



UNIVERSITY OF ST.GALLEN

School of Management, Economics,
Law, Social Sciences, International Affairs and Computer Science

Documentation

Group Project: LawDays Assignment Tool

Group ID: 3417

Project language: Python

Module:

Skills: Programming - Introduction Level

By Mario Silic, PhD

Date of Submission:

26/May/2025

Background of Group Project

As part of my involvement in a student association at the University of St. Gallen, I was responsible for organizing an event called *LawDays* (for more information about the event see <https://www.lawdays.ch>). This career event is specifically designed for law students and features a three-round speed-networking format, where participants have the opportunity to meet various legal employers in small groups for 15-minute sessions.

To enable this, each student must be assigned to three companies. During registration, students are asked to indicate their preferences for the companies they would like to meet.

Until now, the assignment of students to companies was carried out manually using Excel. However, the feedback from last year revealed dissatisfaction with the process, as many students did not receive matches aligned with their stated preferences.

This initiated me to explore a more efficient and fair solution by developing a program in Python to optimize the assignment process based on the students' preferences.

Program Description

This Python program was developed to address the challenge of fairly and efficiently assigning students to companies during the *LawDays* career event. The program was not developed solely for this year's event, but was intentionally designed to be reusable for future editions of *LawDays* or similar events. Its structure is dynamic, meaning that the number of participating students and companies can vary without requiring changes to the code. As long as the input Excel file is formatted correctly, the program automatically adapts to the size of the dataset, making it a flexible and scalable solution for future use.

The goal of the program is to generate an optimized schedule that maximizes the overall satisfaction of the students while meeting the logistical requirements of the event. Specifically, the program ensures that each student is assigned to exactly one company per round, that companies are visited by a reasonable number of students (between three and seven per round), and that no student meets the same company more than once. The students' preferences are incorporated into the algorithm in the form of a weighted scoring system. First-choice companies are awarded five points, second-choice companies four points, and so on, down to one point for fifth-choice companies. Companies that were not ranked receive zero points. The optimization algorithm then seeks to maximize the total sum of these preference points across all student-company-round assignments.

The assignment logic is implemented using a Mixed-Integer Linear Programming (MILP) model, which is solved using the open-source CBC solver provided by Google OR-Tools. The program reads data from an Excel file (*event_input.xlsx*) that contains two worksheets: the first worksheet includes a list of participating companies, while the second contains the names of students and their company priorities.

After solving the optimization problem, the program outputs the results to an automatically generated Excel file named *event_assignment_result.xlsx*. This file contains two sheets: one displaying the individual schedules for each student, and another summarizing the number of students assigned to each company in each round. The student schedule is visually enhanced using color-coding based on how highly the assigned company was ranked by the respective student. For example, a green background indicates a first-choice assignment, while red denotes a fifth-choice, and white indicates that the student did not rank the assigned company at all.

Program execution instructions

To run the program, the user must first prepare an input Excel file (*event_input.xlsx*) and place it in the same folder as the Python script. The first sheet should contain the names of the companies in a single column, while the second sheet should list student names in the first column and their preferences in the subsequent columns, labeled "priority 1" through "priority 5". For the purpose of this project, the input file has already been prepared, with the anonymized data of this year's participant. It can be found on GitHub.

Next, the user must ensure that the required Python packages are installed. These include "pandas", "numpy", "ortools", and "xlsxwriter", which can be installed using the following command:

```
pip install pandas numpy ortools xlsxwriter
```

Once the dependencies are installed and the input file is correctly prepared, the program can be executed by running the Python script within the respective program (e.g. Visual Studio Code, Replit or any interface alike).

After execution, the final Excel output (*event_assignment_result.xlsx*) will be created in the same directory. This file provides a complete and optimized assignment plan.

Summary

In conclusion, this program offers a reliable and scalable solution to a complex allocation problem that previously relied on error-prone manual processing in Excel. By incorporating optimization techniques and user-friendly output formatting, it significantly improves the fairness and transparency of the assignment process for all participants.

Acknowledgment of the use of ChatGPT

While developing this entire project, ChatGPT was utilized as a tool to assist with specific aspects of the development. ChatGPT was especially used in the coding process itself as well as bug fixing and some parts of the documentation. The collaboration with ChatGPT allowed for a more efficient process while ensuring a clear understanding of all components of the application.