# Solving University Course Scheduling Problem Using Genetic Algorithm and analyzing results with Other Algorithms

**Authors**

**Md. Ashraful Alam 12101108**

**Toushika Islam 11101002**

**Duke Joseph 10201038**

**Supervised By**

**Moin Mostakim**

A thesis presented for the degree of B.Sc in Computer Science And Engineering

Department of Computer Science and Engineering

BRAC University

Bangladesh

# Abstract

**Solving University Course Scheduling Problem Using Genetic Algorithm and analyzing results with Other Algorithms**

A study on course timetabling problem which is a combinatorial optimization NP-hard problem. The aim of this thesis is to find optimal or near optimal solution of course scheduling for Computer Science And Engineering Department Of BRAC University. Different solution methods for course timetabling exists hence in this thesis Genetic Algorithms is used to generate feasible solution and Q-learning is action for evaluating results. Experimental data sets are parsed from a given structure. Different constraints are handled with discrete fitness evaluation. Schedule conflicts are handled after producing random generation. Finally, results are tested according to their performance and presented with a feasible representation mode.

# Dedication

To our beloved parents for their love

To our respected teachers for direct and indirect help through our work

To BRAC University for which this thesis became possible.

# Declaration

We, hereby declare that the results of this thesis are obtained by our own work and implementation. Resources and materials of this thesis found and develop by other researchers are carefully mentioned in reference citation. This thesis has not been published or presented in parts or as a whole for any Degree.

Authors Signature

.......................................

Md. Ashraful Alam

.......................................

Toushika Islam

.......................................

Duke Joseph

Supervisor's Signature

.......................................

Moin Mostakim

# Acknowledgment

We wish to express our sincere gratitude to our honorable supervisor, Mr. Moin Mostakim Sir for giving us the opportunity to work with him as well as guiding and tolerating some of our premature thoughts through out the journey. Furthermore, We would like to thank Mrs. Farzana Rashid mam who suggested us to work on this topic and provided us basic progressive thoughts and strategies until she left for PhD. Our special thanks to Md. Shamsul Kaonaine sir for guidance and encouragement in carrying out this thesis. We also express our gratitude to the official and other staff members Of BRAC University who rendered their help during the period of our thesis. Best regards from our heart goes to our parents who have been providing greatest support not only to carry on the this thesis but also throughout our life. Finally, We the participants of our thesis, Md. Ashraful Alam, Toushika Islam and Duke Joseph also like to thank each other to carry out the thesis with harmony.

# Contents

# Chapter 1

# Introduction

An inevitable problem that every University has to solve by any mean is to schedule its courses. More specifically, individual departments are responsible for doing this task. It is very much time consuming to meet all the constraints while making the schedule manually. Automated timetabling is a task to provide optimal or near optimal solutions within a short period of time ensuring a quality schedule using different optimization techniques.

The problem can be defined as a task where a number of University courses' related events are to be allocated in limited resources under a set of constraints. Violation of the constraints should be kept as minimum as possible even though it is considered and proved as NP-complete problem.

Different types of scheduling problem are described in literature which are very much similar to this problem. Examination scheduling, lab scheduling, job scheduling e.t.c. shows similar behavior in functionality and complexity. Each problem varies along with their constraints.

## 1.1 Objectives

In this thesis paper different semester's course schedules from BRAC University CSE department are observed, studied and analyzed for defining patterns and assemble data. Genetic algorithm is used to provide solution under constraints and Q learning is implemented to check whether the solution is fair enough or not. So in short the objective of this thesis are:

- Collection and processing of raw data from different sources.
- Analysis and group them into different sets of data according to the criteria they need to be sorted.
- Implementation of Genetic Algorithm to generate solution
- Evaluate fitness and continue until fitness is acceptable.
- validation and comparison of obtained results.

# Chapter 2

# Scheduling

Scheduling refers to the set of rules and mechanisms to maintain the order of our everyday work.Different institutions have different policies to make up their own schedule based on their requirements. On the contrary, in our everyday life from morning till evening, we all have to go through a certain schedule to perform our work orderly. So, it is inevitable issue to all. A timetable is such a model that could be educational, transport, job, sports or even anything which needs to be satisfied under certain constraints during allocation.

