**PROJECT REPORT**

# IDEAL TIMETABLE FOR A STUDENT

**MATHEMATICAL MODELLING**
**MTH426**

**BY**


| | |
|---|---|
| **Rohit Kumar Bose** | **150596** |
| **Robin Singh** | **150591** |
| **Sandeep Kumar** | **150629** |
| **Ashu Prakash** | **14146** |
| **Tushar Tiwari** | **14765** |


**UNDER THE GUIDANCE OF**
**Prof. Prawal Sinha**



**DEPARTMENT OF MATHEMATICS & STATISTICS**
**Indian Institute of Technology Kanpur**

# Acknowledgements

We are profoundly grateful to **Prof. Prawal Sinha** (Dept. of Mathematics and Statistics, IIT Kanpur) for his expert guidance and continuous encouragement right from its commencement to its completion, without which the work would not have been possible.

# ABSTRACT

Management of time is the most important art of a healthy and balanced life. All successful people are better at one thing: Time Management. Each of us spend a lot of time in creating a realistic timetable. In this research, we have created a mathematical algorithm for IIT Kanpur students to generate an effective and well organised weekly schedule. Timetabling problem needs to be defined and executed such that every student gets a balanced schedule to study and to follow other activities. In the process of this research, we also found some interesting statistics of concentration and confidence of the students of our campus.

# AIM

To output an ideal time table for the coming week given the preoccupations of a student

# Introduction

Timetable creation is a very arduous and time consuming task. To create timetable it takes lots of patience and man hours. Also, in many cases it is difficult to analyse one self in the process of making such a time schedule. A student always tries to restrict himself very tightly that he fails to follow his own proposals. There lies a motivation for this model. A timetable generated with the theories of psychology should definitely be of more benefit rather than its counterpart with no involvement of any science. So far our knowledge and research has shown that no one has ever tried for such a model which can output a unique timetable for a student and thus our research is pure.

In our project, we have proposed an algorithm for generating an ideal timetable for a student after taking into consideration his preoccupations for the coming week. We have used the knowledge of computer programming and mathematical modelling for the completion of this research. Our study is based on the students of IIT Kanpur. As input, our model asks the users for their courses, time of their extra-curricular activities and their confidence level in each course via a set of questionnaires. We have also taken few assumptions for our model.

This paper is a report-cum-manual of the algorithm. This states how the algorithm in the model works. We have tried to simplify the algorithm as much as we can, optimising its use for a student. There may be many future prospects with this model in helping the students.

# Literature Survey

The paper [1] suggests an algorithm for the generation of a classroom timetable. The paper solves the problem with a mimetic hybrid algorithm and genetic artificial immune network. In this study, Genetic Algorithms optimises the trade-off between exploring new points in the search space and exploring the information discovered thus far.

As mentioned in [2] the Pomodoro Technique is a time management tool that was originally intended to optimise personal work and study. The heart of the Pomodoro Technique is 25 minutes of focused, uninterrupted work on one task, then 5 minutes of rest. There are also rules to keep the integrity of Pomodoro, and tactics to deal with internal and external interruptions.

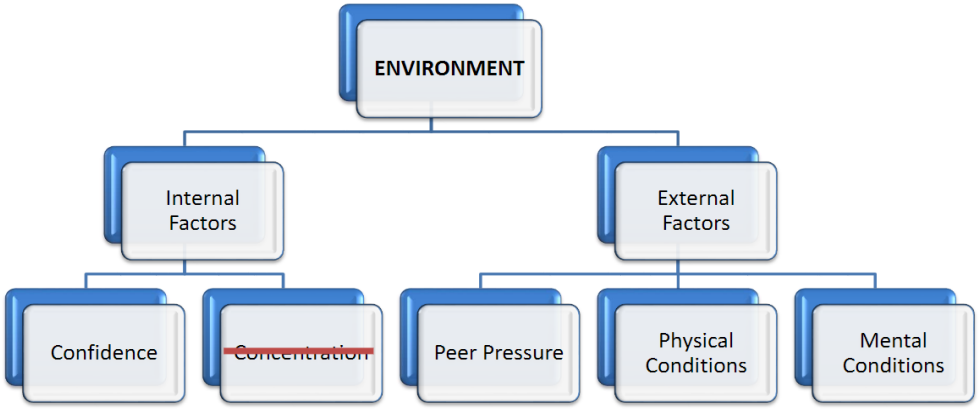
Timetable creation is itself a very time-consuming task. There is always a need to optimise the time by doing more and more work in less amount of time. Genetic algorithms [3] are typically used to solve such kind of problems in schools. Most of the work done these days focuses the lecture timetabling issues. In the research [4] the status of this problem is set to the non-polynomial complete problem. In our paper also, we are choosing a heuristic solution which is good and close to ideality but may not be the best.

In past, all the studies have been done on the construction of timetable for schools [5] in their organisation and not for the students. One common problem with all the models is the difficulty in differentiating the acceptable and non-acceptable timetables. Our model is nothing different from these regarding the ideality conditions. We have tried to validate our model with a primary survey, that we did in IIT Kanpur. After this, we assume that our model tries to approach towards an ideality condition.

# System Characterization

A system is characterized by how it responds to input signals of the mathematical model. It shows how the objective of our model is inferred by the environment. In our research, we have only one object: 'Student'. This is because the model works on the basis preoccupations of the student. The system is being affected by the environment which in turn is affected by the internal and external factors. Confidence and concentration are responsible for the change in internal factors. The external factors include aspects like peer pressure, physical conditions and mental state of the student. Note that we are only including internal factors in our model. External factors will be too vague to be introduced and if taken into account, they will add random variables making the model non-deterministic. We will not be considering concentration as a variable in our model, although it might seem reasonable to do so. This is explained further in the report.

