Ho Chi Minh city University of Technology
Faculty of Computer Science and Engineering

ADVANCED ALGORITHMS

Course work

A priority of preferences for teacher assignment problem

Tutors
TRANG HONG SON, HUYNH TUONG NGUYEN

Group ĐTVT
THUY Nguyen Chinh - 2170402
PHUONG Le Van Hoang - 2170403
TUAN Vu Duc - 2170091
THIEN Do Tieu - 2170401
CHAU Do Ngoc Thanh - 1970584

Contents

1 Introduction

The sudden outbreak of a deadly disease called Covid-19 caused by the Corona Virus (SARS-CoV-2) has shaken the whole world. The World Health Organization declared it a pandemic. This situation has challenged education systems worldwide and nationally; forcing educators to switch to many new teaching methods to ensure the health of both teachers and learners. In the past half year, the pandemic has hit Ho Chi Minh City hard. The entire city and neighboring provinces were forced to blockade and each resident had to self-isolate in his/her family. This has affected everyone's work and psychology. In that context, when looking at the field of education, many lecturers have pictures of acting depressed and uncontrollable when communicating with learners. In the scope of this research, the team considers the classroom delivery process of the lecturers and proposes to change the method to partially support the teacher's attitude and work harmony in order to reduce pressure and thereby, increase the enthusiasm for teaching and learners can gain knowledge better in a more relaxed and respectful environment.

According to the credit training regulations, students will have to create their own learning path to complete the training program and must actively register for subjects in each of their semesters. To prepare for the student's course registration, usually at the beginning of each semester, the training department and the head of the department will work together to carry out the following steps: (1) based on the program's reference road map and individual study plan of students in the upcoming semester, the training department counts the number of classes that need to be opened for each subject (referred to as subject classes) and sends a list of these subject classes to the head of the department, (2) the head of the department assigns the permanent faculty members of the department to be in charge of teaching these subject classes, if not enough, additional visiting lecturers must be invited outside the school, (3) based on the assignment proposed by the head of the department, the training department arranges the timetable for these subject classes, (4) students register for the course on this schedule, then the training department and the head of the department will make adjustments (if any) to suit the student's actual course registration situation such as opening more subject classes or canceling subject classes, (5) after the final schedule, instruction for that semester will take place.

Currently, in Step 2, the head of the department assigns full-time lecturers in charge of teaching classes difficultly and emotionally. The difficulty is because the number of subject classes is large, as well as the number of full-time lecturers in the subject, and the expertise of the full-time lecturers is very different. The emotionality is because at present there is only one measurable criterion, which is to assign full-time lecturers in charge of subject classes as much as possible so as not to have to invite additional visiting lecturers outside the school. However, the criteria of teacher satisfaction have not been clearly paid attention to.

In this study, the authors propose a quantitative way of teachers' satisfaction by collecting their interest in subjects through priority for each subject and their maximum teaching capacity in each semester. In fact, some teachers like to teach their familiar subjects, while others like to teach new subjects. Therefore, in this study, we use the prioritization information of taught subjects provided by the lecturers themselves. In the two factors affecting the satisfaction of teachers, the maximum teaching ability should be satisfied, then the priority for each subject should be met as much as possible.

This problem is to assign full-time lecturers in charge of teaching subject classes so that they satisfy 2 criteria in order of priority:

TC1. Assign teachers to be in charge of subject classes as much as possible.

TC2. Satisfaction of all instructors is obtained as high as possible.

This is the lexico graphical order approach, which means the TC1 is more prioritized than the TC2. It is noticeable that there cannot exist an optimal solution for TC1 that exceeds the maximum teaching capacity of a certain teacher. Because satisfaction in TC2 is modeled by teacher preferences, in which, teachers have the right to propose the upper bound of the teaching load. In other words, the upper bound of the teaching load for each lecturer is a hard-constraint of the problem. Thus, the optimization of TC1 does not cause any contradiction with TC2.