According to, Anthony Wren(1996)[22] scheduling can be defined "Time tabling is the allocation, subject to constraints, of given resources of objects being placed in space-time, in such a way as to satisfy as nearly as possible a set of desirable objectives." [2]

As being an NP-complete problem, no method is able to solve it avoiding polynomial amount of time.[7][2]

## 2.1 Non-Academic Scheduling

There are several tasks which are very important to be organized in the parameter of time. Any kind of violation could be disasters in this type of scheduling. Economic,social even life threatening situation could be arisen. Some of these kinds of scheduling are:

- Airport Scheduling
- Sports Scheduling
- Traffic Scheduling
- Employee Scheduling

## 2.2 Academic Scheduling

Different types of models can be found in use of academic scheduling which behaves several ways in their characteristics. Most of the models are handling academic scheduling chaos. Most common models for academic scheduling are :

- **School Scheduling** is one of the most important phenomenon of educational structure. To manage the resources of school ,the authority provides an optimal schedule maintaining the constraints. Basically,every year due to change of students number,number of class room, teachers number the schedule might have changed. The authority's main focus is to provide an optimal schedule for the school by analyzing it's previous data.
- **College Scheduling** is similar to the school scheduling. But, over here the

management have to be more careful to generate this schedule. Rather than school, in college there are more subjects, extra curricular activities, the management gets into trouble to fix up an optimal routine . On the other hand, each college subject consists of two parts and each part has it's own lab as well as theory class. So, the management usually try to complete one part in a year and another part in next year. As a result they have to provide at least two different routine to perform this task.

- **University Scheduling** is another major scheduling on regarding platform. Different Universities have their own way to provide their own routines. Some University Provide yearly routines and few university provides semester routine(consists of consecutive few months) depending on the resources of their curriculum.

- **Exam Scheduling** topic comes at the end of a semester or a term ; so that, the authority can know about students progress. From it's they can also take decision where to progress and also think about the improvement of their providing routine so that student can make the best use of that schedule.For this reason, all the educational institutions generates their own schedule by considering student,instructors and management's suitable times.

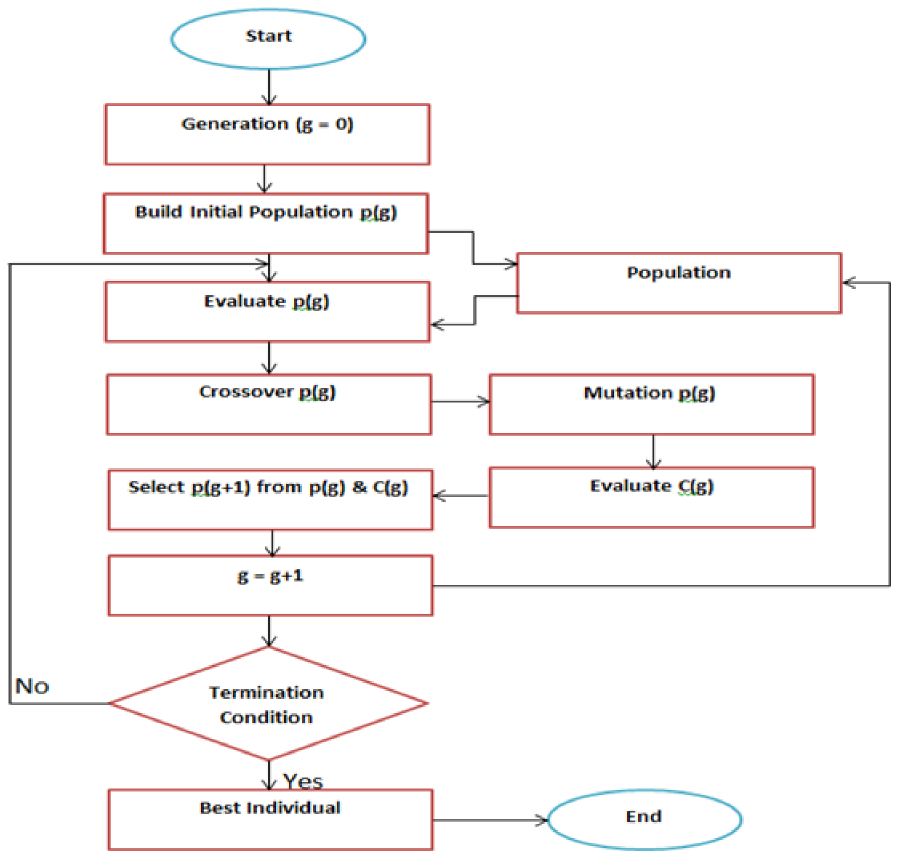
- **Lab Scheduling** is a part of curriculum design. Without hindering other schedule the authority also provide a schedule for performing lab work.Every educational institutions have their own lab schedule according to their willing. When they generates this routine they mainly focus on to manage atleast one day to perform that lab work.

