Mathematical Modeling of a Time table scheduling system for the University of Yaounde 1

Folong Emerson
Fomubad Borista Fondi
Fouotsop Fosso Patrick
Kemtho Paul Zidane
Ouanko Belvain Aubrey
Groupe 6

May 18, 2024

Contents

1 Problem formulation

Increase in number of students every years against the number of teachers and available classrooms makes it difficult to plan classes in the university of Yaounde 1. Classrooms are overbooked leaving no rooms for students on free periods to study, Teachers are overworked and many students are late for classes. The university of Yaounde 1 is looking for a solution to this problem by developing a scheduling system that will help to optimize the use of classrooms and teachers. The system should be able to assign courses to classrooms and teachers in a way that will minimize the number of late students, the number of overworked teachers and increase the number of available rooms for students on free periods to study

2 Sets identification and definition

2.1 Sets and Variables

1.
2. $D = \{mon, tue, wed, thur, fri, sat, sun\}$: set of days of the week
3. $P = \{p_1, p_2, p_3, p_4, p_5\}$: set of class periods (time slots), with :
   (a) p_1 = 7:00am - 9:55:am
   (b) p_2 = 10:05am - 12:55:am
   (c) p_3 = 1:05pm - 3:55:pm
   (d) p_4 = 4:05pm - 6:55pm
   (e) p_5 = 7:05pm - 9:55:pm
4. $C = \{c_1, \ldots, c_n\}$: set of n courses
5. $T = \{t_1, \ldots, t_m\}$: set of m teachers
6. $R = \{r_1, \ldots, r_l\}$: set of l classrooms
7. $L = \{l_1, \ldots, l_k\}$: set of k levels e.g. L_1-computer sciences, L_2-mathematics etc
8. $W = \{w | w \text{ is a week}\}$: set of weeks in a semester

3 Mathematical Formulation

3.1 Variables and Parameters

3.1.1 Variables

1. X_{crpd}: a boolean variable true when course c is assigned to room r on day d and period p
2. Y_{tcpd}: a boolean variable true when teacher t is assigned to course c on day d and period p
3. P_{wtc}: a boolean variable true when teacher t misses course c in week w
4. Z_{lc}: a boolean variable true when level l has course c in its curriculum
5. W_{tcd}: a teacher gives course c on day d

3.1.2 Parameters

1. S_l: size of the level L, i.e., number of students in level l
2. S_r: capacity of room r, the maximum number of students that can fit in room r

3.2 Constraints

3.2.1 Hard constraints

1. No more than one course can be set in any room on a day d and period p
2. No more than one course can be assigned to any teacher on day d and period p
3. No more than one course scheduled at period p on day d for any level
4. The maximum number of students taking a course in a room is not greater than the capacity of the room.

3.2.2 Soft constraints

1. A course should not be scheduled within working hours
2. A teacher should not be assigned to more than 3 courses in a day
3. A teacher should not be assigned to more than 5 courses in a week
4. A level should not have more than 3 courses in a day

3.3 Solution modeling

3.3.1 Constraints

1. *Course assignment constraint*

$$X_{crpd} = \begin{cases} 1 & \text{if course } c \text{ is assigned to room } r \text{ on day } d \text{ and period } p \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{c \in C} X_{crpd} \leq 1 \forall r \in R \quad \forall p \in P \quad \forall d \in D$$

2. *Teacher assignment constraint*

$$Y_{tcpd} = \begin{cases} 1 & \text{if teacher } t \text{ is assigned course } c \text{ on day } d \text{ and period } p \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{c \in C} Y_{tcpd} \leq 1 \forall t \in T \quad \forall p \in P \quad \forall d \in D$$

3. *Level assignment constraints*

$$Z_{lc} = \begin{cases} 1 & \text{if level } l \text{ has course } c \text{ in its curriculum} \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{l \in L} Z_{lc} \leq 1 \quad \forall c \in C$$

4. *Room capacity constraint*

$$X_{crpd} Z_{lc} S_l \leq S_r$$

5. *Teacher workload constraints*

$$W_{tcd} = \begin{cases} 1 & \text{if teacher } t \text{ is assigned to course } c \text{ on day } d \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{c \in C} W_{tcd} \leq 3 \quad \forall t \in T \quad \forall d \in D$$

$$\sum_{c \in C} \sum_{d \in D} W_{tcd} \leq 20 \quad \forall t \in T$$

mon	tue	wed	thu	Fri	Sat	Sun	mor	aft	evn
5	5	3	3	1	1	0.5	3	2	1

Table 1: Preference scale

3.3.2 Penalty function

The penalty function defines whether a teacher should receive or not a penalty, it returns true or false based on the number of scheduled classes missed by a teacher in a week. We define :

$$P : (T, W) \to \{0, 1\}$$

$$P(t, w) = \begin{cases} 0 & \sum_{c \in C} \sum_{d \in D} W_{tcd} P_{wtc} \leq 2 \\ 1 & \text{otherwise} \end{cases}$$

3.3.3 Objective functions

1. *Students*
   This function is maximized, meaning the goal is to schedule the courses in such a way that the total preference of the students for the scheduled periods is as high as possible. This will result in a timetable that is optimized according to the students' preference weight for different periods of the day. It is defined as follows

$$F_s = \max \sum_{c \in C} \sum_{r \in R} \sum_{d \in D} w_p * X_{crpd} \text{ with } w_p, \text{ criteria weight of period } p$$