2 Related works

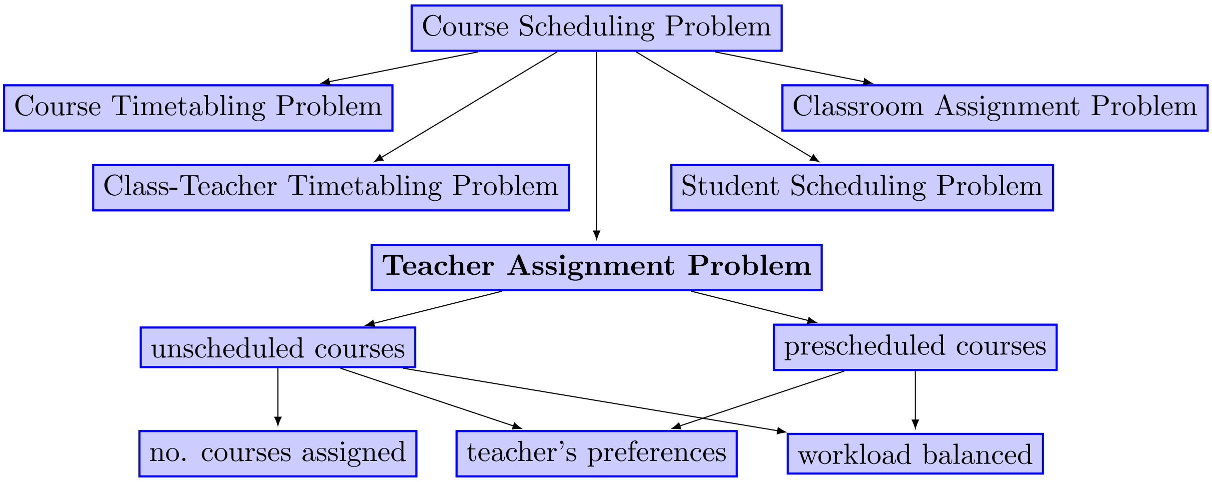

Figure 1: Classification of teacher assignment problem

According to [2], the course scheduling problem is normally decomposed into five different subproblems as shown in Figure 1. These are: (1) course timetabling problem: assign courses to time periods, (2) class-teacher timetabling problem: schedule class-teacher meetings without creating conflicts and while satisfying some side constraints on the spread and sequencing of courses (this problem arises mostly in high schools), (3) student scheduling problem: once students have selected their courses, they must be assigned to sections (this problem occurs when courses are taught in multiple sections), (4) teacher assignment problem: assign teachers to courses while maximizing a preference function, (5) classroom assignment problem: assign events to specific rooms to satisfy the size, location, and facility preferences and restrictions. Each of these problems, although seemingly simple at first, is actually relatively complex, so these problems (including the teacher assignment problem) are NP-hard problems. Therefore, each problem is often solved separately from the other problems related to the university timetabling ([1]).

There are usually 3 approaches to solve NP-hard problems, that is to build mathematical programming models or use algorithms, or combine both to determine feasible solutions to the problem.

3 Problem statement

The problem of assigning lecturers based on priority is described as the following:

- Each training program has a list of subjects, and each subject has a list of full-time lecturers.
- At the beginning of each year, the head of the department asks the teachers to register the subjects that the teacher can teach in the curriculum according to the priority that is not the same for each subject, numbered from 1 to n, with 1 being the most desired to teach.
- At the beginning of each semester, the training department calculates the number of courses that needs to be opened for each subject and sends this list to the head of the department.
- Then, the head of the department sends the list of these subject classes to the teacher and the teachers register the maximum number of classes that the teacher can take this semester.

- Based on the maximum number of classes the teacher can take in the semester and the subjects that the teacher has registered, the head of the department assigns the teacher to the list of subject classes to satisfy two main goals:

  O1. Maximum number of classes that can be assigned to teachers.

  O2. Minimum total priority of all assigned teachers.

An illustrative data for this problem is shown as follows:

- There are 10 subjects in the curriculum that are numbered from C01 to C10 in the Table 1, and there are 5 teachers in the subject that are numbered from T01 to T05 in the Table 2.
- At the beginning of year, teachers register the subjects that can be taught with the priority as shown in the Table 20:
  - Teacher T01 registers the course C01 with priority 1, course C02 with priority 2, course C03 with priority 3, course C04 with priority 4.
  - Teacher T02 registers the course C03 with priority 1, course C04 with priority 2, course C05 with priority 3, course C06 with priority 4.
  - Teacher T03 registers the course C05 with priority 1, course C06 with priority 2, course C07 with priority 3, course C08 with priority 4.
  - Teacher T04 registers the course C07 with priority 1, course C08 with priority 2, course C09 with priority 3, course C10 with priority 4.
  - Teacher T05 registers the course C09 with priority 1, course C10 with priority 2, course C01 with priority 3, course C02 with priority 4.
- At the beginning of semester, the training department counts the number of classes that need to be opened for each course as shown in Table 21:
  - Course C03 needs to be opened with 1 class.
  - Course C04 needs to be opened with 2 classes.
  - Course C05 needs to be opened with 3 classes.
  - Course C06 needs to be opened with 2 classes.
  - Course C07 needs to be opened with 1 class.
- Teachers register the maximum number of classes that they can take in a semester as shown in Table 22:
  - Teacher T01 can take maximum 2 classes.
  - Teacher T02 can take maximum 3 classes.
  - Teacher T03 can take maximum 2 classes.
  - Teacher T05 can take maximum 1 class.