# Chapter 3

# Algorithms

## 3.1 Genetic Algorithm

Computational models belong to evolutionary Algorithms class as well Artificial Intelligence. A Function optimizer works on basis of natural evolution like Selection, Crossover, Mutation, and Inheritance which is also used solving complex optimization problems comparatively for bigger search spaces. A Genetic Algorithm has many steps maintaining few consequences. First of , it generates a random population using different parametric data and condition. At the next step, it evaluates the fitness and select them for further steps. Finally, It(GA) manipulates some of it's offspring for better fitness if needed.



### 3.1.1 Selection

Reproduction (or selection) is an operator that makes more copies of better strings in a new population. Reproduction is usually the first operator applied on a population. During each successive generation, a proportion of existing population is selected to breed a new generation. Individual solutions are selected through fitness-based process, where fitter solutions are typically more likely to be selected. Selection methods rate the fitness of each individual and preferentially select the best solution.

**Common Methods of Selection are:**

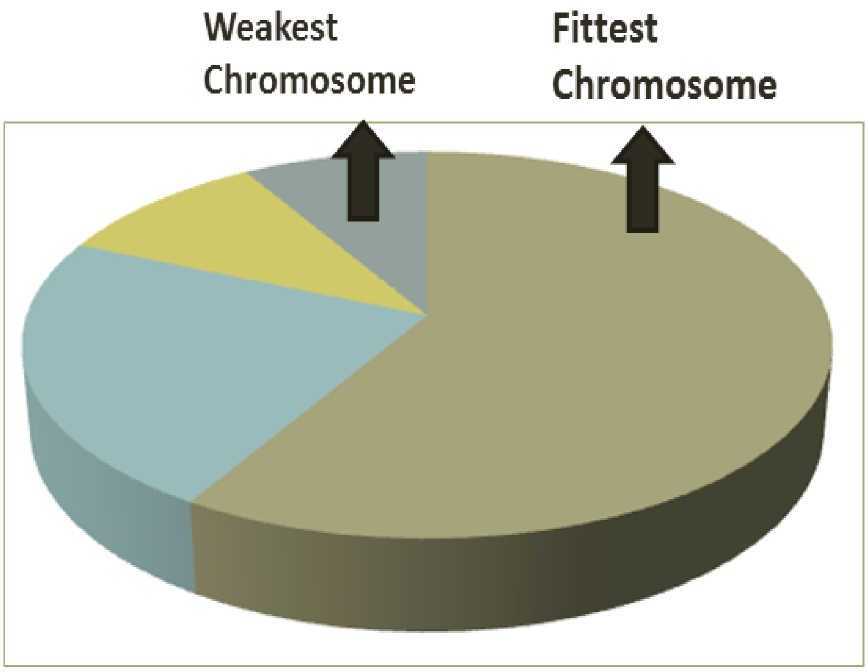
- Roulette wheel method.
- Tournament selection.
- Stochastic remainder selection.

We are using Roulette wheel for our research work.