**Figure 1**



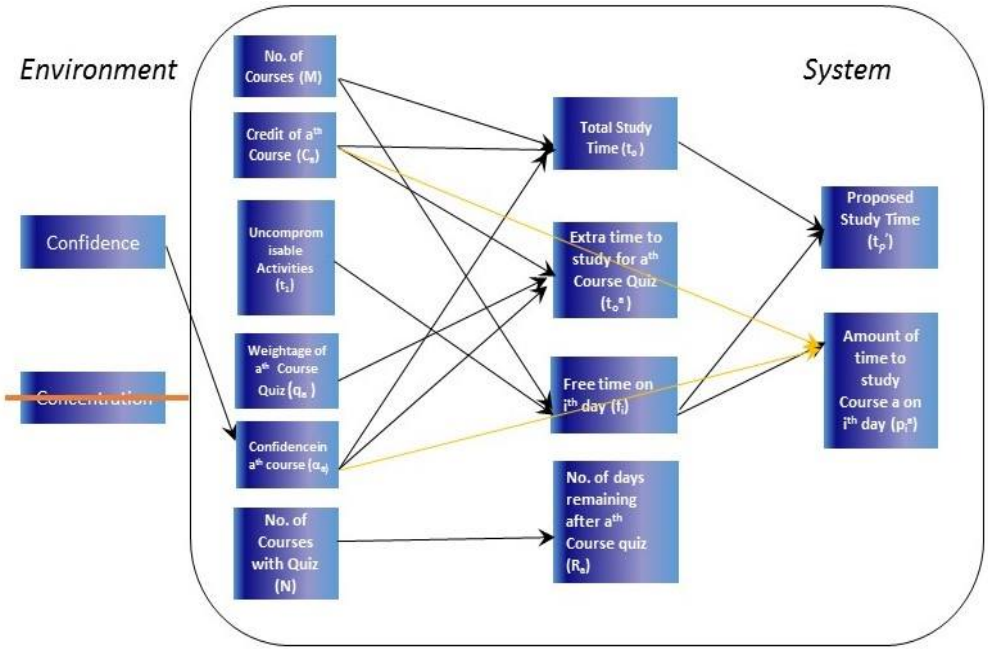

**Input Variables**

- Course and Credit (Name and $C_a$)
- Total free time ($t$)
- No. of hours for uncompromisable activities ($t_1$)
- Weightage of a$^{th}$ course Quiz ($q_a$)
- Day of a$^{th}$ course Quiz ($q_a^D$)
- Confidence in a$^{th}$ course ($\alpha_a$)
- Number of quizzes ($N$) and their days
- Number of Courses ($M$)

**Intermediate variables**

- Proposed Study time ($t_0$)
- No. of days remaining after a$^{th}$ course quiz ($R_a$)
- Free time on i$^{th}$ day ($f_i$)
- Extra time to study for a$^{th}$ course Quiz ($t_0^a$)

**Figure 2:** Causal Relationship Diagram




**Output variables**

- Proposed Study Time ($t_0'$)
- Amount of time to study of course a on i$^{th}$ day ($p_i^a$)

**Parameters**

- Credits for a$^{th}$ course
- Sleeping time
- Time for breakfast, lunch and dinner

Based on our current environment, variables and parameters with some assumptions, our model has the following characterization:

- **OPEN:** The variables of the system depends upon the environment. Since confidence is the part of the environment and it affects the working of the object in the model.
- **BLACK BOX:** The variables of the system depend upon the internal factors. This model does not use any concrete theory of human concentration.
- **STATIC:** Our model is completely static. We are modelling the timetable only for a week of the student. This means a student will have to generate the output from the model on every week basis.
- **DETERMINISTIC:** We are generating a timetable as an output. In the model, we have not taken external factors like peer pressure, physical or mental conditions of the student. Thus, we have not used any random variable into the model.

## Assumption

1. Concentration and Confidence are mutually exclusive quantities.
2. Number of hours to be dedicated to a course per week depends upon the credits ($C_a$) and confidence ($\alpha_a$) in that course.
3. Study time for each course is constant irrespective of quiz.
4. Student studies for at least 20 and at most 25 minutes at a stretch.(Pomodoro technique)
5. There are at most 3 quizzes in a week.
6. Parameters are strictly adhered to.

   Sleeping time = 12 am to 7 am
   Breakfast = 7 am to 8 am
   Lunch = 1 pm to 2 pm
   Dinner = 8 pm to 9pm
   Sanitary activities are managed by the student in this time as well.

# Survey Result

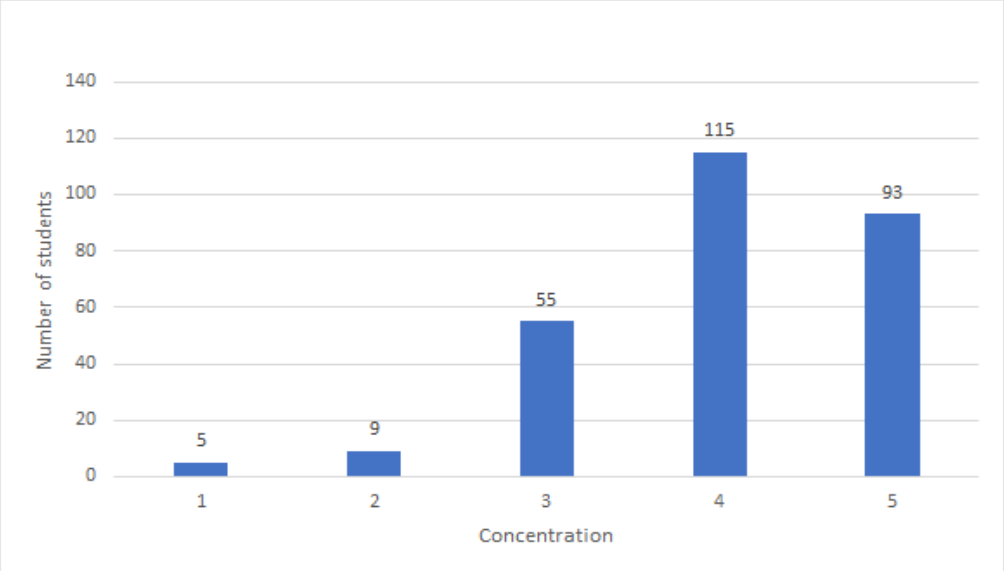
We conducted a primary survey involving the students of IIT Kanpur. This survey was filled by 277 campus students which really helped us in deciding the key elements of our model. We asked questions which allowed us to determine the confidence and concentration of a student.