Table 1: Full courses in the curriculum

Course ID	**Course Name**	**Basic**
C01	Course 01	1
C02	Course 02	1
C03	Course 03	1
C04	Course 04	1
C05	Course 05	1
C06	Course 06	0
C07	Course 07	0
C08	Course 08	0
C09	Course 09	0
C10	Course 10	0

Table 2: Full teachers in the department

Teacher ID	**Teacher Name**
T01	Teacher 01
T02	Teacher 02
T03	Teacher 03
T04	Teacher 04
T05	Teacher 05

Table 3: Registration of courses with the priority of the teacher

Course\Teacher	T01	T02	T03	T04	T05
C01	1				3
C02	2				4
C03	3	1			
C04	4	2			
C05		3	1		
C06		4	2		
C07			3	1	
C08			4	2	
C09				3	1
C10				4	2

Table 4: Classes to be opened for this semester

Course ID	No. Classes
C03	1
C04	2
C05	3
C06	2
C07	1
Total	**9**

Table 5: Registered teachers for this semester

Teacher ID	Max. Classes
T01	2
T02	3
T03	2
T05	1
Total	**8**

Table 6: Feasible solutions for assigning teachers to classes for this semester

No.	Class ID	Course ID	Solution 1		Solution 2		Solution 3		Solution 4	
			Teacher ID	Priority	Teacher ID	Priority	Teacher ID	Priority	Teacher ID	Priority
1	C03-1	C03	T01	3	T01	3	T02	1	T01	3
2	C04-1	C04	T01	4	T01	4	T01	4	T01	4
3	C04-2	C04	n/a	n/a	T02	2	T01	4	T02	2
4	C05-1	C05	T02	3	T02	3	T03	1	T02	3
5	C05-2	C05	T02	3	T02	3	T03	1	T03	1
6	C05-3	C05	T02	3	n/a	n/a	T02	3	T03	1
7	C06-1	C06	T03	2	T03	2	T02	4	T02	4
8	C06-2	C06	T03	2	T03	2	n/a	n/a	n/a	n/a
9	C07-1	C07	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Count			**7**	**-**	**7**	**-**	**7**	**-**	**7**	**-**
Sum			**-**	**20**	**-**	**19**	**-**	**18**	**-**	**18**
StdDev			**-**	**-**	**-**	**-**	**-**	**1.51**	**-**	**1.27**

Based on the maximum classes that teachers can teach in a semester in the Table 22 and classes registered by teachers in the Table 20, the department head can have solutions to assign teachers into classes as in the Table 6:

- Solution 1: The number of assigned classes is 7, sum of priority of assigned teachers is 20, standard deviation of priority of assigned teachers is 0.69.
- Solution 2: The number of assigned classes is 7, sum of priority of assigned teachers is 19, standard deviation of priority of assigned teachers is 0.76.
- Solution 3: The number of assigned classes is 7, sum of priority of assigned teachers is 18, standard deviation of priority of assigned teachers is 1.51.
- Solution 4: The number of assigned classes is 7, sum of priority of assigned teachers is 18, standard deviation of priority of assigned teachers is 1.27.

Based on the criterion $O1$, these 4 solutions has the number of assigned classes is 7. Based on the criterion $O2$, solution 3 and 4 has the minimum sum of priority of assigned teachers, which is 18.

4 Mathematical modeling

Firstly, we build a priority matrix ($n \times m$), where n is the number of teachers register to teach in the semester and m is the number of classes that need to be opened in the semester. In which, values in cells are teacher's priority values for that subject, and if the teacher does not register to teach a subject, the value in that box will be 999 (large enough) as in the Table 7.