**Roulette wheel Selection:**

The $i_t h$ string in the population is selected with a probability proportional to $f_i$.Since the population size is usually kept fixed in a simple Genetic Algorithm, thus the sum of the probability of each string being selected for the mating pools must be one. Therefore, the probability for selecting the string is [8]

$$P_i = \frac{f_i}{\sum_{i=1}^{n} f_i}$$



**Roulette wheel pseudo code:**

```
for all members of population
          sum += fitness of this individual
          end for
                    for all members of population
                              probability = sum of probabilities +
                              (fitness / sum)
                              sum of probabilities += probability
                    end for
          loop until new population is full
                    do this twice
                              number = Random between 0 and 1
                              for all members of population
                                        if number > probability
                                        but less than next probability
                                        then you have been selected
                              end for
                    end
                    create offspring
          end loop
```

### 3.1.2 Crossover

A crossover operator is used to recombine two strings to get a better string. The next step after selection is crossover which generates a second generation population of solutions from those selected through selection.
Some Common types of Crossover:[19] [8]

- One site crossover

- Two site crossover
- Trade of Uniform Crossover(TOUC).

Currently, in our research work we are using one site crossover. Crossover is advantageous over the conventional method because through crossover we can easily exchange the information in the timetable which makes it optimal and effective as per the requirements.

### 3.1.3 Mutation

Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima. It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeat use of reproduction and crossover operators.[15][13]

## 3.2 Simulated Annealing

### 3.2.1 Background From Physics

From the solid state physics, we first come to know about annealing process from where the simulated annealing algorithm is introduced. First, a solid gets high temperature and then again it comes back to it's normal state. The solid particles achieve higher mobility due to increase higher temperature. Different locations can be reached around the solids by the particles because the temperature gets lower slow enough. There is term called Metropolis loop which indicates a repeating procedure which runs till the thermal equilibrium is achieved.

In the solid state physics, solid matters have characteristics to be at energy state $E$ with temperature $T$. The probability of remaining at state $E$ is [2]

$$P(E) = (\frac{1}{Z(T)} * e^{\frac{e_s}{K_Z * T}})$$

Here, $K_Z$ is Boltzmann constant and the temperature dependent normalization factor is $Z(T)$.

### 3.2.2 SA Pseudo code

In pseudo code Simulated annealing can be written as [2]

```
⇒ Random initial solution $S = S_0$
⇒ Select an initial Temperature $T = T_0 > 0$
⇒ Select a Temperature reduction rate $\alpha$
⇒ **loop**
⇒   **loop**
⇒     $S' =$ NeighborhoodSearching($S$)
⇒     $\delta = F(S') - F(S)$
⇒     **if**($\delta <= 0$ or exp($-\frac{\delta}{T} < rand[0,1]$))
⇒     $S = S'$
⇒     **end if**
⇒   **until** Temperature normal
⇒   T = CoolingSchedule(t)
⇒ **until** stop condition = true
```

### 3.2.3 Neighborhood Search

One of the key components of simulated annealing method is neighborhood searching which performs to find out next possible solution set.

- Firstly,it search for randomly chosen neighbors as solution. This part is called simple searching neighborhood.
- Secondly, the algorithm goes for swapping the neighbors as per needed.
- Finally, the method runs with the searching and the swapping both.

### 3.2.4 Cost Calculation

Cost calculation depends on problem definition, as the problems varies cost calculation parameters also changes. For scheduling purpose, we consider cost in three types:[2]

- Search Cost
- Swap Cost
- Constraint Cost

#### Search Cost

In neighborhood structure, for each solution set different types of complexity comes up like data structures and run time complexity. This leads us the search cost.

#### Swap Cost

When a huge number of solution set needs to be swapped among neighborhood, it charges a cost function and get added to the total cost.

**Constraint Cost** [2]

Several types of constraints we face in scheduling problem and each constraint allows cost due to violation. If time slot constraint is violated then the cost

$$F_1 = w_1 \sum_{i=1}^{n} T_i$$

For Instructor assignment violation,

$$F_2 = w_2 \sum_{i=1}^{m} I_i$$

For Course violation,

$$F_3 = w_3 \sum_{i=1}^{r} C_i$$

For slot violation,

$$F_4 = w_4 \sum_{i=1}^{p} S_i$$

For soft constraint violation,

$$F_5 = w_5 \sum_{i=1}^{q} Sc_i$$

For Room Conflict,

$$F_6 = w_6 \sum_{i=1}^{u} R_i$$

So in total the cost for schedule by simulated annealing would be[2]

$$F = F_1 + F_2 + F_3 + F_4 + F_5 + F_6$$

### 3.2.5 Cooling Schedule

It's an optimization algorithm which uses heuristic values. In every step of iteration, the new temperature is being calculated using previous temperature multiplied by cooling rate. This process runs for the whole schedule until the energy comes down to normal for the timetable.

