UNIVERSIDAD EAFIT®

Vigilada Mineducación

Data Structures and Algorithms II
Final Project

Juan G. Lalinde-Pulido
jlalinde@eafit.edu.co

April 2, 2020

Abstract

As students of *ST2047 Data Structures and Algorithms II*, you have been selected to solve a problem at EAFIT University. Every semester, the university carries out the academic scheduling based on the scheduling of the immediately preceding year. Now, because the physical plant of the university has grown a lot, they are asking you to develop a program that starts from the academic scheduling and modifies the assignment of classrooms in a way that minimizes the average distance a person must travel when finishing a class and moving on to the next one.

1 Problem statement

One of the problems that every educational institution must solve is classroom scheduling. The scheduling must meet the following conditions:

1. No teacher has two different classes scheduled simultaneously, that is, at the same time on the same day.
2. No classroom is allowed to have two classes scheduled simultaneously.
3. All groups are assigned a teacher and a classroom.

Ideally, all students should be able to take all the courses they are entitled to. However, when there is more than one group per course, and the institution allows its students to select the group to which they wish to belong, there may be cases

in which the student did not find a group available as a result of decisions made by their peers at the time of enrollment.

This problem has a trivial solution: having all classes at different times in different classrooms, but this is not a real option. For this reason, educational institutions see academic programming as an optimization problem, where the optimal solution is the one that requires the minimum number of resources for its implementation. Now, this problem is an NP-complete problem[1], that is why heuristics are used to get approximate solutions in a reasonable time. Naturally, many soft restrictions are not taken into account in this process[1]. However, some of them are desirable.

Given valid academic programming, other valid solutions can easily be generated. Any permutation in which the type of classroom is not changed and the capacity of the classrooms is appropriate is also a valid solution.

One of the university's concerns is to ensure that teachers and students who have limited mobility can easily access their assigned classrooms. Additionally, it is desirable to minimize distance that teachers and students must walk when moving from one class to another. You, recognized for your great problem-solving skills, have been hired to generate a new academic programming that meets two characteristics:

Hard restriction: It must be ensured that all persons with mobility limitations have their classes scheduled in easily accessible classrooms.

Soft Restriction: The average time it takes for a person to move from one class to another should be minimized.

2 Data Description

Academic Programming: The list of all groups scheduled for the semester. Each group has an associated professor and classroom. It is delivered in a text file where each line corresponds to an hour assigned to a group and has the following fields separated by commas[2]:

- `course`: sequence of six (6) alphanumeric symbols. The first two are letters and represent the academic department responsible for the

[1] A soft restriction is one that is desirable but whose noncompliance does not invalidate the solution. Hard restrictions are those whose noncompliance invalidates the solution-

[2] That format is called CSV, Comma Separated Values

course and four digits which are the course code.

- `group`: sequence of three (3) digits that identifies the group. The combination `course, group` identifies each programmed group.
- `professor`: integer that identifies the teacher
- `classroom`: sequence of alphanumeric symbols that identifies the classroom where the group is programmed
- `day`: an alphabetical symbol representing the day of the week like this:

  L Monday

  M Tuesday

  W Wednesday

  J Thursday

  V Friday

  S Saturday
- `starting time` integer table number digits representing the start time of the class in military format [3].
- `end time` integer number of table digits representing the end time of the class in military format

Enrollment: The list of all students in the university with the courses they have enrolled in. It's a text file with the values separated by commas.

- `student`: integer identifying the student
- `course`: sequence of six (6) alphanumeric symbols. The first two are letters and represent the academic department responsible for the course and four digits which are the course code.
- `group`: sequence of three (3) digits that identifies the programmed group. The combination `course, group` identifies each programmed group.

Classrooms: The list of all the classrooms in the university. It's a text file with the values separated by commas.

[3] Military time is represented as a four digit number where the first two are the hour of the day (00-23) and the last two are the minutes (00-59).

- `classroom`: sequence of alphanumeric symbols that identifies the classroom where the group is programmed
- `type`: integer value representing the type of classroom
- `capacity`: maximum number of students who can attend class simultaneously in the classroom.
- `access`: an integer indicating the type of access to the classroom, like this:

  0 Inclusive access that allows access to any person

  1 Does not have inclusive access, so people with mobility limitations cannot access the classroom.

The map of the university is represented as a graph, where each block is a node. For simplicity, we assume that we can neglect the distance between two classrooms that are in the same block, so only the distance between blocks is provided. Two blocks are connected by an arc if there is a direct path between them that does not pass through any other blocks. This is a very similar representation to the one used by GPS systems. The information is in a text file in a comma-separated format, where each line contains three entries:

- `block_1`: sequence of alphanumeric symbols that identifies the initial block
- `block_2`: sequence of alphanumeric symbols identifying the final block
- `Distance`: A number in floating point format representing the distance between the two blocks

Persons with mobility impairments: A list containing the identifications of all persons who have mobility impairments. Each line has a single element that is an integer and represents the identity of the person who has mobility limitations.

References

[1] CHEN, R.-M., AND SHIH, H.-F. Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search. *Algorithms 6*, 2 (apr 2013), 227–244.