Table 7: Priority matrix ($n \times m$)

Teacher ID n Class ID	**C03-1**	**C04-1**	**C04-2**	**C05-1**	**C05-2**	**C05-3**	**C06-1**	**C06-2**	**C07-1**
T01	3	4	4	999	999	999	999	999	999
T02	1	2	2	3	3	3	4	4	999
T03	999	999	999	1	1	1	2	2	3
T05	999	999	999	999	999	999	999	999	999

Selecting or not selecting a cell in the priority matrix is assigning or not assigning a teacher to a corresponding subject class. The IP model for selecting cells in the priority matrix is shown below:

- Decision variable: $x_{i,j} \in \{0, 1\}$ has value 1 if the teacher i $(i = 1, \ldots, n)$ is assigned to the class j $(j = 1, \ldots, m)$, and 0 otherwise. Basic-subject classes are denoted as C_B.
- Objective function: minimum sum of priority of assigned teachers, where $c_{i,j}$ is the priority at the cell (i, j).

$$\min(\sum_{i=1}^{n}\sum_{j=1}^{m} c_{i,j} \times x_{i,j}) \tag{1}$$

- Subject to:
  - Each class is assigned with maximum one teacher:

$$\sum_{i=1}^{n} x_{i,j} \leq 1, \ \ \forall j = 1, \ldots, m \tag{2}$$

  - Each teacher is assigned with a maximum of mc of classes, where mc is the maximum number of classes that the teacher registers to teach in the semester:

$$\sum_{j=1}^{m} x_{i,j} \leq mc_i, \ \ \forall i = 1, \ldots, n \tag{3}$$

  - Either all classes are assigned with 1 teacher, or all teachers are assigned with mc classes that they registered:

$$(\sum_{i=1}^{n} x_{i,j} - 1) \times (\sum_{j=1}^{m} x_{i,j} - mc_i) = 0, \ \ \forall i = 1, \ldots, n, \ \ \forall j = 1, \ldots, m \tag{4}$$

  - Basic-subject classes must be opened:

$$\sum_{i=1}^{n} x_{i,j} = 1, \ \ \forall j \in C_B \tag{5}$$

5 Proposed method

Our target is transforming the original problem into a naive assignment problem which can be solved by the Hungarian algorithm. To do so, the final problem needs to be free-constraint so that the Hungarian algorithm can be directly applied.

The structure in this section is organized as the following. Firstly, we describe how to transform the original problem into a standard assignment problem. Then, a new mathematical model is re-formulated. Lastly, we also introduce a Genetic algorithm to solve the original problem in order to have a comparison with the proposed method.

5.1 Transform the original problem into a standard assignment problem

Firstly, consider the **constraint 1** in the Eq. 2, the cost matrix is unbalanced because the number of classes and the maximum number of classes can be taught by teachers can be different. In case of the number of classes is greater than the maximum number of classes can be taught by teachers, some classes may not be assigned to a teacher, resulting the '$<$' sign, otherwise, the '$=$' sign happens. This constraint is valid itself because the output of the assignment cannot assign a class into two teachers and some classes may have no teacher. Therefore, we can ignore this constraint without affecting the result of the original problem.

Secondly, regarding the **constraint 2** in the Eq. 3, a teacher may receive multiple classes, but cannot exceed the maximum number that they registered at the beginning. To solve this, we replicate teachers by their corresponding maximum number of classes can be taught, and treat these replicated teachers and original teachers independently. For example, if teacher T01 can teach maximum 2 classes, we will duplicate it into T01-1 and T01-2. Two these augmented teachers share the same priority, but are considered as independent instances in the assignment process. This do not violate the constraint because if the teacher T01 can be assigned with his maximum classes (2 classes), T01-1 and T01-2 are equivalently assigned all. If he is assigned with 1 class, then either T01-1 or T01-2 is assigned. Then, the priority matrix turns into:

Table 8: Replicated priority matrix ($n_1 \times m_1$)

Teacher ID n **Class ID**	**C03-1**	**C04-1**	**C04-2**	**C05-1**	**C05-2**	**C05-3**	**C06-1**	**C06-2**	**C07-1**
T01-1	3	4	4	999	999	999	999	999	999
T01-2	3	4	4	999	999	999	999	999	999
T02-1	1	2	2	3	3	3	4	4	999
T02-2	1	2	2	3	3	3	4	4	999
T02-3	1	2	2	3	3	3	4	4	999
T03-1	999	999	999	1	1	1	2	2	3
T03-2	999	999	999	1	1	1	2	2	3
T05-1	999	999	999	999	999	999	999	999	999