## 3.3 Q-Learning

[3][20] Q learning, a form of reinforcement learning in which the agent learns to assign values to state-action pairs. We need first to make a distinction between what is true of the world and what the agent thinks is true of the world. First let's consider what's true of the world. If an agent is in a particular state and takes a particular action, we are interested in any immediate reinforcement that's received but also in future reinforcements that result from ending up in a new state where further actions can be taken, actions that follow a particular policy. Given a particular action in a particular state followed by behavior that follows a particular policy, the agent will receive a particular set of reinforcements. This is a fact about the world. In the simplest case, the Q-value for a state-action pair is the sum of all of these reinforcements, and the Q-value function is the function that maps from state-action pairs to values. But the sum of all future reinforcements may be infinite when there is no terminal state, and besides, we may want to weight the future less than the here-and-now, so instead a discounted cumulative reinforcement is normally used: future reinforcements are weights by a value gamma between 0 and 1. A higher value of gamma means that the future matters more for the Q-value of a given action in a given state.

If the agent knew the Q-values of every state-action pair, it could use this information to select an action for each state. The problem is that the agent initially has no idea what the Q-values of any state-action pairs are. The agent's goal, then, is to settle on an optimal Q-value function, one which that assigns the appropriate values for all state/action pairs. But Q-values depend on future reinforcements, as well as current ones. How can the agent learn Q-values when it only seems to have access to immediate reinforcement? It learns using these two principles, which are the essence of reinforcement learning:

- If an action in a given state causes something bad to happen, learn not to do that action in that situation. If an action in a given state causes something good to happen, learn to do that action in that situation.
- If all actions in a given state cause something bad to happen, learn to avoid that state. That is, don't take actions in other states that would lead you to be in that bad state. If any action in a given state causes something good to happen, learn to like that state.

The second principle is the one that makes the reinforcement learning magic happen. It permits the agent to learn high or low values for particular actions from a particular state, even when there is no immediate reinforcement associated with those actions. For example, in our time Scheduling problem, the agent receives a reward when it reaches the goal from the random state. It now knows that the path is a good one to go to because you can get rewarded in only one move from it.

**Mathematical Explanation**

Here is the mathematical detail. First, consider the optimal Q-value function, the one that represents what's true of the world.

$$Q^*(x_t, u_t) = r(x_t, u_t) + \gamma \max_{u_{t+1}} Q^*(x_{t+1}, u_{t+1})$$

That is, the optimal Q-value of for a particular action in a particular state is the sum of the reinforcement received when that action is taken and the discounted best Q-value for the state that is reached by taking that action[1][2]. The agent would like to approach this value for each state-action pair. At any given time during learning, the agent stores a particular Q-value for each state-action pair. At the beginning of learning, this value is random or set at some default. Learning should move it closer to its optimal value. In order to do this, the agent repeatedly takes actions in particular states and notes the reinforcements that it receives. It then updates the stored Q-value for that state-action pair using the reinforcement received and the stored Q-values for the next state. Assuming the Q-values are stored in a lookup table, the agent could use one of these update equations:

$$Q^{new}(X_t, U_t) = (1-\eta)Q^{old}(x_t, u_t) + \eta[r(x_t, u_t) + \gamma \max_{u_{t+1}} Q^{old}(x_{t+1}, u_{t+1})]$$

The first equation sets the Q-value to be the sum of the reinforcement received and the discounted best Q-value for the next state. But this is usually a bad idea because the information just received may be faulty for one reason or another. It is better to update more gradually, to use the new information to move in a particular direction, but not to make too strong a commitment. The second update equation reflects this strategy. There is learning rate, which controls the learning step size, that is, how fast learning takes place. The new Q-value for the state and action is the weighted combination of the old Q-value for that state and action and what the new information would lead us to believe. Later we will see how a neural network can replace the lookup table for storing Q-values.