We plotted a graph between the CPI of a student and his/her concentration and observed a considerable fluctuation in concentration level for any CPI range. This is really counter-intuitive. Since we cannot draw any reasonable relationship between these two quantities, we decided to exclude concentration as a variable from our model.

There was one question which asked the student to provide the number of extra hours which he/she will study for a quiz of 10% weightage of an 11 credit course 5 days from now. We took the average of all the responses for this question and found out that a student studies 4 hours extra than his/her normal study hours for a quiz of 10% weightage of an 11 credit course.

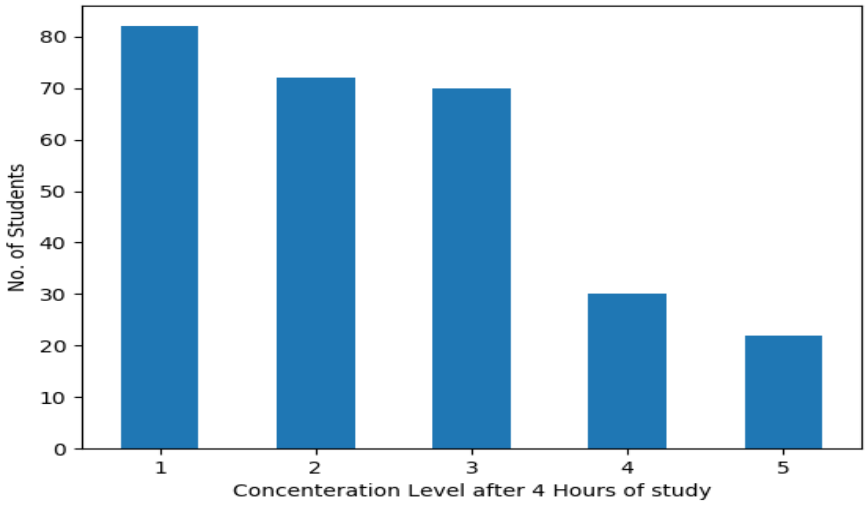
The first question of the survey was:
*How will you rate your concentration of study when you are highly motivated to study?*
As observed from the graph, on an average the highest possible value of concentration is 4.0180. Here, we assume that this is the highest possible value, because concentration level cannot be higher in cases when the student is not motivated to study.

**Figure 3**




We included a question in the survey asking for the concentration level of the student after 4 hours of study. From the responses, we inferred that it is much less than the average concentration ( = 2.4224). We have included this in our model by generating a timetable that does suggest 5 hours of continuous study. Such results from the timetable may occur at nights or on weekends for which, we may suggest the student for studying more than 5 hours. We have followed the Pomodoro technique, which suggests that it is best to do any involved task in intervals of 25 minutes with short breaks of 5 minutes. For example, for a slot of one hour, according to the technique mentioned [2] a student should study for a maximum of 50 minutes taking a break of 10 minutes. These breaks decrease the time of study we can incorporate in the free time available as per the student. Thus, in our model, the timetable will never result in more than 4 hours of continuous study.

**Figure 4**




The last question of the survey was:
*How much do you study from Monday onward if you get to know about a quiz of 10% weightage of an 11 credit course that will be held on Friday?*
We got a very wide range of responses varying from 1 hour to 12 hours. These entries in the data can't be ignored since they are not outliers (nearly 22.7% of the total responses). This huge variation in the data can be explained since these correspond to a large variety of students. Our timetable is an ideal timetable and we assume that the students who participated in the survey do not follow an ideal timetable which is true in general. Most of the students in campus work haphazardly. Students start preparing for the quiz when it is very near and they have very less confidence in the quiz, this explains the data entries which are very high. Now the average value of data is 3.9855 which is approximately equal to 4. In our model, we have calculated extra study time for quiz as

$$t_0^a = \frac{4q_a C_a}{\alpha_a}$$

The maximum value of extra study time for the quiz, according to our model comes when the concentration is minimum i,e $\alpha_a = 1$ which is 4.40, now this must be greater than the $3.9855 \approx 4$ because worst case value of our model must be greater than the average of the actual value.

# Formulation

## Confidence

The following questions will be asked for a particular course (say course ***a***), and the user has to enter a value between 1 to 5 for each question.

(1) *How will you rate your confidence level if you have attended all the scheduled classes?* $\alpha_a^1$
(2) *How much will you rate your interest in this course?* $\alpha_a^2$
(3) *How will you rate your confidence level if you havent attended any of your scheduled classes?* $\alpha_a^3$
(4) *How will you rate your confidence level one day before your mid-sem exam?* $\alpha_a^4$
(5) *How will you rate your confidence on the basis of the grading pattern for that course?* $\alpha_a^5$

The final confidence rating for course ***a*** is given by $\alpha_a$, where:

$$\alpha_a = \frac{\sum_{i=1}^{5} \alpha_a^i}{5}$$

## Initial Hypothesis

- $t$: free time
- $t_0$: proposed study time = $\sum_M \frac{c_a}{\alpha_a}$
- $t_1$: time for non-compromisable activities (given by the user)
- $t_2$: time for compromisable activities (given by the user),

where $c_a$= no. of credits of the $a^{th}$ course and $\alpha_a$ = confidence of the user in the same.