Then, for the **constraint 3** in the Eq. 4, we break it into two sub-constraints corresponding to the two multipliers in the left side of the equation. The former ($\sum_{i=1}^{n} x_{i,j} - 1 = 0$) means all classes are assigned with 1 teacher. This is equivalent to the case that the number of teachers is not smaller than the number of classes. Indeed, when there are more teachers than classes, i.e. $n_1 > m_1$, we can pad zero columns into the Replicated priority matrix to be square. Then, using Hungarian algorithm, each class is matched to a teacher, which meets the criterion that all classes are assigned. Regarding the latter ($\sum_{j=1}^{m} x_{i,j} - mc_i = 0$), it means all teachers are assigned with their maximum mc_i classes. Similarly, we can prove that this case is equivalent to the number of classes is larger than the number of teachers, then we can pad zero rows to obtain square cost matrix and solve by using Hungarian matching. Finally, the cost matrix becomes:

Table 9: Padded priority matrix ($n_2 \times n_2$)

Teacher ID n **Class ID**	**C03-1**	**C04-1**	**C04-2**	**C05-1**	**C05-2**	**C05-3**	**C06-1**	**C06-2**	**C07-1**
T01-1	3	4	4	999	999	999	999	999	999
T01-2	3	4	4	999	999	999	999	999	999
T02-1	1	2	2	3	3	3	4	4	999
T02-2	1	2	2	3	3	3	4	4	999
T02-3	1	2	2	3	3	3	4	4	999
T03-1	999	999	999	1	1	1	2	2	3
T03-2	999	999	999	1	1	1	2	2	3
T05-1	999	999	999	999	999	999	999	999	999
Txxx	0	0	0	0	0	0	0	0	0

Lastly, in term of the **constraint 4** in the Eq. 5, basic-subject classes are required to be opened if existing student register, while, optional-subject classes are not compulsory and the department head try to assign as much as possible teachers to these classes. Therefore, basic-subject classes must be paid more attention than optional-subject ones. To reveal this attention, in the priority matrix, we add the max priority (e.g. 4) into cells corresponding to optional-subject classes. As a result, Hungarian algorithm will prioritize basic-subject ones. The priority matrix becomes:

Table 10: Added priority matrix ($n_2 \times n_2$). Bold and italic numbers are added to the max priority

Teacher ID n **Class ID**	**C03-1**	**C04-1**	**C04-2**	**C05-1**	**C05-2**	**C05-3**	**C06-1**	**C06-2**	**C07-1**
T01-1	3	4	4	999	999	999	999	999	999
T01-2	3	4	4	999	999	999	999	999	999
T02-1	1	2	2	3	3	3	***8***	***8***	999
T02-2	1	2	2	3	3	3	***8***	***8***	999
T02-3	1	2	2	3	3	3	***8***	***8***	999
T03-1	999	999	999	1	1	1	***6***	***6***	***7***
T03-2	999	999	999	1	1	1	***6***	***6***	***7***
T05-1	999	999	999	999	999	999	999	999	999
Txxx	0	0	0	0	0	0	0	0	0

5.2 Reformulate mathematical model

The mathematical model is reformulated as the following:

- The cost matrix $C_2 = \{c_{ij}\}$ $(\forall i = 1, \ldots, n_2; \forall j = 1, \ldots, n_2)$ is the latest transformed priority matrix provided shown in the Tab. 10.
- Decision variable: $x_{i,j} \in \{0, 1\}$ has value 1 if the teacher i $(i = 1, \ldots, n_2)$ is assigned to the class j $(j = 1, \ldots, n_2)$, and 0 otherwise.
- Objective function: minimum sum of priority of assigned teachers, where $c_{i,j}$ is the priority at the cell (i, j).

$$\min\left(\sum_{i=1}^{n_2}\sum_{j=1}^{n_2} c_{i,j} \times x_{i,j}\right) \tag{6}$$

This transformed problem can be directly solved by the Hungarian algorithm. After matching, the assignment is then post-processed to obtain a final semantic result. There are following cases:

- If the assignment pair has value of 999, this pair would be canceled because the teacher does not register the class.
- In terms of basic-subject classes, if the number of matched teachers are not enough, an alert is notified so that the department head can contact part-time lectures in order to ensure unassigned classes must be opened.

5.3 Genetic algorithm

We also introduce Genetic algorithm-based solution to compare with the above proposed approach solved by Hungarian algorithm. Firstly, we describe the representation for the Genetic algorithm, then describe how the Genetic algorithm work to solve the assignment problem.

