# Optimised Time-table Generator

R.Basuhi                                      140010037
Shachi Shailesh Deshpande        140110047
Yashaswini K Murthy                  140010054

# Problem Statement

Given, input by professors and TAs as to their preferred days and slots for teaching a particular course, generate an (optimised) time-table that'll make most of them happy :)

# Goals:

1. Taking in the preferences of Teachers of all courses(i.e preferred day and slot)
2. Using preference values of the teachers to construct objective function and optimising it.
3. Generating optimised time-table.
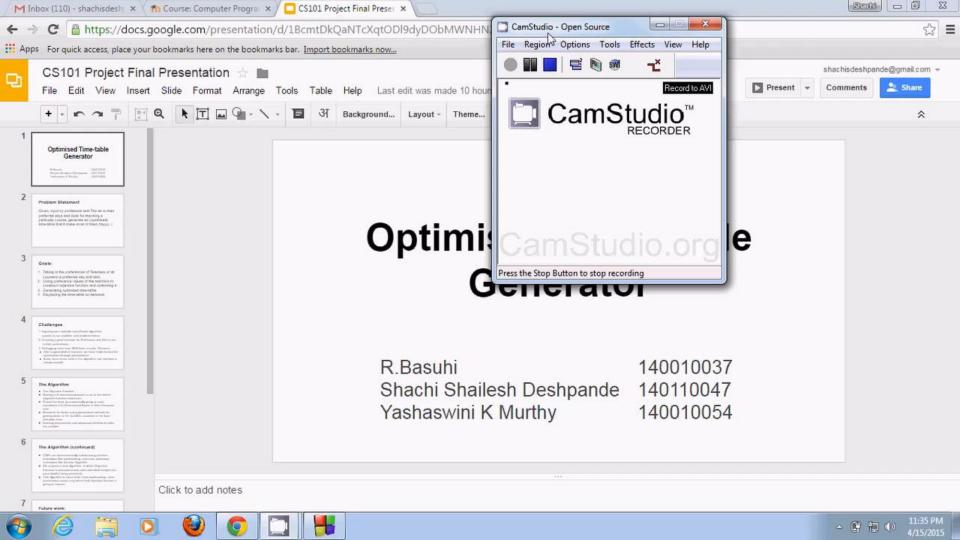4. Displaying the time-table on demand.

# Challenges

1. Figuring out a suitable and efficient algorithm specific to our problem, and implementation.
2. Creating a good interface for Professors and TA's to put in their preferences.
3. Debugging more than 2000 lines of code. Ofcourse.

● After a good deal of research, we have implemented the optimisation through perturbations.
● Since heart of our code is the algorithm, our interface is simple console

# The Algorithm
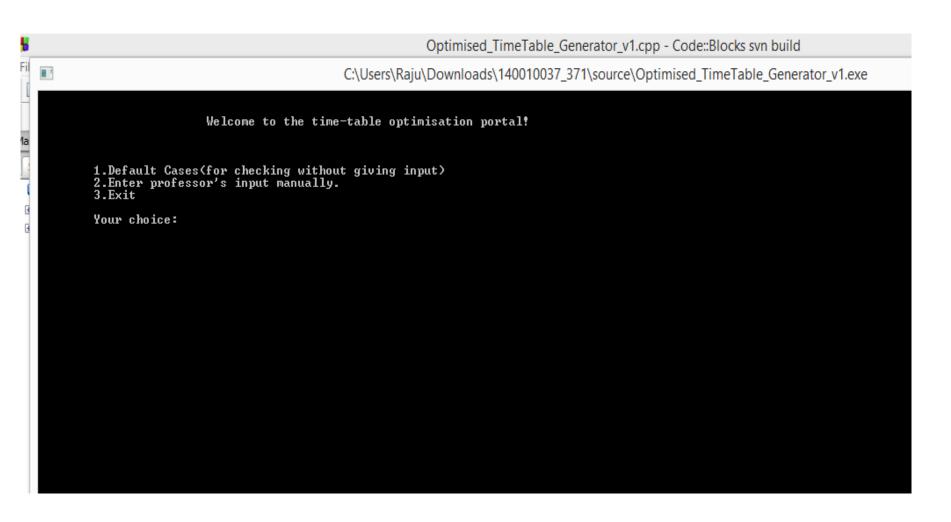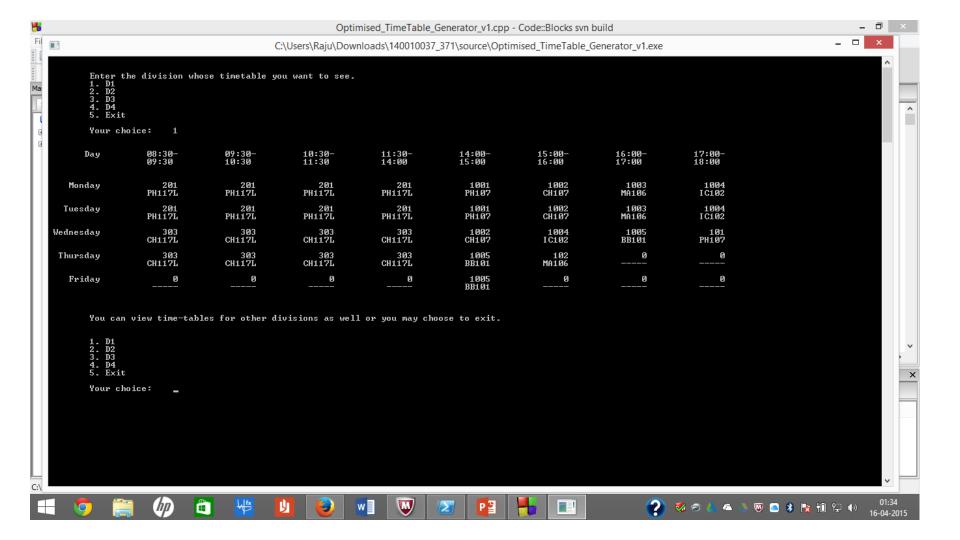
★ The Objective Function
★ Moving in N-dimensional space so as to see where objective function maximises.
★ Cannot be done by sequentially going to each coordinate in N-Dimensional Space in finite Computer time.
★ Research for better and sophisticated methods for getting closer to the GLOBAL maximas in the least computer time.
★ Existing deterministic and advanced methods to solve the problem.