All the above values are proposed these are not the actual assigned values. Let the variables $t_0', t_1'$ and $t_2'$ denote the assigned values of the variables $t_0, t_1$ and $t_2$ respectively.

### Possible cases

The values of variables $t_0', t_1'$ and $t_2'$ depends upon the values of the variables $t_0, t_1$ and $t_2$. To calculate these values we have to consider following cases:

- **Case 1:-** $t \geq t_0 + t_1 + t_2$
  In the given case we have enough time to allocate to study and other activities. Hence we allocate time equal to proposed to time.

$$t_0' = t_0$$

$$t_1' = t_1$$

$$t_2' = t_2$$

- **Case 2:-** $t < t_0 + t_1 + t_2$
  This case can be further reduced to the following sub cases

  - **Sub case 1:-** $t \geq t_0 + t_1$
    For this case we have enough time to allocate to the academics and non-compromisable but not enough for compromisable. We will allocate time equal to their proposed values for academics and non-compromisable and the remaining time to compromisable.

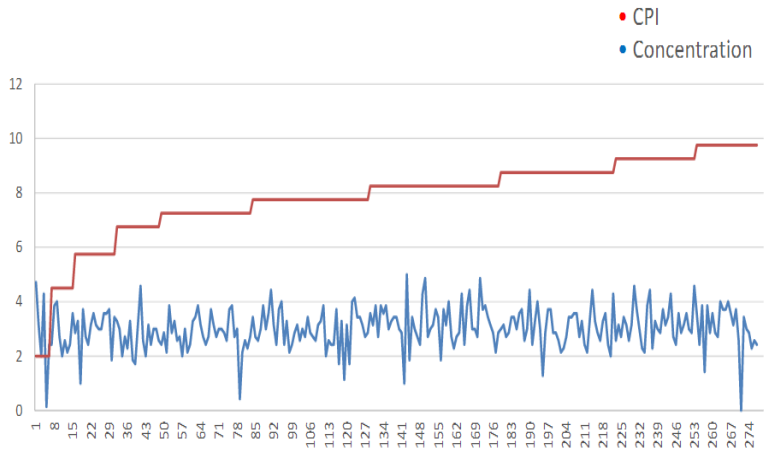
$$t_0' = t_0$$
$$t_1' = t_1$$
$$t_2' = t - t_0' - t_1'$$

– **Sub case 2:-** $t < t_0 + t_1$
This is the most compli1cated case. The approach we tried to use involves the previous academic performance of the user. Since we cannot allocate proposed time to both academics and non-compromisable activities, we have to reduce time for both here. The amount of time to be reduced from $t_0$ and $t_1$ depends on the concentration level: if he has high concentration then he needs less time to study and he can devote more time to non-compromisable activities and vice-versa.

$$t_0' = t_0 - \frac{\beta - 1}{4}x$$
$$t_1' = t_1' - \frac{5 - \beta}{4}x$$
$$t_2' = 0$$

here $\beta$ is a parameter which is directly proportional to the **concentration** and **CPI** of the user. For the above hypothesis we assumed that CPI increases with concentration and vice-versa. But the data collected from the primary survey proves that this assumption is not valid.

**Figure 5**




In the above graph, red line represent CPI and blue denotes concentration. Clearly a part of CPI curve which is parallel to X-axis denotes that a range of CPI and for any particular range of CPI the value of concentration is completely random. Hence we can't find a relation between concentration and CPI. Hence, our above hypothesis is wrong. ‘

Since the above hypothesis is wrong and we have start from scratch. The problem is to find a hypothesis which can be generalised for all the students and validate it from the data which covers a huge number of students from different departments and have done a huge variety of courses.

## Current Hypothesis

Since it turns out there is no relationship between the concentration and previous academic record of a student, we discarded these two quantities from our model, as they could not be of any substantial use in designing a proper time-table.

### Parameters

- $c_a$: credits of course $\boldsymbol{a}$, where $\boldsymbol{a}$ varies
- Sleeping time
- Time for breakfast, lunch and dinner

### Input Variables

- $M$: number of courses
- $N$: number of quizzes
- $t$: total free time
- $t_1$: time for non-compromisable activities
- $\alpha_a$: confidence rating in course $\boldsymbol{a}$
- $q_a$: % weightage of quiz of course $\boldsymbol{a}$
- $q_a^D$: day of quiz (1: Monday, 2: Tuesday etc)
- Slots for uncompromisable activities given explicitly

### Intermediate Variables

- $t_0$: proposed study time

$$t_0 = \sum_M \frac{c_a}{\alpha_a}$$

- $t_0^a$: Extra time to study for quiz of course $\boldsymbol{a}$, if present

$$t_0^a = \frac{4 q_a c_a}{\alpha_a}$$

- $f_i$: free time on $i^{th}$ day
- $R_a$: no. of days remaining after quiz of course $\boldsymbol{a}$

$$R_a = 7 - q_a^D$$

### Output Variables

- $p_i^a$: If course $\boldsymbol{a}$ has no quiz in the coming week, amount of time to study for it on $i^{th}$ day:

$$p_i^a = \frac{c_a}{\alpha_a} * \frac{f_i}{\sum_{i=1}^{7} f_i}$$

- $t_0^{'}$: proposed study time

$$t_0^{'} = \sum_{i=1}^{7} \sum_M p_i^a$$

## Algorithm to distribute courses in slots

We sort courses in increasing order of their study time, so that if there is a quiz on the next day, the student studies in the last slots. This will help him to retain information better for the quiz.

```
Divide courses into slots for a day
{
    M <- no. of courses
    S <- no. of slots
    current course <- 0
    for all the slots
    {
        if slot is empty
        {
            if next slot is not empty
            {
                while all the courses are not finished & time alloted to current course
                    is less than time available in the present slot
                {
                    time alloted in current slot <- time in current course + existing
                    so to next course
                }
                // continue till slot is full
            }
            if next slot is empty
            {
                do the same thing as above but time alloted is 25 min per slot
            }
        }
    }
}
```