5.3.1 Representation for the Genetic algorithm

We still follow the mathematical model describe in Section 5.2, but some representation are changed in order to be compatible to Genetic algorithm, which are class and teacher representations. For classes, we represent them by a vector:

$$O = [1, 2, 3, ..., \#classes] \tag{7}$$

while teachers are represented by a vector:

$$P = [1, 2, 3, ..., \#teachers] \tag{8}$$

Because we pad the cost matrix to be square, so the number of classes and teachers are expected to be equal (i.e. $\#classes = \#teachers$). Therefore we also have the two vector O and P have the same length:

$$|O| = |P| = n_2 \tag{9}$$

In term of assignment, a class in vector O is assigned to a teacher in vector P with the same location:

$$(O_i, P_i), \ \forall i = 1, \ldots, n_2 \tag{10}$$

The target of the Genetic algorithm is to fix the vector O and find a permutation of vector P so that the total priority is as small as possible.

5.3.2 Pipeline of the Genetic algorithm

The pipeline of Genetic algorithm-based solution is depicted in Fig. 2. Specifically, we initialize a population with 20 individuals, then, based on the fitness function (i.e. the summed priority of assigned teacher-class pairs), we select 2 individuals as a parent with the smallest fitness values. Afterwards, crossover and mutation step (described in Alg. 1) is performed in order to create children. Each parent will create 2 children. These 2 new individuals join to the population and the re-population is performed to eliminate two individuals having largest fitness scores. After that, the process is repeated with the parent selection step and the loop is conducted until existing an individual has its fitness score lower than a pre-defined threshold.

6 Experimental results

Firstly, we define 5 test cases, then report results of the 2 methods (e.g., Hungarian-based and Genetic-based) for these test cases.

- Case 1: The number of classes is equal to the number of teachers.

Table 11: (Case 1) Registration of courses with the priority of the teacher

Course\Teacher	**T01**	**T02**	**T03**	**T04**	**T05**
C01	1	3	1	0	2
C02	2	2	2	3	3
C03	3	1	3	2	0
C04	4	0	0	1	1

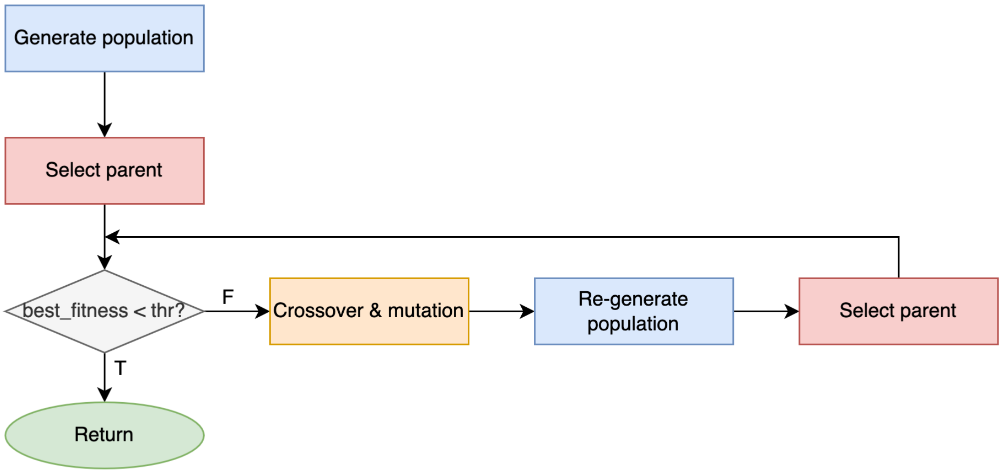

Figure 2: Genetic algorithm pipeline

Algorithm 1 Crossover-mutation step in the Genetic algorithm

Data: Dad and mom vectors: $\{d_i\}, \{m_i\}\ \forall i = 1, \ldots, n_2$
Result: Child vector: $\{c_i\}\ \forall i = 1, \ldots, n_2$

```
for i in range(n2) do
    if i is odd then
        if d[i] not in c then
            c[i] = d[i]                               ▷ take from dad
        else
            missedIds.append(i)                       ▷ ignore, fill it later
        end
    end
    if i is even then
        if m[i] not in c then
            c[i] = m[i]                               ▷ take from mom
        else
            missedIds.append(i)                       ▷ ignore, fill it later
        end
    end
end
for i in missedIds do
    randomly select remaining elements to fill the child vector
end
```

Table 12: (Case 1) Classes to be opened

Course ID	Basic	No. Classes
C01	1	3
C02	0	2
C03	1	2
C04	0	4
Total		**11**

Table 13: (Case 1) Registered teachers

Teacher ID	Max. Classes
T01	3
T02	2
T03	2
T04	3
T05	1
Total	**11**

- Case 2: The number of classes is smaller than the number of teachers.

