HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

MINI PROJECT

PROJECT 9

TimeTable assign slot and room to classes
Largee

LUU HIEU AN – ID: 202400093

an.lh2400093@sis.hust.edu.vn

Specialization: Data Science and Artificial Intelligence

Hanoi, 00/00/2023

1 Introduction and objectives

There are N classes numbered $1, 2, \ldots, N$ that need to be scheduled in a timetable. Each class i has $t(i)$ as the number of periods, $g(i)$ as the teacher assigned to teach that class, and $s(i)$ as the number of students in the class. There are M classrooms numbered $1, 2, \ldots, M$, where $c(i)$ is the seating capacity of room i. The week consists of 5 days (from Monday to Friday), each day divided into 12 periods (6 periods in the morning and 6 periods in the afternoon). The periods throughout the week are numbered consecutively from 1 to 60.

Requirements

- Create a timetable by assigning each class a day, period, and classroom.
- Classes taught by the same teacher must be scheduled at different times.
- The number of students in each class must be less than or equal to the seating capacity of the assigned classroom.
- The number of classes scheduled must be maximized.

The data format

- Line 1: Two integers N and M ($1 \leq N \leq 1000$, $1 \leq M \leq 100$).
- Next N lines: Each line contains three integers $t(i)$, $g(i)$, and $s(i)$ ($1 \leq t(i) \leq 4$, $1 \leq g(i) \leq 100$, $1 \leq s(i) \leq 200$).
- Line $N+2$: M integers $c(1), c(2), \ldots, c(M)$ representing seating capacities ($1 \leq c(i) \leq 300$).

Output

- Line 1: A positive integer Q, the number of classes scheduled.
- Next Q lines: Each line contains three positive integers i, u, and v, where class i is assigned to period u and room v.

Generated data
The input size is denoted by $(number\_of\_classes * number\_of\_rooms)$

- 1 sample tests : (10 x 2).
- 50 Small tests: $N * M$ $(1 \leq N \leq 50,\ 1 \leq M \leq 10)$
- 10 Medium tests: $N * M$ $(51 \leq N \leq 200,\ 11 \leq M \leq 30)$
- 10 Large tests: $N * M$ $(201 \leq N \leq 600,\ 31 \leq M \leq 60)$
- 5 Huge tests: $N * M$ $(601 \leq N \leq 1000,\ 61 \leq M \leq 100)$

2 Modeling the problem

2.1 CP Model

Model Variables

- **Input Data:**
  - N: number of classes
  - M: number of rooms
  - t_i: duration (in time slots) of class i, where $1 \leq i \leq N$
  - g_i: teacher ID of class i
  - s_i: number of students in class i
  - c_i: capacity of room i, where $1 \leq i \leq M$
- **Decision Variables:**
  - $x_i \in \{0, 1\}$:
    * $x_i = 1$ if class i is scheduled
    * $x_i = 0$ otherwise
  - $room_i \in \{-1, 0, 1, \ldots, M-1\}$: the room assigned to class i.
    If not scheduled, $room_i = -1$.
  - $start_i \in \{-1, 0, 1, \ldots, 59\}$: starting time slot of class i.
    If not scheduled, $start_i = -1$.
  - $assigned_{i,r} \in \{0, 1\}$:
    * $= 1$ if class i is scheduled in room r
    * $= 0$ otherwise
  - $end_i = start_i + t_i$: ending time slot of class i
  - $interval_{i,r}$: optional interval variable representing class i occupying room r
- **Objective Function:**

  $$\max \sum_{i=1}^{N} x_i$$

  Maximize the number of scheduled classes.
- **Constraints**
  - **(C1) Room and Time Validity:** A class can only be assigned a valid room and start time if it is scheduled:

    $$x_i = 1 \Rightarrow 0 \leq room_i \leq M - 1$$

    $$x_i = 0 \Rightarrow room_i = -1, start_i = -1$$
  - **(C2) Room Capacity Constraint:** A class can only be scheduled in a room if the room has sufficient capacity:

    $$assigned_{i,r} = 1 \Rightarrow s_i \leq c_r$$

– **(C3) No Teacher Conflict:** No teacher may teach two classes at overlapping time slots:

$$g_i = g_j \wedge x_i = x_j = 1 \Rightarrow (start_i + t_i \leq start_j \quad or \quad start_j + t_j \leq start_i)$$

– **(C4) No Room Overlap:** No two classes assigned to the same room can overlap in time:

$$\texttt{AddNoOverlap}(\{interval_{i,r} \mid assigned_{i,r} = 1\})$$

– **(C5) Interval Activation:** The interval variable is only active if class i is assigned to room r:

$$interval_{i,r} isactive \iff assigned_{i,r} = 1$$

2.2 Heuristic

Problem Statement

In the context of higher education, constructing class schedules is a complex task due to numerous constraints, such as the number of classes, teachers, students, limited number of classrooms, and fixed time slots. The goal of the problem is to allocate classes into time slots and classrooms so as to:

* Maximize the number of scheduled classes.
* Ensure no teacher is assigned to multiple classes at the same time.
* Ensure each classroom is used for only one class at any given time.
* Ensure the classroom can accommodate the number of students in the class.

This problem belongs to the class of combinatorial optimization problems and is NP-Hard, meaning exact methods are often infeasible for large-scale instances. In this mini-project, I apply a heuristic algorithm of the Greedy type to generate a near-optimal solution for class scheduling.

Heuristic Approach Used

The algorithm used is a Greedy algorithm. The main idea is:

* Sort the list of classes based on a certain priority criterion (e.g., descending order of student count).
* Iterate through each class in that order and attempt to assign the class to the first feasible time slot and classroom that satisfies all constraints.
* If no suitable slot is found, skip that class.

The constraints checked include:

* The teacher is not already scheduled for another class at the same time.
* The classroom is not already in use at that time.
* The classroom has enough capacity for the class.

This approach is simple, easy to implement, and can yield relatively good results in a short time, especially suitable for real-world problems of moderate size.

Solution Representation

Input includes:

* List of classes (class name, number of students, assigned teacher).
* List of classrooms (room name, capacity).
* Total number of time slots in a week.

Each solution is a mapping from class $\rightarrow$ (room, time slot) that satisfies the constraints.
Specifically:

* A class is assigned to one classroom at one unique time slot.
* A class is only scheduled if the teacher is free, the room is available, and the room has enough space.

Output: A list of scheduled classes along with the assigned room and time slot information.

Results and Analysis

The Greedy algorithm was tested on a simulated dataset including:

* N classes.
* M classrooms with varying capacities.
* 60 time slots (over 5 days, 6 slots per day).

Results:

* Total scheduled classes: a_i.
* Classes skipped due to lack of rooms or teacher conflicts: $N - a_i$.
* Execution time: under 1 second.

Analysis:

* The algorithm delivers fast and feasible results for medium-sized data.
* However, due to its greedy nature, the algorithm may miss globally optimal solutions — e.g., a class might be schedulable if a later slot were chosen.
* Classes with more students are prioritized, which may disadvantage smaller classes.