**Listing 1:** Numerical computation of the model

```
for i ∈ [ 0,1,...,quizdayA ]
    p_i^A(new) = p_i^A(without quiz)+ (f_i / Σf_i) t_0^A
    //where Σf_i is till quiz day of course A
for i ∈ [ quizdayA,....,7 ]
    p_i^A(new) = p_i^A(without quiz)−t_0^A/R_A
```

**Listing 2:** Pseudo code for quiz

# Algorithmic Implementation

For simulation of the model, we wrote the following computer program (in Python). Visit:
https://github.com/rohitkrbose/Time-Table-Automation
Download all the files. Edit input files, and execute Scheduler.py from the terminal. A new file will be created, which is the proposed time table.

CODE

```
import numpy as np
import pandas as pd
import csv as csv

# Maps each day to a particular number
def getDayMap(s):
  if (s=="Sunday"):
    return 0
  if (s=="Monday"):
    return 1
  if (s=="Tuesday"):
    return 2
  if (s=="Wednesday"):
    return 3
  if (s=="Thursday"):
    return 4
  if (s=="Friday"):
    return 5
  if (s=="Saturday"):
    return 6
  return 7

# Reads EXISTING TIME TABLE from csv file and stores it as a matrix
def getTimeTable ():
  T = pd.read_csv("Time Table.csv",sep=",").fillna(0).as_matrix()[0:,1:]
  time_slots = T.shape[0]
  days = T.shape[1]
  return (T,time_slots,days)

# Reads QUIZ info from csv file and stores it in a matrix
def getQuizList():
  quizList = pd.read_csv("Quiz List.csv",sep=",").fillna(0).as_matrix()
  for i in range (0,quizList.shape[0]):
    quizList[i][2] = getDayMap(quizList[i][2])
  return quizList

# Reads COURSE info from csv file and stores it in a matrix
def getCourseList():
  return (pd.read_csv("Course List.csv",sep=",").as_matrix())

# Stores free time per day in an array
def getFreeTime (T):
  freetime = np.zeros(7) # stores number of hours free in a days
  hh = 0.0
  for j in range (0,days):
    for i in range (0,time_slots):
      if (T[i][j]!=0):
        continue
      hh = hh + 1
    freetime[j] = float(hh/2)
    hh=0.0
  return freetime
```

```
53
54 # Allots time to be devoted to each course on each day on the basis of credits,
       confidence, free time, and presence of quiz
55 def studyTimePerDay (courseList, quizList, freetime):
56   M = courseList.shape[0] # number of courses
57   p = np.zeros((M,7)) # study time of each course on each day
58   p_orig = np.zeros((M,7))
59   qt = np.zeros(M) # extra time to be studied for quiz
60   F = np.zeros(M) # free time till that quiz day (incl.)
61   R = np.zeros(M) # remaining number of days after quiz
62   for i in range (0,M):
63     for j in range (0,7):
64       p_orig[i][j] = float(courseList[i][1])/courseList[i][2]*freetime[j]/np.sum(freetime
           )
65       p[i][j] = p_orig[i][j]
66   # calculate net extra time to be studied for quiz per course
67   for i in range (0,M):
68     qt[i] = 4/float(courseList[i][2])*quizList[i][1]*courseList[i][1]/100
69   # calculate remaining number of days after a quiz, remaining days include the day of
       the quiz as well
70   for i in range (0,M):
71     if (quizList[i][2] < 7):
72       R[i] = 7 - quizList[i][2]
73     else:
74       R[i] = 7
75   # net free time (on all days) TILL each quiz
76   for i in range (0,M):
77     F[i] = np.sum(freetime[0:quizList[i][2]])
78   # do the job:
79   for i in range (0,M):
80     if (quizList[i][2] != 7):
81       for j in range (0,quizList[i][2]):
82         p[i][j] = p[i][j] + freetime[j]*qt[i]/F[i]
83       for j in range (quizList[i][2],7):
84         p[i][j] = p_orig[i][j] - qt[i]/R[i]
85   for i in range (0,M):
86     for j in range (0,7):
87       if (p[i][j] < 0):
88         p[i][j] = 0
89   return p
90
91 # Ranks courses to be studied in each day in order of the number of hours devoted to it
       using bubble sort
92 def returnCourseOrder (p,courseList):
93   M = courseList.shape[0]
94   p_mat = p
95   courses = courseList[:,0]
96   course_mat = np.empty((M,7),dtype='a7')
97   for j in range (0,7):
98     course_mat[:,j] = courses
99   for i in range (0,7):
100    for j in range (0,M):
101      for k in range (0,M-j-1):
102        if (p_mat[k][i]>p_mat[k+1][i]):
103          p_mat[k][i],p_mat[k+1][i] = p_mat[k+1][i],p_mat[k][i]
104          course_mat[k][i],course_mat[k+1][i] = course_mat[k+1][i],course_mat[k][i]
105  return course_mat,p_mat
106
107 # Fit time to be studied in empty slots in the EXISTING TIME TABLE
```

```
def makeDay (T, course_mat, p_mat):
  T_copy = np.copy(T)
  T = np.vstack([T,[0,0,0,0,0,0,0]]) # add dummy row for calculation purpose
  T_text = np.copy(T)
  slot_count = T.shape[0] # number of slots on a day
  course_count = course_mat.shape[0] # number of courses
  current_course = 0
  for j in range (0,7):
    day = T[:,j] # slots
    ttt = T_text[:,j]
    for k in range (0,slot_count):
      if (ttt[k]==0):
        ttt[k] = ""
    course_order = course_mat[:,j]
    p_order = p_mat[:,j]*60.0 # converting to minutes
    current_course = 0
    for i in range (0,slot_count):
      if (day[i] != 0): # if slot is not empty
        continue
      if (i+1 < slot_count):
        if (day[i+1] != 0): # if next slot is not empty
          # continue filling till slot is full
          while (current_course < course_count and p_order[current_course]<=20-day[i]):
            day[i] = day[i] + p_order[current_course]
            x = int(round(p_order[current_course],0))
            if (x>0):
              ttt[i] = ttt[i] + str(course_order[current_course]) + ":" + str(x) + " "
            current_course = current_course + 1
          # if course study time takes up entire time of the slot
          if (current_course < course_count and p_order[current_course] > 20-day[i]):
            p_order[current_course] = p_order[current_course] - (20-day[i])
            x = int(round(20-day[i],0))
            day[i] = 20
            if (x>0):
              ttt[i] = ttt[i] + str(course_order[current_course]) + ":" +  str(x) + " "
        else: # if next slot is empty
          # continue filling till slot is full
          while (current_course < course_count and p_order[current_course] <= 25-day[i]):
            day[i] = day[i] + p_order[current_course]
            x = int(round(p_order[current_course],0))
            if (x>0):
              ttt[i] = str(course_order[current_course]) + ":" +  str(x) + " "
            current_course = current_course + 1
          # if course study time takes up entire time of the slot
          if (current_course < course_count and p_order[current_course] > 25-day[i]):
            p_order[current_course] = p_order[current_course] - (25-day[i])
            x = int(round(25-day[i],0))
            day[i] = 25
            if (x>0):
              ttt[i] = ttt[i] + str(course_order[current_course])  + ":" +  str(x) + " "
    T_text[:,j] = ttt
  # formatting
  T_text = np.delete(T_text,slot_count-1,0)
  for j in range (0,7):
    for i in range (0,slot_count-1):
      if (T_text[i][j] == 0):
        T_text[i][j] = ''
  return T_text