Table 14: (Case 2) Registration of courses with the priority of the teacher

Course\Teacher	**T01**	**T02**	**T03**	**T04**	**T05**	**T06**
C01	1	3	1	0	2	1
C02	2	2	2	3	3	0
C03	3	1	3	2	0	0
C04	4	0	0	1	1	0

Table 15: (Case 2) Classes to be opened

Course ID	**Basic**	**No. Classes**
C01	1	3
C02	0	2
C03	1	2
C04	0	4
Total		**11**

Table 16: (Case 2) Registered teachers

Teacher ID	**Max. Classes**
T01	3
T02	2
T03	2
T04	3
T05	1
T06	3
Total	**14**

- Case 3: The number of classes is larger than the number of teachers.

Table 17: (Case 3) Registration of courses with the priority of the teacher

Course\Teacher	**T01**	**T02**	**T03**	**T04**	**T05**	**T06**
C01	1	3	1	0	2	1
C02	2	2	2	3	3	0
C03	3	1	3	2	0	0
C04	4	0	0	1	1	0
C05	0	4	0	4	0	2

Table 18: (Case 3) Classes to be opened

Course ID	**Basic**	**No. Classes**
C01	1	3
C02	0	2
C03	1	2
C04	0	4
C05	1	4
Total		**15**

Table 19: (Case 3) Registered teachers

Teacher ID	**Max. Classes**
T01	3
T02	2
T03	2
T04	3
T05	1
T06	1
Total	**12**

- Case 4: As described in Section 3.
- Case 5: There is a basic subject but no one want to teach it.

Table 20: (Case 5) Registration of courses with the priority of the teacher

Course\Teacher	**T01**	**T02**	**T03**	**T04**	**T05**
C01	1	0	0	0	3
C02	2	0	0	0	4
C03	3	1	0	0	0
C04	4	2	0	0	0
C05	0	3	1	0	0
C06	0	4	2	0	0
C07	0	0	3	1	0
C08	0	0	4	2	0
C09	0	0	0	3	1
C10	0	0	0	4	2

Table 21: (Case 5) Classes to be opened

Course ID	**Basic**	**No. Classes**
C01	1	0
C02	1	0
C03	1	1
C04	1	2
C05	1	3
C06	0	2
C07	0	1
C08	0	0
C09	0	0
C10	0	0
Total		**9**

Table 22: (Case 5) Registered teachers

Teacher ID	**Max. Classes**
T01	2
T02	3
T03	2
T04	0
T05	1
Total	**8**

Results of the Hungarian-based and Genetic-based methods are evaluated via metrics:

Count	The total number of assigned teacher-class pairs
Sum	The sum of assigned teachers' priorities
Std	The standard deviation of assigned teachers' priorities
Mean	The mean of assigned teachers' priorities
Runtime (ms)	The average runtime over 100 tries, measured in millisecond

Then, we summarize assignment results as the following:

- Case 1: The assignment results of the Hungarian-based and Genetic-based methods are shown in Tab. 23. It can be seen that both the methods have a comparable solution in terms of all metrics, except runtime. The Hungarian-based one is faster than the Genetic-based one.

Table 23: (Case 1) Assignment results of the Hungarian-based and Genetic-based methods

Course	**Hungarian**		**Genetic**	
	Teacher	**Priority**	**Teacher**	**Priority**
C01	T01	1	T03	1
C01	T01	1	T01	1
C01	T01	1	T01	1
C02	T03	2	T03	2
C02	T03	2	T01	2
C03	T02	1	T02	1
C03	T02	1	T02	1
C04	T04	1	T04	1
C04	T04	1	T05	1
C04	T04	1	T04	1
C04	T05	1	T04	1
Count	11		11	
Sum		13		13
Std		0.4		0.4
Mean		1.18		1.18
Time(ms)		6		260

- Case 2: The assignment results of the Hungarian-based and Genetic-based methods are shown in Tab. 24. It can be seen that both the methods have a comparable solution in term of all metrics, excepting runtime. The Hungarian-based one is faster the Genetic-based one.

