gyakorlattarto 1   |                        |                        |                        |                        |
|        1. felev        |                        |                        |                        |                        |
|                        |                        |                        |                        |                        |
|     Fizika 1 labor     |                        |                        |                        |                        |
|          R4C           |                        |                        |                        |                        |
|    FIZ labortarto 1    |                        |                        |                        |                        |
|        1. felev        |                        |                        |                        |                        |
|                        |                        |                        |                        |                        |
| Mernok leszek gyakorla |                        |                        |                        |                        |
|         QBF09          |                        |                        |                        |                        |
| BTK gyakorlattarto 1   |                        |                        |                        |                        |
|        1. felev        |                        |                        |                        |                        |
|                        |                        |                        |                        |                        |
|   Mernok leszek labor  |                        |                        |                        |                        |
|          R4B           |                        |                        |                        |                        |
|    BTK labortarto 2    |                        |                        |                        |                        |
|        1. felev        |                        |                        |                        |                        |
+------------------------+------------------------+------------------------+------------------------+------------------------+
year
id:                 2
modified_timestamp: 2018-12-06 02:57:52.725823
is_deleted:         0
name:               2. felev
+------------------------+------------------------+------------------------+------------------------+------------------------+
|         MONDAY         |        TUESDAY         |       WEDNESDAY        |        THURSDAY        |         FRIDAY         |
+------------------------+------------------------+------------------------+------------------------+------------------------+
|                        |                        |                        |                        |                        |
|                        |                        |                        |    Analizis 2 labor    |                        |
|                        |                        |                        |          R4I           |                        |
|                        |                        |                        |    MAT labortarto 1    |                        |
|                        |                        |                        |        2. felev        |                        |
|                        |                        |                        |                        |                        |
|                        |                        |                        | A programozas alapjai  |                        |
|                        |                        |                        |          R4B           |                        |
|                        |                        |                        |    IIT labortarto 1    |                        |
|                        |                        |                        |        2. felev        |                        |
|                        |                        |                        |                        |                        |
|                        |                        |                        |     Fizika 2 labor     |                        |
|                        |                        |                        |          R4C           |                        |
|                        |                        |                        |    FIZ labortarto 1    |                        |
|                        |                        |                        |        2. felev        |                        |
|                        |                        |                        |                        |                        |
|                        |                        |                        | Rendszermodellezes gya |                        |
|                        |                        |                        |         QBF08          |                        |
|                        |                        |                        |  MIT gyakorlattarto 1  |                        |
|                        |                        |                        |        2. felev        |                        |
|                        |                        |                        |                        |                        |
|                        |                        |                        | Szamitogep architektur |                        |
|                        |                        |                        |          R4A           |                        |
|                        |                        |                        |    HIT labortarto 1    |                        |
|                        |                        |                        |        2. felev        |                        |
+------------------------+------------------------+------------------------+------------------------+------------------------+
|                        |                        |                        |                        |                        |
|                        |                        |                        |  Analizis 2 gyakorlat  |                        |
|                        |                        |                        |         QBF08          |                        |
|                        |                        |                        |  MAT gyakorlattarto 1  |                        |
|                        |                        |                        |        2. felev        |                        |
|                        |                        |                        |                        |                        |
|                        |                        |                        | Rendszermodellezes lab |                        |