```

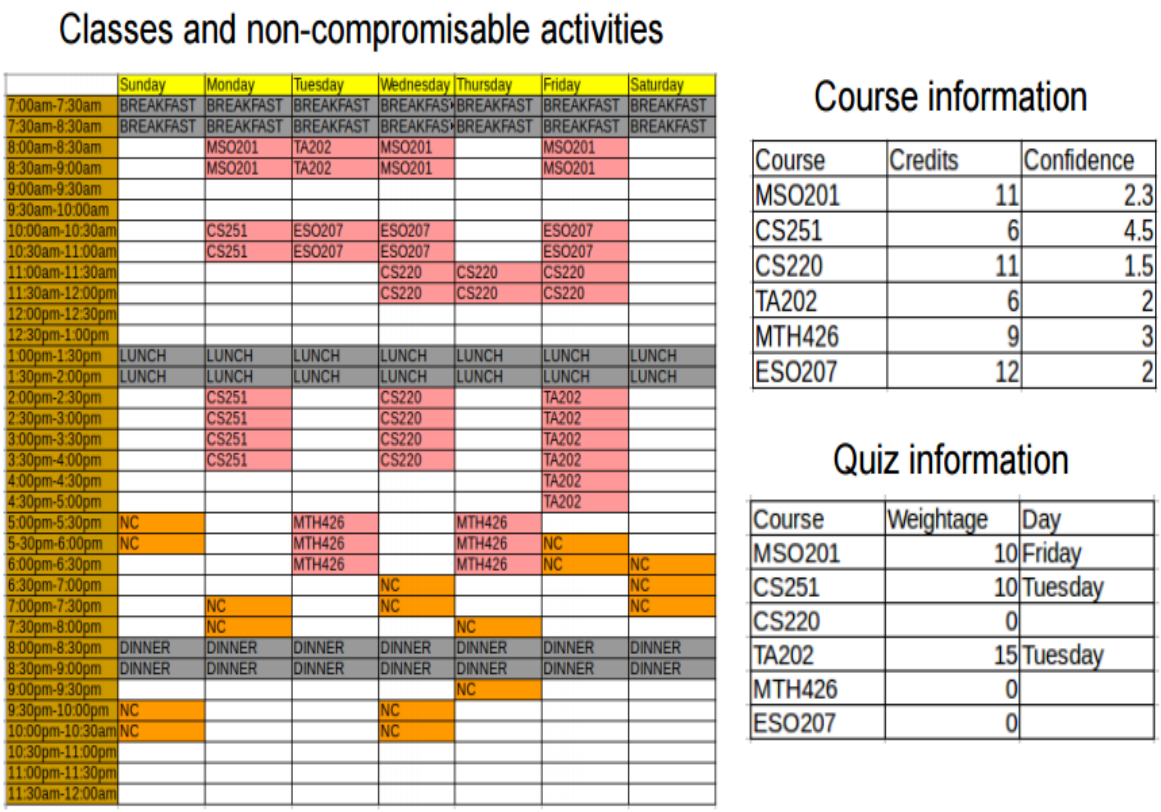
```
# Formats the output
def outputTimeTable (T_text):
  T_temp = pd.read_csv("Time Table.csv",sep=",",header=None).fillna(0).as_matrix()
  T_temp[1:T_text.shape[0]+1,1:8] = T_text
  T_temp[0,0] = ''
  with open('Proposed Time Table.csv', 'wb') as f:
    csv.writer(f).writerows(T_temp)

# Run code:
T, time_slots, days = getTimeTable()
freetime = getFreeTime(T)
courseList = getCourseList()
quizList = getQuizList()
p = studyTimePerDay(courseList,quizList,freetime)
course_mat, p_mat = returnCourseOrder(p,courseList)
T_text = makeDay(T,course_mat,p_mat)
outputTimeTable(T_text)
```