### Q-Learning Algorithm

Q-Learning learns the optimal policy even when actions are selected according to a more exploratory or even random policy. The procedural form of the algorithm is:

```
Initialize **Q(s,a)** arbitrarily Repeat (for each episode):
Initialize **s**
Repeat (for each step of episode):
choose **a** from **s** using policy derived from **Q**
Take action **a**, observe **r**,**s'**
    $Q(s,a) = Q(s,a) + \alpha[r + \gamma max_{\alpha}, Q(s',a') - Q(s,a)]$
    s = s';
until **s** is terminal.
```

This procedural approach can be translated into plain English steps as follows:

- Initialize the Q-values table, Q(s, a).
- Observe the current state, s.
- Choose an action, a, for that state based on one of the action selection policies.
- Take the action, and observe the reward, r, as well as the new state, s'.
- Update the Q-value for the state using the observed reward and the maximum reward possible for the next state. The updating is done according to the formula and parameters described above.

- Set the state to the new state, and repeat the process until a terminal state is reached.

**Simplified:** The transition rule of Q learning is a very simple formula:
Q(state, action) = R(state, action) + gamma * Max[Q(next state, all actions)]
The gamma parameter has a range of 0 to 1 (0 <= *gamma* < 1), and ensures the convergence of the sum. If gamma is closer to zero, the agent will tend to consider only immediate rewards. If gamma is closer to one, the agent will consider future rewards with greater weight, willing to delay the reward.
The Q-Learning algorithm goes as follows:

- Set the gamma parameter, and environment rewards in matrix R.
- Initialize matrix Q to zero.
- For each episode: Select a random initial state.

  While the goal state hasn't been reached.

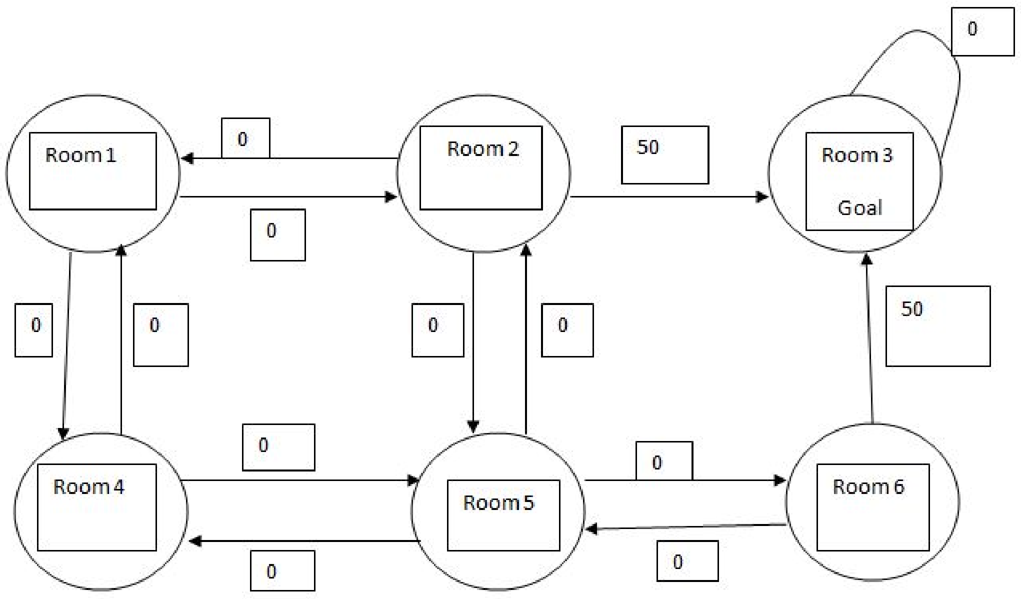
- Select one among all possible actions for the current state.
- Using this possible action, consider going to the next state.
- Get maximum Q value for this next state based on all possible actions.
- Compute: Q(state, action) = R(state, action) + gamma * Max[Q(next state, all actions)]
- Set the next state as the current state.

