Sistemi Intelligenti

Academic Timetable Optimization Using Constraint Satisfaction Problem

INGEGNERIA DELLE TECNOLOGIE PER L'IMPRESA DIGITALE

Lucia Blazquez Cintas

1. Introduction

As an Erasmus student enrolled in multiple courses across different faculties and time slots, it becomes essential to find an efficient solution to the academic scheduling problem. This project aims to design an intelligent system capable of generating an optimal schedule from the student's perspective, maximizing class attendance while minimizing overlaps and building transitions.

The proposed approach is based on solving a Constraint Satisfaction Problem, complemented by heuristic evaluation techniques and pathfinding over a graph that represents the university campus. The system models a realistic and practical situation where class times are fixed and the student must adapt to them in the most efficient way possible.

2. Project Objective

The main objective is to develop a system that automatically generates a personalized academic timetable that satisfies all time conflict and mobility constraints, while optimizing the student's experience by:

- Maximizing attendance,
- Reducing building transitions,
- Minimizing dead time between classes.

3. System Requirements

Functional Requirements:

- The system must allow input of subjects with multiple weekly time slots.
- There must be no overlap between selected classes.
- The system must maximize the number of classes the student can attend, within defined limits.
- A minimum transition time between consecutive classes in different buildings must be respected.
- The system should generate multiple valid combinations and rank them based on quality.

Non-Functional Requirements:

- The system must be extensible and easy to modify to add new constraints or parameters.
- It must be understandable from an academic point of view, supported by formal AI models.
- Execution must be fast and the code should be well-organized for ease of evaluation and analysis.

4. Modeling the Problem as a CSP

The core of the system is based on a Constraint Satisfaction Problem model, where:
- Variables are the student's subjects.
- Domains are the possible combinations of time slots per subject (if more than one is available).
- Constraints are rules preventing time overlaps and physically infeasible transitions between consecutive classes.

For subjects with two weekly sessions, both are selected if they do not conflict with other classes. This simulates a more realistic model in which the student wishes to attend all available sessions.
Constraints are implemented as binary functions between pairs of variables, evaluating both direct overlaps and whether a physical transition between class locations is possible within the time gap.

5. Optimization and Pathfinding

To represent the campus layout, a graph is constructed where nodes are buildings and edges indicate estimated walking time between them.
When two classes are scheduled consecutively in different buildings, the system uses a shortest path algorithm to check if the transition is feasible. If the required time exceeds the available gap, that schedule is discarded.

Once valid solutions are generated, a heuristic scoring function ranks them according to:
- Total dead time: sum of idle gaps between classes on the same day.
- Number of building changes: estimates logistic effort.
- Number of days with classes: compact schedules are preferred.
- Friday off: included as a bonus objective.
- Total classes attended: highest priority metric.

This hybrid approach not only filters valid combinations but also highlights those offering a better student experience.

6. Project Structure

The codebase is modular and organized into the following components:
1. Parsers and validators: interpret time slots and detect conflicts.
2. CSP definition: declares variables, domains, and cross-subject constraints.
3. Pathfinding engine: evaluates physical feasibility of transitions between sessions.
4. Heuristic evaluator: scores each solution using the criteria above.
5. Result presenter: displays the top-ranked timetables with statistics.

The modular structure allows for easy modifications of both CSP rules and evaluation priorities, making it adaptable to various student profiles and institutional scenarios.