Table 24: (Case 2) Assignment results of the Hungarian-based and Genetic-based methods

Course	**Hungarian**		**Genetic**	
	Teacher	**Priority**	**Teacher**	**Priority**
C01	T01	1	T01	1
C01	T01	1	T03	1
C01	T01	1	T06	1
C02	T03	2	T01	2
C02	T03	2	T01	2
C03	T02	1	T02	1
C03	T02	1	T02	1
C04	T04	1	T04	1
C04	T04	1	T05	1
C04	T04	1	T04	1
C04	T05	1	T04	1
Count	11		11	
Sum		13		13
Std		0.4		0.4
Mean		1.18		1.18
Time(ms)		7		500

- Case 3: The assignment results of the Hungarian-based and Genetic-based methods are shown in Tab. 25. It can be seen that both the methods have a comparable solution in term of all metrics, excepting runtime. The Hungarian-based one is faster the Genetic-based one.

Table 25: (Case 3) Assignment results of the Hungarian-based and Genetic-based methods

Course	**Hungarian**		**Genetic**	
	Teacher	**Priority**	**Teacher**	**Priority**
C01	T01	1	T01	1
C01	T01	1	T01	1
C01	T01	1	T03	1
C02	T03	2	T03	2
C02	T03	2	T01	2
C03	T02	1	T02	1
C03	T02	1	T02	1
C04	n/a	n/a	T05	1
C04	n/a	n/a	n/a	n/a
C04	n/a	n/a	n/a	n/a
C04	T05	1	n/a	n/a
C05	T06	2	T06	2
C05	T04	4	T04	4
C05	T04	4	T04	4
C05	T04	4	T04	4
Count	12		12	
Sum		24		24
Std		1.28		1.28
Mean		2.0		2.0
Time(ms)		8		596

- Case 4: The assignment results of the Hungarian-based and Genetic-based methods are shown in Tab. 26. It can be seen that the former is partially better than the latter. Concretely, both the methods have a comparable solution in term of Count, Sum, and Mean, while for Std and Time, the Hungarian-based one is better than the Genetic-based one.

Table 26: (Case 4) Assignment results of the Hungarian-based and Genetic-based methods

Course	**Hungarian**		**Genetic**	
	Teacher	**Priority**	**Teacher**	**Priority**
C03	T01	3	T02	1
C04	T02	2	T01	4
C04	T01	4	T01	4
C05	T03	1	T02	3
C05	T03	1	T03	1
C05	T02	3	T03	1
C06	T02	4	T02	4
C06	T05	n/a	T05	n/a
C07	n/a	n/a	n/a	n/a
Count	7		7	
Sum		18		18
Std		1.27		1.51
Mean		2.57		2.57
Time(ms)		8		123

- Case 5: The assignment results of the Hungarian-based and Genetic-based methods are shown in Tab. 27. It can be seen that the Hungarian-based one is totally better than the Genetic-based one because the former can exploit more teacher than the latter. It is noticeable that the course C05 is a basic-subject but no one is assigned to teach this class (highlight in the red color). Therefore, the head of department must contact part-time lectures in order to host these basic-subject classes.

Table 27: (Case 5) Assignment results of the Hungarian-based and Genetic-based methods

Course	**Hungarian**		**Genetic**	
	Teacher	**Priority**	**Teacher**	**Priority**
C03	T02	1	T02	1
C04	T02	2	T02	2
C04	T01	4	T02	2
C05	n/a	n/a	T01	n/a
C05	T01	n/a	T01	n/a
C05	T05	n/a	T05	n/a
C06	T02	4	T03	2
C06	T03	2	T03	2
C07	T03	3	n/a	n/a
Count	6		5	
Sum		16		9
Std		1.21		0.45
Mean		2.67		1.80
Time(ms)		9		280

7 Conclusion

In this work, we propose to transform the original teacher-class assignment problem with constraints into a standard assignment problem so that the Hungarian algorithm can be used to solve it. In addition, we also exploit the Genetic algorithm as an alternative solution to compare with the proposed method. The proposed method can not only obtain a better solution but also be faster than the competitor, while the Genetic-based solution does not always achieve a comparable output due to the randomness. However, we believe that the Genetic algorithm is more flexible than the proposed method in term of constraints. In other words, embedding constraints into the Genetic algorithm is likely to be easier than transforming the original problem to a standard assignment problem. In the future, we will investigate other conditional assignment problems to verify this. The source code of this work can be found in the link https://github.com/dotieuthien/Genetic-Algorithm-for-Assignment-Problem.

References

[1] Avella, P., & Vasil'Ev, I. (2005) A computational study of a cutting plane algorithm for university coursetimetabling. **Journal of Scheduling** 8, 497 – 514.

[2] Carter, M. W., & Laporte, G. (1998) Recent developments in practical course timetabling. **Practice and Theory of Automated Timetabling II**, 1408, 3 – 19.