Examination Timetabling Problem (ETP) Project Report

1. Introduction

Timetabling problems are essential and complex tasks that occur periodically in various domains, including academic institutions. The objective is to assign time slots and resources to meetings or exams while satisfying a set of constraints. This report focuses on the Examination Timetabling Problem (ETP), where the aim is to schedule exams in an examination period such that certain constraints are met, and the overall penalty is minimized.

2. Problem Description

The ETP involves a set of exams (E) and a set of students (S) who are enrolled in subsets of these exams. The examination period is divided into ordered time slots (T). The problem's objective is to assign each exam to a time slot while adhering to the following constraints:

- Each exam is scheduled exactly once.
- Conflicting exams (exams with at least one common student) cannot be scheduled in the same time slot.
- The total penalty for scheduling conflicting exams within five time slots of each other is minimized.

Penalties are calculated based on the number of students enrolled in both exams and the distance between their scheduled time slots.

3. Instances

The project includes solving benchmark instances to validate the model and solution. Each instance is defined by:

- `instanceXX.exm`: Number of students enrolled per exam.
- `instanceXX.slo`: Number of available time slots.
- `instanceXX.stu`: Exams in which each student is enrolled.

4. Model Formulation

4.1 Basic ETP Formulation

The problem is formulated as an Integer Linear Programming (ILP) model:

- **Variables**: Binary variables are used to indicate if an exam is scheduled in a particular time slot.
- **Constraints**:

    - Each exam is scheduled exactly once.
    - No conflicting exams are scheduled in the same time slot.
    - Additional constraints for equity measures and restrictions.
- **Objective**: Minimize the total penalty for scheduling conflicting exams within close time slots.

4.2 Equity Measures

In addition to minimizing penalties, the model includes equity measures:

- **Maximum Distance Between Conflicting Exams**: Ensure conflicting exams are scheduled far apart to provide students more time between exams.
- **Number of Students with Back-to-Back Exams**: Minimize the number of students who have consecutive exams.

4.3 Additional Restrictions

The model incorporates additional restrictions:

- At most three consecutive time slots can have conflicting exams.
- If two consecutive time slots have conflicting exams, no conflicting exam can be scheduled in the next three time slots.
- A bonus is included for periods where no conflicting exams are scheduled for six consecutive time slots.
- At most three conflicting pairs can be scheduled in the same time slot.

5. Implementation

The model is implemented in Python using the Gurobi optimizer. The code reads input files, builds the conflict graph, formulates the ILP model, and solves it within a specified time limit.

5.1 Reading Input Files

The input files are read to obtain the number of students enrolled per exam, the number of available time slots, and the exams in which each student is enrolled.

5.2 Building the Conflict Graph

The conflict graph is built by identifying pairs of exams that have at least one common student. This helps in formulating constraints to ensure conflicting exams are not scheduled in the same time slot.

5.3 Formulating the ILP Model

The ILP model is formulated by:

- Defining binary variables for each exam and time slot combination.
- Adding constraints to ensure each exam is scheduled exactly once.
- Adding constraints to ensure conflicting exams are not scheduled in the same time slot.
- Incorporating equity measures and additional restrictions into the constraints and objective function.

5.4 Solving the Model

The Gurobi optimizer is used to solve the model within a specified time limit. The objective function is to minimize the total penalty for close conflicts while considering equity measures.

6. Solution Validation and Visualization

6.1 Validation

The solution is validated by checking that:

- No conflicting exams are scheduled in the same time slot.
- All additional constraints and equity measures are satisfied.

6.2 Visualization

The timetable is visualized to provide a clear representation of the exam schedule across the available time slots. This helps in verifying the feasibility and quality of the solution.

7. Main Function and Execution

The main function orchestrates the entire process:

- It reads the input files.
- Builds the conflict graph.
- Formulates and solves the ILP model.
- Validates the solution.
- Visualizes the timetable.
- Writes the output to a file.

8. Results and Analysis

After running the model on the benchmark instances, the results are analyzed based on the total penalty, maximum distance between conflicting exams, and the number of students with back-to-back exams. The model provides feasible solutions that adhere to the constraints and additional restrictions, minimizing the overall penalty.

9. Conclusion

The Examination Timetabling Problem (ETP) project successfully formulates and solves the ETP using Integer Linear Programming and the Gurobi optimizer. The model incorporates essential constraints, equity measures, and additional restrictions to create equitable and feasible examination timetables. The solutions are validated, and the results demonstrate the effectiveness of the model in minimizing penalties and improving student schedules.

10. Future Work

Future improvements can include exploring more advanced optimization techniques, incorporating more equity measures, and extending the model to handle more complex real-world constraints and scenarios.