|                        |                        |                        |          R4A           |                        |
|                        |                        |                        |    MIT labortarto 1    |                        |
|                        |                        |                        |        2. felev        |                        |
|                        |                        |                        |                        |                        |
+------------------------+------------------------+------------------------+------------------------+------------------------+
|                        |                        |                        |                        |                        |
|                        |                        |   Fizika 2 eloadas     | Bevezetes a szamitasel |                        |
|                        |                        |         IB028          |         IB028          |                        |
|                        |                        |   FIZ eloadastarto 1   |  SZIT eloadastarto 1   |                        |
|                        |                        |       2. felev         |        2. felev        |                        |
|                        |                        |                        |                        |                        |
|                        |                        |                        |                        |                        |
```

		Szamitogep architektur IB028 HIT eloadastarto 1 2. felev	Analizis 2 eloadas Q-I MAT eloadastarto 1 2. felev	
		A programozas alapjai IB028 IIT eloadastarto 1 2. felev	Rendszermodellezes elo IB028 MIT eloadastarto 1 2. felev	
		A programozas alapjai QBF09 IIT gyakorlattarto 1 2. felev Bevezetes a szamitasel QBF12 SZIT gyakorlattarto 1 2. felev Bevezetes a szamitasel R4A SZIT labortarto 1 2. felev Fizika 2 gyakorlat E401 FIZ gyakorlattarto 1 2. felev Szamitogep architektur QBF08 HIT gyakorlattarto 1 2. felev		

```
year
id:                 3
modified_timestamp: 2018-12-06 02:57:52.725823
is_deleted:         0
name:               3. felev
```

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
		Kodolastechnika labor R4A HIT labortarto 2 3. felev Kommunikacios halozato QBF09 HIT gyakorlattarto 1 3. felev Kommunikacios halozato R4I HIT labortarto 1 3. felev Szoftvertechnologia gy E402 IIT gyakorlattarto 2 3. felev	Rendszerelmelet eloada IB028 HVT eloadastarto 1 3. felev	
		A programozas alapjai R4I IIT labortarto 1 3. felev Adatbazisok gyakorlat QBF12 TMIT gyakorlattarto 2 3. felev Rendszerelmelet gyakor QBF09 HVT gyakorlattarto 1 3. felev Szoftvertechnologia la R4A		

		IIT labortarto 3 3. felev Valoszinusegszamitas l R4C SZIT labortarto 1 3. felev		
		A programozas alapjai QBF09 IIT gyakorlattarto 1 3. felev Adatbazisok labor R4B TMIT labortarto 1 3. felev Kodolastechnika gyakor QBF12 HIT gyakorlattarto 1 3. felev Rendszerelmelet labor R4A HVT labortarto 1 3. felev Valoszinusegszamitas g QBF08 SZIT gyakorlattarto 1 3. felev		
	Szoftvertechnologia el Q-I IIT eloadastarto 1 3. felev	Adatbazisok eloadas E1B TMIT eloadastarto 1 3. felev		
	Valoszinusegszamitas e E1B SZIT eloadastarto 1 3. felev	Kodolastechnika eloada E1B HIT eloadastarto 1 3. felev		
	A programozas alapjai Q-I IIT eloadastarto 1 3. felev	Kommunikacios halozato Q-I HIT eloadastarto 1 3. felev		

year
id: 4
modified_timestamp: 2018-12-06 02:57:52.725823
is_deleted: 0
name: 4. felev

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
	Szoftvertechnikak eloa E1B AUT eloadastarto 1 4. felev	Algoritmuselmelet labo R4B SZIT labortarto 2 4. felev Operacios rendszerek l R4C MIT labortarto 2 4. felev Szoftver projekt labor R4J IIT labortarto 2 4. felev Szoftvertechnikak gyak E401 AUT gyakorlattarto 1		

		4. felev			
	Szamitogepes grafika e E1B IIT eloadastarto 1 4. felev	Algoritmuselmelet eloa E1B SZIT eloadastarto 1 4. felev			
	Menedzsment es vallalk IB027 BTK eloadastarto 1 4. felev	Szoftver projekt labor Q-I IIT eloadastarto 1 4. felev			
	Kommunikacios halozato IB027 TMIT eloadastarto 1 4. felev	Operacios rendszerek e Q-I MIT eloadastarto 1 4. felev			
	Algoritmuselmelet gyak QBF08 SZIT gyakorlattarto 1 4. felev Kommunikacios halozato R4A TMIT labortarto 1 4. felev Menedzsment es vallalk E401 BTK gyakorlattarto 1 4. felev Operacios rendszerek g QBF09 MIT gyakorlattarto 2 4. felev Szamitogepes grafika g QBF12 IIT gyakorlattarto 1 4. felev				
	Kommunikacios halozato QBF08 TMIT gyakorlattarto 1 4. felev Menedzsment es vallalk R4C BTK labortarto 1 4. felev Szamitogepes grafika l R4B IIT labortarto 1 4. felev Szoftver projekt labor QBF12 IIT gyakorlattarto 1 4. felev Szoftvertechnikak labo R4I AUT labortarto 1 4. felev				

year

```
id:                5
modified_timestamp: 2018-12-06 02:57:52.725823
is_deleted:        0
name:              5. felev
```

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
	IT eszkozok technologi IB028 EET eloadastarto 1 5. felev	Mikro- es makrookonomi IB028 BTK eloadastarto 1 5. felev		
	Mesterseges intelligen QBF08 MIT gyakorlattarto 1 5. felev Mobil- es webes szoftv R4C AUT labortarto 1 5. felev Uzleti jog labor R4B BTK labortarto 1 5. felev Informatikai rendszert R4I MIT labortarto 1 5. felev	Informatikai rendszert IB028 MIT eloadastarto 1 5. felev		
	IT eszkozok technologi E401 EET gyakorlattarto 1 5. felev Mikro- es makrookonomi QBF12 BTK gyakorlattarto 1 5. felev Mikro- es makrookonomi R4C BTK labortarto 2 5. felev Informatikai rendszert QBF08 MIT gyakorlattarto 1 5. felev Ipari informatika labo R4A IIT labortarto 1 5. felev			
	IT eszkozok technologi R4A EET labortarto 1 5. felev Mesterseges intelligen R4I MIT labortarto 1 5. felev Mobil- es webes szoftv QBF08 AUT gyakorlattarto 1 5. felev Uzleti jog gyakorlat E401 BTK gyakorlattarto 1 5. felev Ipari informatika gyak QBF12 IIT gyakorlattarto 1 5. felev			
Ipari informatika eloa Q-I IIT eloadastarto 1 5. felev	Mesterseges intelligen IB028 MIT eloadastarto 1 5. felev			

```
|   Uzleti jog eloadas    | Mobil- es webes szoftv |                        |                        |                        |                        |
|         IB028           |         IB028          |                        |                        |                        |                        |
|   BTK eloadastarto 1    |   AUT eloadastarto 1   |                        |                        |                        |                        |
|        5. felev         |        5. felev        |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |                        |
+-------------------------+------------------------+------------------------+------------------------+------------------------+
year
id:                 6
modified_timestamp: 2018-12-06 02:57:52.725823
is_deleted:         0
name:               6. felev