**Listing 3:** Main Code

# Analysis

**Figure 6:** Input



Classes and non-compromisable activities

| | Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday |
|---|---|---|---|---|---|---|---|
| 7:00am-7:30am | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST |
| 7:30am-8:30am | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST |
| 8:00am-8:30am | | MSO201 | TA202 | MSO201 | | MSO201 | |
| 8:30am-9:00am | | MSO201 | TA202 | MSO201 | | MSO201 | |
| 9:00am-9:30am | | | | | | | |
| 9:30am-10:00am | | | | | | | |
| 10:00am-10:30am | | CS251 | ESO207 | ESO207 | | ESO207 | |
| 10:30am-11:00am | | CS251 | ESO207 | ESO207 | | ESO207 | |
| 11:00am-11:30am | | | | CS220 | CS220 | CS220 | |
| 11:30am-12:00pm | | | | CS220 | CS220 | CS220 | |
| 12:00pm-12:30pm | | | | | | | |
| 12:30pm-1:00pm | | | | | | | |
| 1:00pm-1:30pm | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH |
| 1:30pm-2:00pm | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH |
| 2:00pm-2:30pm | | CS251 | | CS220 | | TA202 | |
| 2:30pm-3:00pm | | CS251 | | CS220 | | TA202 | |
| 3:00pm-3:30pm | | CS251 | | CS220 | | TA202 | |
| 3:30pm-4:00pm | | CS251 | | CS220 | | TA202 | |
| 4:00pm-4:30pm | | | | | | TA202 | |
| 4:30pm-5:00pm | | | | | | TA202 | |
| 5:00pm-5:30pm | NC | | MTH426 | | MTH426 | | |
| 5-30pm-6:00pm | NC | | MTH426 | | MTH426 | NC | |
| 6:00pm-6:30pm | | | MTH426 | | MTH426 | NC | NC |
| 6:30pm-7:00pm | | | | NC | | | NC |
| 7:00pm-7:30pm | | NC | | NC | | | NC |
| 7:30pm-8:00pm | | NC | | | NC | | |
| 8:00pm-8:30pm | DINNER | DINNER | DINNER | DINNER | DINNER | DINNER | DINNER |
| 8:30pm-9:00pm | DINNER | DINNER | DINNER | DINNER | DINNER | DINNER | DINNER |
| 9:00pm-9:30pm | | | | | NC | | |
| 9:30pm-10:00pm | NC | | | NC | | | |
| 10:00pm-10:30am | NC | | | NC | | | |
| 10:30pm-11:00pm | | | | | | | |
| 11:00pm-11:30pm | | | | | | | |
| 11:30am-12:00am | | | | | | | |

Course information

| Course | Credits | Confidence |
|---|---|---|
| MSO201 | 11 | 2.3 |
| CS251 | 6 | 4.5 |
| CS220 | 11 | 1.5 |
| TA202 | 6 | 2 |
| MTH426 | 9 | 3 |
| ESO207 | 12 | 2 |

Quiz information

| Course | Weightage | Day |
|---|---|---|
| MSO201 | 10 | Friday |
| CS251 | 10 | Tuesday |
| CS220 | 0 | |
| TA202 | 15 | Tuesday |
| MTH426 | 0 | |
| ESO207 | 0 | |

Note that the course ESO207 has maximum credits and the confidence in that course of the student is pretty low. So in spite of the other quizzes, the student is supposed to study this course in a considerable amount. And the courses having no quizzes are distributed according to their credits and confidence.

Further, let's suppose we have 3 quizzes, 2 on Tuesday of CS251 and TA202 and one of MSO201 on Friday. As we can see from input that the students confidence is very high in CS251 and the number of credits of this course is also quite low, therefore he is studying this course for a very short amount of time in the days prior to the quiz and it nearly disappears from the timetable afterwards.

Coming to the next quiz of TA202 on Tuesday, this course also has 6 credits but the confidence in this course is really low. Thats why the student is studying this course for a larger amount of time on both Sunday and Monday. After this, we come to the quiz of MSO201 which is on Friday and as we can see that the confidence in this course is very low and the number of credits of this course is very high. All these factors lead to a large amount of study time for MSO201 till Friday.