2. *Teachers*
   Similarly for the teachers we define our objective function based on the teacher period preference weight and the day preference weight of the teacher. It is defined as follows

$$F_t = \max \sum_{c \in C} \sum_{p \in P} \sum_{d \in D} \sum_{r \in R} w_p * w_d * X_{crpd} \text{ with } w_p, \text{ criteria weight of period } p \text{ and } w_d, \text{ criteria weight of day } d$$

3.4 Model calculation using AHP

The preference scores set by the teachers and students for periods are:

- 5 for p1 and p2
- 3 for p3 and p4
- 1 for p5

The preference scores set by teachers for teaching days are:

- 5 for monday and tuesday
- 3 for wednesday and thursday
- 1 for friday and saturday
- 0 for sunday

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	mor	aft	evn
Mon	1	2	3	4	5	6	7	7	9	11
Tue	1/2	1	2	3	4	5	6	6	8	10
Wed	1/3	1/2	1	2	3	4	5	5	7	9
Thu	1/4	1/3	1/2	1	2	3	4	4	6	8
Fri	1/5	1/4	1/3	1/2	1	2	3	3	5	7
Sat	1/6	1/5	1/4	1/3	1/2	1	2	2	4	6
Sun	1/7	1/6	1/5	1/4	1/3	1/2	1	1	3	5
mor	1/7	1/6	1/5	1/4	1/3	1/2	1	1	3	5
aft	1/9	1/8	1/7	1/6	1/5	1/4	1/3	1/3	1	3
evn	1/11	1/10	1/9	1/8	1/7	1/6	1/5	1/5	1/3	1

Table 2: Pair wise comparison matrix for days and periods preference

Criterion	**Weight**
Mon	0.28933268
Tue	0.20914535
Wed	0.15063165
Thur	0.10830148
Fri	0.07743636
Sat	0.0548
Sun	0.03822925
Mor	0.03822925
Aft	0.02100165
Evn	0.01289233

Table 3: Criteria weights

The preference scores obtained for days prefered by teachers and periods of the day, here p_1 and p_2 for mor (morning) p_3 and p_4 for aft (afternoon) and p_5 for evn (evening)

Table 2 shows the pair wise comparison matrix for the preferences scores of the teachers, the preferences weight were assigned based on the number of teachers who selected the preference. We run a python program that calculates the criteria weights of each preferred day and period

The criteria weights are given as follows
$W_g = \{w_{mon}, w_{tue}, w_{wed}, w_{thr}, w_{fri}, w_{sat}, w_{sun}, w_{mor}, w_{aft}, w_{evn}\}$
With the results above we get the actual value of the objective function for teachers as follows

$$Ft = max \sum_{c \in C} \sum_{d \in D} \sum_{p \in P} \sum_{r \in R} w_p * w_d * x_{crpd} \quad w_p \in W_g$$

4 Conclusion

By taking into account the preferences of students and teacher we can compute the actual criteria weights necessary in the objective function of teachers and students by applying AHP. The function represents all the possible combination of courses, rooms, periods and days. F_t is maximized so as to consider combinations of courses periods rooms and days based on the most preferred days and periods of teachers and students ahp.py > ...

```python
import numpy as np
from fractions import Fraction

matrix = np.array([
    [1, 2, 3, 4, 5, 6, 7, 7, 9, 11],
    [1/2, 1, 2, 3, 4, 5, 6, 6, 8, 10],
    [1/3, 1/2, 1, 2, 3, 4, 5, 5, 7, 9],
    [1/4, 1/3, 1/2, 1, 2, 3, 4, 4, 6, 8],
    [1/5, 1/4, 1/3, 1/2, 1, 2, 3, 3, 5, 7],
    [1/6, 1/5, 1/4, 1/3, 1/2, 1, 2, 2, 4, 6],
    [1/7, 1/6, 1/5, 1/4, 1/3, 1/2, 1, 1, 3, 5],
    [1/7, 1/6, 1/5, 1/4, 1/3, 1/2, 1, 1, 3, 5],
    [1/9, 1/8, 1/7, 1/6, 1/5, 1/4, 1/3, 1/3, 1, 3],
    [1/11, 1/10, 1/9, 1/8, 1/7, 1/6, 1/5, 1/5, 1/3, 1]
])

# Calculate column sum
col_sum = np.sum(matrix, axis=0)

# Normalize the matrix
normalized_matrix = matrix / col_sum

# Calculate criteria weights (average of rows in the normalized matrix)
criteria_weights = np.mean(normalized_matrix, axis=1)

# Calculate lambda_max (sum of the product of col_sum and criteria_weights)
lambda_max = np.sum(col_sum * criteria_weights)

# Calculate consistency index (CI)
n = len(criteria_weights)  # number of criteria
CI = (lambda_max - n) / (n - 1)

# Calculate consistency ratio (CR)
RI = {1: 0, 2: 0, 3: 0.58, 4: 0.90, 5: 1.12, 6: 1.24, 7: 1.32, 8: 1.41, 9: 1.45, 10: 1.49}  # random index
CR = CI / RI[n]


criteria_weights, CI, CR
```

Figure 1: Calculating the criteria weights for all days and periods