

User Manual for Southwestern University Course Scheduler Service

Computer Science Capstone '24

Yahya Hamdallah, Mark Mueller, Kate Nguyen, Colby Sullivan

Instructor: Dr. Barbara Anthony

Overview

The Southwestern University Course Scheduler offers faculty members with a useful tool that includes various features to complete course scheduling easily. This documentation will cover what the course scheduler is, important components of it, and how to utilize it.

What is it?

Course scheduling is a laborious and complicated task for faculty members every year. The inconsistency with compatible teaching times, availability of professors, and years a course is offered introduces more complexity into this undertaking. It would be beneficial to have a user-friendly, web-based program to complete this task for them. The Southwestern University Course Scheduler is the solution to this problem.

This Python tool takes in CSV files containing course and faculty restrictions defined by the users within the application, implements them to generate a linear program in PyGLPK that will schedule the courses.

This application includes intuitive interfaces to create CSV files, run the software tool to generate the linear program, and display the generated course schedule. In addition, it will include some other helpful features such as importing existing CSV files or altering already generated schedules to check for feasibility.

Linear Programming

What makes course scheduling so complex is adhering to all of the restrictions and requests made by faculty members while also producing the most efficient schedule. This is where the linear programming comes in.

Linear programming is a mathematical modeling method to create the best outcome where the constraints and the objective are represented by linear relationships. The constraints that the schedule must follow are the course and faculty restrictions that will be specified by the users. These include, but are not limited to, a faculty member's teaching time availability, the classes being taught, and when they are offered. In the context of our application, it is difficult to quantify a 'best' schedule, so our tool focuses on simply creating a schedule that adheres to these constraints.

The constraints will be defined by the user in the **Create CSV** page that will be explained more in depth in a later section. The application will implement a linear program with the objective and the constraints, and then using PyGLK, a software package that solves linear programs, will output the course schedule for the year.

CSV Files

Comma-separated values (CSV) is a text file format that stores tabular data in regular text, using commas to separate values. Each line in the file represents one data record in the table. These files are a large component of the Southwestern Course Scheduler application, in that they are used to define the course and faculty restrictions included in the linear program to generate the course schedule. The files are difficult to interpret, as they are meant for the script to parse, not to be understandable by a user. Because of this, the site offers a page for users to create these files.

From the CSV creation page, the Course Scheduler implements a Python script that takes in these CSV files and generates a linear program that is then run through PyGLPK to schedule the courses.

Alternatively, users can choose a previously created CSV and import it into the application later. Additionally, CSV files can be created outside of the application and imported, as long as it adheres to the formatting.

Database

While it is not a significant aspect of our application, our program uses a Google Firebase database to store essential information about courses. It is used in the Create CSV page. When users enter a course into the course table, its name, abbreviation, the number of sections, and its course ID are added into the database. This allows the information to be saved and easily retrievable for future use. It allows the useful feature of autofill in the course table.

Python Script

The Scheduler operates off of a Python script running Python GNU Linear Programming Kit (PyGLPK). The script is fed in 1-2 CSV files that contain all of the course and faculty constraints. A linear program is created based off of these constraints modeled as a 2D matrix. This linear program is then passed into PyGLPK, where it is solved, checked for feasibility, and then exported to a separate CSV file. This CSV file is passed back to the website for proper displaying.

Ongoing Work

While the core feature of viewing a generated schedule is functional, the ability for faculty to edit the schedules is in the works. This will allow faculty members to swap two courses, or to change a course's time and then check for feasibility. Reporting on what constraint causes a schedule to be infeasible is also desirable, since it would show what fixes would need to be made. The database currently contains only math and computer science courses; when provided appropriate constraint information, the program can be more widely used.

How to Use

The goal of the Southwestern University Course Scheduler was to make the course scheduling process as simple as possible. Therefore, the steps to accomplish this task are quite minimal. The requirements of the user include creating the CSV files and uploading them in the Python tool. The application will run PyGLPK and the resulting course schedule will be outputted for the user to download. Follow the steps below to do this.

1. From the Home page, click the **Get Started** button. This will take you to a page that gives two options: **Create A New CSV** or **Import A CSV**.
 - a. The import option is to be chosen if the user has an existing CSV file they would like to make alterations to.
 - i. If so, click **Import A CSV**
 - ii. Choose the CSV file to upload and click the **Upload** button and the outputted schedule will be displayed

Import A CSV

The Course Scheduler allows users to upload existing CSV files to create schedules. To do so, upload one below.

Choose File No file chosen

- b. If not, click **Create A New CSV**.
2. You have now been navigated to the Create CSV page. This is where all course and faculty restrictions will be defined.
 - a. Add all courses

- i. For each course, enter its name, abbreviation, specify if it is a four contact hour course or not, and the number of sections this course has. Check every time that the course is unable to be taught and then specify the course's ID.
 - ii. If there are other courses that conflict with this one, click the **Add Conflicting Course** and write the name of that course. This can be done multiple times.
- b. Add all faculty
- i. For each faculty member, enter their name, indicate whether they need to teach in the prime time or no, and check every time that the professor is unable to teach.
 - ii. Then specify another course the professor is teaching. If there are other courses being taught, click the **Add Course Taught** button and write the name of that course. This can be done multiple times.