**Figure 7:** Output

## Personalized time-table showing study time distribution

| | Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday |
|---|---|---|---|---|---|---|---|
| 7:00am-7:30am | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST |
| 7:30am-8:30am | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST | BREAKFAST |
| 8:00am-8:30am | MTH426:25 | MSO201 | TA202 | MSO201 | TA202:6 MTH426:13 | MSO201 | TA202:11 MTH426:6 |
| 8:30am-9:00am | MTH426:7 CS251:18 | MSO201 | TA202 | MSO201 | MTH426:14 ESO207:11 | MSO201 | MTH426:25 |
| 9:00am-9:30am | CS251:14 ESO207:11 | MTH426:24 CS251:1 | TA202:6 MTH426:13 | MTH426:18 ESO207:5 | ESO207:25 | MTH426:18 ESO207:5 | MTH426:2 ESO207:23 |
| 9:30am-10:00am | ESO207:25 | CS251:20 | MTH426:14 ESO207:6 | ESO207:20 | ESO207:20 CS220:5 | ESO207:20 | ESO207:25 |
| 10:00am-10:30am | ESO207:25 | CS251 | ESO207 | ESO207 | CS220:25 | ESO207 | ESO207:18 CS220:7 |
| 10:30am-11:00am | ESO207:2 CS220:23 | CS251 | ESO207 | ESO207 | CS220:20 | ESO207 | CS220:25 |
| 11:00am-11:30am | CS220:25 | CS251:3 ESO207:22 | ESO207:25 | CS220 | CS220 | CS220 | CS220:25 |
| 11:30am-12:00pm | CS220:25 | ESO207:25 | ESO207:25 | CS220 | CS220 | CS220 | CS220:23 |
| 12:00pm-12:30pm | CS220:4 MSO201:21 | CS220:25 | CS220:25 | ESO207:12 CS220:13 | CS220:17 MSO201:8 | ESO207:12 CS220:13 | |
| 12:30pm-1:00pm | MSO201:20 | CS220:20 | CS220:20 | CS220:20 | MSO201:20 | CS220:20 | |
| 1:00pm-1:30pm | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH |
| 1:30pm-2:00pm | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH | LUNCH |
| 2:00pm-2:30pm | MSO201:25 | CS251 | CS220:22 MSO201:3 | CS220 | MSO201:25 | TA202 | |
| 2:30pm-3:00pm | MSO201:12 TA202:13 | CS251 | MSO201:25 | CS220 | MSO201:16 | TA202 | |
| 3:00pm-3:30pm | TA202:25 | CS251 | MSO201:25 | CS220 | | TA202 | |
| 3:30pm-4:00pm | TA202:25 | CS251 | MSO201:16 | CS220 | | TA202 | |
| 4:00pm-4:30pm | TA202:25 | CS220:13 MSO201:12 | | CS220:12 MSO201:13 | | TA202 | |
| 4:30pm-5:00pm | TA202:6 | MSO201:25 | | MSO201:25 | | TA202 | |
| 5:00pm-5:30pm | NC | MSO201:22 TA202:3 | MTH426 | MSO201:8 | MTH426 | CS220:12 | |
| 5-30pm-6:00pm | NC | TA202:25 | MTH426 | | MTH426 | NC | |
| 6:00pm-6:30pm | | TA202:25 | MTH426 | | MTH426 | NC | NC |
| 6:30pm-7:00pm | | TA202:17 | | NC | | | NC |
| 7:00pm-7:30pm | | NC | | NC | | | NC |
| 7:30pm-8:00pm | | NC | | | NC | | |
| 8:00pm-8:30pm | DINNER | DINNER | DINNER | DINNER | DINNER | DINNER | DINNER |
| 8:30pm-9:00pm | DINNER | DINNER | DINNER | DINNER | DINNER | DINNER | DINNER |
| 9:00pm-9:30pm | | | | | NC | | |
| 9:30pm-10:00pm | NC | | | NC | | | |
| 10:00pm-10:30am | NC | | | NC | | | |
| 10:30pm-11:00pm | | | | | | | |
| 11:00pm-11:30pm | | | | | | | |
| 11:30am-12:00am | | | | | | | |

In this figure the boxes in Pink correspond to the original timetable and the one in Orange represent the Non Compromisable activities and the boxes in fluorescent green represent the time slots in which the student is supposed to study for the mentioned time.

# Results and Discussions

The model we have proposed is a novel way to determine the schedule of a student. Since our model determines a personalized time-table for every student, it is not possible to validate it, as variables vary way too much from person to person. However, we have tried to ensure that our model takes into account the fact that students are humans, and included breaks of 5 and 10 minutes in between study, as per the widely known Pomodoro technique. Moreover, we also took into account the fact that concentration for normal students fall off after about 4 hours of study, so have not allowed any student to read for more than 4 hours at a stretch. This was reinforced by the survey that we conducted.

# Conclusion and Future Scope

## Conclusion

We created a model that generates the time table of a student that is unique for each student as it depends on his/her courses and extra-curricular activities and confidence in each course. We have also tried to take into consideration the relevant factors. This can be made much more efficient if we take some more factors into account such as concentration.

## Future Scope

There are lots of future prospects available with this model. We may pair up this algorithm with the Office Automation System of IIT kanpur and create a portal for the students in which they will enter their roll number and answer the questions related to confidence and concentration and our portal will output his/her time table. We have taken some assumptions into our model which if removed can make our time table much more efficient and realistic. Some examples are as follows:

- Restriction on the number of quizzes.
- Taking concentration and confidence to be mutually exclusive quantities.
- Also the questionnaire for confidence can be improved as well.

For now, the model incorporates only IIT Kanpur system, we may develop it further for all university and school students around the world. This is completely a new model and no one has ever researched on generating such an algorithm as per our knowledge.

# References

[1] Sayed, S. Ahmed, A. Aamir, A. and Zaeem, A. *Automated Timetable Generator*. International Journal for Innovative Research in Science & Technology, 1(11):118-121, 2015.

[2] Wang, X. Gobbo, F. and Lane, M. *Turning Time from Enemy into an Ally Using the Pomodoro Technique*. [Agility Across Time & Space] Springer-Verlag, Chapter:149-166, 2010.

[3] Mittal, D. Doshi, H. Sunasra, M. and Nagpure, R. *Automatic Timetable Generation using Genetic Algorithm*. International Journal of Advanced Research in Computer & Communication Engineering, 4(2):245-248, 2015.

[4] Nanda, A. Pai, Manisha P. and Gole, A. *An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach*. International Journal of Machine Learning and Computing, 2(4):492-495, 2012.

[5] Willemen, R.J. *School timetable construction : algorithms and complexity*. Universiteitsdrukkerij Technische Universiteit Eindhoven, 2002.