# The Algorithm (continued)

★ CSPs are deterministically solved using primitive techniques like backtracking, and more advanced techniques like Genetic Algorithm.

★ We propose a new algorithm, in which Objective Function is associated with each individual entity(in our case faculty) being perturbed.

★ This algorithm is much better than backtracking, since perturbation occurs only when total objective function is going to improve.

Optimised Time-table Generator

R.Basuhi                                          140010037
Shachi Shailesh Deshpande        140110047
Yashaswini K Murthy                     140010054

C:\Users\Raju\Downloads\140010037_371\source\Optimised_TimeTable_Generator_v1.exe

```
           Welcome to the time-table optimisation portal!


1.Default Cases(for checking without giving input)
2.Enter professor's input manually.
3.Exit

Your choice:
```

```
Enter the division whose timetable you want to see.
1. D1
2. D2
3. D3
4. D4
5. Exit

Your choice:     1
```

| Day | 08:30-09:30 | 09:30-10:30 | 10:30-11:30 | 11:30-14:00 | 14:00-15:00 | 15:00-16:00 | 16:00-17:00 | 17:00-18:00 |
|---|---|---|---|---|---|---|---|---|
| Monday | 201 PH117L | 201 PH117L | 201 PH117L | 201 PH117L | 1001 PH107 | 1002 CH107 | 1003 MA106 | 1004 IC102 |
| Tuesday | 201 PH117L | 201 PH117L | 201 PH117L | 201 PH117L | 1001 PH107 | 1002 CH107 | 1003 MA106 | 1004 IC102 |
| Wednesday | 303 CH117L | 303 CH117L | 303 CH117L | 303 CH117L | 1002 CH107 | 1004 IC102 | 1005 BB101 | 101 PH107 |
| Thursday | 303 CH117L | 303 CH117L | 303 CH117L | 303 CH117L | 1005 BB101 | 102 MA106 | 0 ----- | 0 ----- |
| Friday | 0 ----- | 0 ----- | 0 ----- | 0 ----- | 1005 BB101 | 0 ----- | 0 ----- | 0 ----- |

```
You can view time-tables for other divisions as well or you may choose to exit.

1. D1
2. D2
3. D3
4. D4
5. Exit

Your choice:     _
```

# Future work:

★ Making mathematical computation of computer time and resources required for the algorithm, so as to compare it rigorously with available methods of solving CSPs.
★ Improving and modifying the algorithm so that it can be used for a bigger and more practical problem of time-table generation, and some other kinds of problems using huge collection of variables.
★ Aesthetic addition: including a Graphic User Interface for input.

# Future Work (continued)

★ Expanding current problem to more number of courses, to practically cover all the courses of the Institute.

★ Including some other important practical variables, like variable number of subjects a single Professor teaches, variable number of students registered for different courses, variable sizes of classrooms available at a time,etc.

★ Managing some other practical constraints like distances between different Lecture Halls the Professor and Students have to travel to, between 2 successive lectures,etc.