+-------------------------+------------------------+------------------------+------------------------+------------------------+
|         MONDAY          |        TUESDAY         |       WEDNESDAY        |        THURSDAY        |         FRIDAY         |
+-------------------------+------------------------+------------------------+------------------------+------------------------+
|                         |                        |                        |                        |                        |
| Informacios rendszerek  | Intelligens elosztott  |                        |                        |                        |
|         IB028           |          Q-I           |                        |                        |                        |
|   TMIT eloadastarto 1   |   MIT eloadastarto 1   |                        |                        |                        |
|        6. felev         |        6. felev        |                        |                        |                        |
|                         |                        |                        |                        |                        |
+-------------------------+------------------------+------------------------+------------------------+------------------------+
|                         |                        |                        |                        |                        |
| Alkalmazasfejlesztesi   |                        |                        |                        |                        |
|         IB028           |                        |                        |                        |                        |
|   AUT eloadastarto 1    |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |
+-------------------------+------------------------+------------------------+------------------------+------------------------+
|                         |                        |                        |                        |                        |
|   IT biztonsag labor    |                        |                        |                        |                        |
|          R4A            |                        |                        |                        |                        |
|    HIT labortarto 1     |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |
| Intelligens elosztott   |                        |                        |                        |                        |
|         QBF09           |                        |                        |                        |                        |
|  MIT gyakorlattarto 1   |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |
| Intelligens elosztott   |                        |                        |                        |                        |
|          R4C            |                        |                        |                        |                        |
|    MIT labortarto 2     |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |
| Rendszertervezes labor  |                        |                        |                        |                        |
|         QBF08           |                        |                        |                        |                        |
|  MIT gyakorlattarto 2   |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |
| Rendszertervezes labor  |                        |                        |                        |                        |
|          R4B            |                        |                        |                        |                        |
|    MIT labortarto 1     |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
+-------------------------+------------------------+------------------------+------------------------+------------------------+
|                         |                        |                        |                        |                        |
| Informacios rendszerek  |                        |                        |                        |                        |
|         QBF08           |                        |                        |                        |                        |
| TMIT gyakorlattarto 1   |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |
| Informacios rendszerek  |                        |                        |                        |                        |
|          R4A            |                        |                        |                        |                        |
|   TMIT labortarto 1     |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |
| IT biztonsag gyakorlat  |                        |                        |                        |                        |
|          E401           |                        |                        |                        |                        |
|  HIT gyakorlattarto 1   |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |
| Alkalmazasfejlesztesi   |                        |                        |                        |                        |
|         QBF09           |                        |                        |                        |                        |
|  AUT gyakorlattarto 1   |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
|                         |                        |                        |                        |                        |
| Alkalmazasfejlesztesi   |                        |                        |                        |                        |
|          R4B            |                        |                        |                        |                        |
|    AUT labortarto 1     |                        |                        |                        |                        |
|        6. felev         |                        |                        |                        |                        |
+-------------------------+------------------------+------------------------+------------------------+------------------------+
|                         |                        |                        |                        |                        |
| Rendszertervezes labor  |                        |                        |                        |                        |
```

IB028 MIT eloadastarto 1 6. felev				
IT biztonsag eloadas E1B HIT eloadastarto 1 6. felev				

year
id: 7
modified_timestamp: 2018-12-06 02:57:52.725823
is_deleted: 0
name: 7. felev

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
			Deklarativ programozas IB028 SZIT eloadastarto 1	

			7. felev		
			Deklarativ programozas QBF09 SZIT gyakorlattarto 1 7. felev Deklarativ programozas R4A SZIT labortarto 1 7. felev Rendszertervezes labor QBF08 MIT gyakorlattarto 1 7. felev Rendszertervezes labor R4B MIT labortarto 1 7. felev		
			Rendszertervezes labor IB028 MIT eloadastarto 1 7. felev		