Course Table

-	Class name	Abbreviation	4 Contact Hours	Sections	Unavailable Times	Select CourseID
-	<input type="text" value="Enter New Course"/>	<input type="text" value="Enter Course Abbreviation"/>	No	<input type="text" value="Number of Sections"/>	MWF <input type="checkbox"/> 8:00 <input type="checkbox"/> 9:30 <input type="checkbox"/> 11:00 <input type="checkbox"/> 2:00 <input type="checkbox"/> 3:30 TTh <input type="checkbox"/> 8:30 <input type="checkbox"/> 10:00 <input type="checkbox"/> 11:30 <input type="checkbox"/> 1:00 <input type="checkbox"/> 2:30	<input type="text" value="Enter Course ID"/>

Faculty Table

-	Professor Name	Prime time	Classes	Unavailable Times
-	<input type="text" value="Enter Faculty Name"/>	No	<input type="text" value="Course Abbreviation Taught"/>	MWF <input type="checkbox"/> 8:00 <input type="checkbox"/> 9:30 <input type="checkbox"/> 11:00 <input type="checkbox"/> 2:00 <input type="checkbox"/> 3:30 TTh <input type="checkbox"/> 8:30 <input type="checkbox"/> 10:00 <input type="checkbox"/> 11:30 <input type="checkbox"/> 1:00 <input type="checkbox"/> 2:30

3. Once these tables are filled out, click the **Save as CSV** button in the top right corner. This will download the CSV files to your machine.

4. Now that the CSV files have been created, they can be uploaded to the application to generate the schedule. Click the **Next** button. This will navigate to the import page. Import your recently downloaded csv file

5. Click **Upload** and the schedule will be generated

Additional Features

Now that the course schedule has been generated for users, our application contains some additional features to make changes to this schedule if necessary or desired. These include forcing a course to be at a specific time and the ability to swap two courses.

We know there are some courses that can only be taught at a specific day and time. To accommodate this, we implemented a function that lets users force a course to be at a time specified by them. It will take this restraint and prioritize it above the others to guarantee the course is assigned to that time. To do this, follow the steps below:


1. Navigate to the page displaying the generated schedule
2. At the bottom of the page, select the desired course to force
3. Select the day/time to be taught and click **Submit**

Courses Schedule				
Time	Course Col 1	Faculty	Course Col 2	Faculty
MWF at 8:00 AM	CS11	Prof_2	\$	
MWF at 9:30 AM	DIS1	Prof_3	\$	
MWF at 11:00 AM	ORG1	Prof_5	\$	
MWF at 2:00 PM	CS21	Prof_5	\$	
MWF at 3:30 PM	EXP1	Prof_1	\$	
TTH at 8:30 AM	ARI1	Prof_5	\$	
TTH at 10:00 AM	PRL1	Prof_1	\$	
TTH at 11:30 AM	ALG1	Prof_1	\$	
TTH at 1:00 PM	OPR1	Prof_1	\$	
TTH at 2:30 PM	TOC1	Prof_2	\$	
MWF at 8:00 AM	DAT1	Prof_5	\$	
MWF at 9:30 AM	SYS1	Prof_5	\$	
MWF at 11:00 AM	CAP1	Prof_1	\$	
MWF at 2:00 PM	NUA1	Prof_4	\$	

Choose a course to force:

When will it be taught:

Swap with



Because our application is deterministic, the schedule generated will be the same if the input variables are the same. Because of this, we understand there is a possibility that courses could be assigned to more preferred times than the one initially designated. To allow for these changes to occur, we implemented a function that lets users switch the position two courses are assigned. To do this, follow the steps below:

1. Navigate to the page displaying the generated schedule
2. At the bottom of the page, select the two courses to be swapped
3. Click **Submit**

Courses Schedule				
Time	Course Col 1	Faculty	Course Col 2	Faculty
MWF at 8:00 AM	CS11	Prof_2	\$	
MWF at 9:30 AM	DIS1	Prof_3	\$	
MWF at 11:00 AM	ORG1	Prof_5	\$	
MWF at 2:00 PM	CS21	Prof_5	\$	
MWF at 3:30 PM	EXP1	Prof_1	\$	
TTH at 8:30 AM	ARI1	Prof_5	\$	
TTH at 10:00 AM	PRL1	Prof_1	\$	
TTH at 11:30 AM	ALG1	Prof_1	\$	
TTH at 1:00 PM	OPR1	Prof_1	\$	
TTH at 2:30 PM	TOC1	Prof_2	\$	
MWF at 8:00 AM	DAT1	Prof_5	\$	
MWF at 9:30 AM	SYS1	Prof_5	\$	
MWF at 11:00 AM	CAP1	Prof_1	\$	
MWF at 2:00 PM	NUA1	Prof_4	\$	

Choose a course to force:

When will it be taught:

Swap with