**Implementation and Result**

Q-learning is a technique for letting the AI learn by itself by giving it reward or punishment. We have implied this method for a simple sector of our Time Scheduling Project. Suppose, we have taken all the room numbers of a university, we used all of them as a state. Our example shows the Q-learning used for path finding to get the exact goal which is the appropriate room. According to the method it learns where it should go from any state. The process starts at a random place, it keeps memory of the score while it explores the area, whenever it reaches the goal, we repeat with a new random start. After enough repetitions the score values will be stationary (convergence). In this example the action outcome is deterministic (transition probability is 1) and the action selection is random. The score values are calculated by the Q-learning algorithm Q(s,a). The image shows the states (Room1,Room2,Room3,Room4,Room5,Room6) possible actions from the states and the reward given.

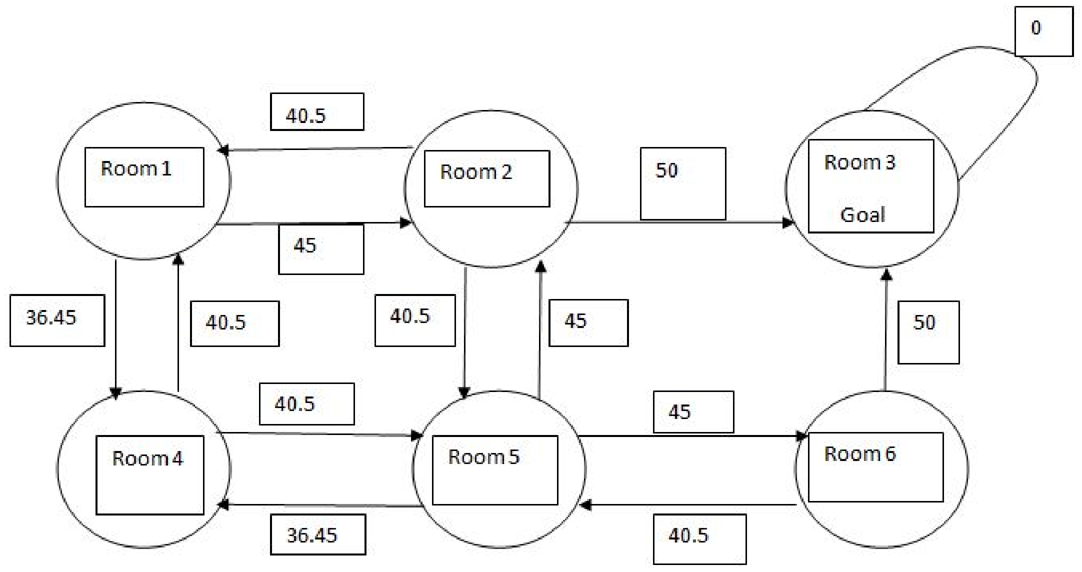
**Step-1**

We set reward 50 to only the nearby states of goal, rest all are 0.



**Step-2**

According to the algorithm we find the reward and choose path with maximum value which reaches to the goal.



## 3.4 Human Machine Interaction

Human machine interaction is a method which finds a solution using another solution. First of all, the method finds one single solution which can be feasible enough to use further. Then , it improves the solution manually.This model is based on mainly approximation and modification. The shortcomings of this method is the cost of computation which is very high on basis of different parameter.

# Chapter 4

# Problem Definition

University course timetabling is one of the most difficult and time consuming problem to solve. Here,events like course,seminar,sessions etc are to be assigned in different resources like time slots, participants/lecturer/instructor and rooms.The fact is this allocation must maintain different types of constraints which could be divided into two categories which are:[8][13]

- **Hard Constraints:** These types of constraints are never to be violated by any mean. Usually these types of constraint are very carefully defined so that if any sudden critical situation arise could be handled in shortest possible time. Some of the hard constraints we define in our thesis are:

  - One room can not be assigned more than one courses in single time slot.
  - One lecturer is allowed to take one class at a time.
  - Teachers must be assigned according to their specialization and preference.

- **Soft Constraints:** Some constraints are maintained in scheduling, violation of which are not that much disastrous. Maintaining these constraints enrich the quality of the timetabling.

- Teachers preferable time slots to be assigned .

- Back to back classes for teachers are suggested to avoid.

- Senior teachers should get highest priority for assigning courses and slot.

The quality of auto-generated University Course Schedule depends on how many constraints they meet. It is quite impossible to fulfill all of them.

## 4.1 Problem for BRAC University

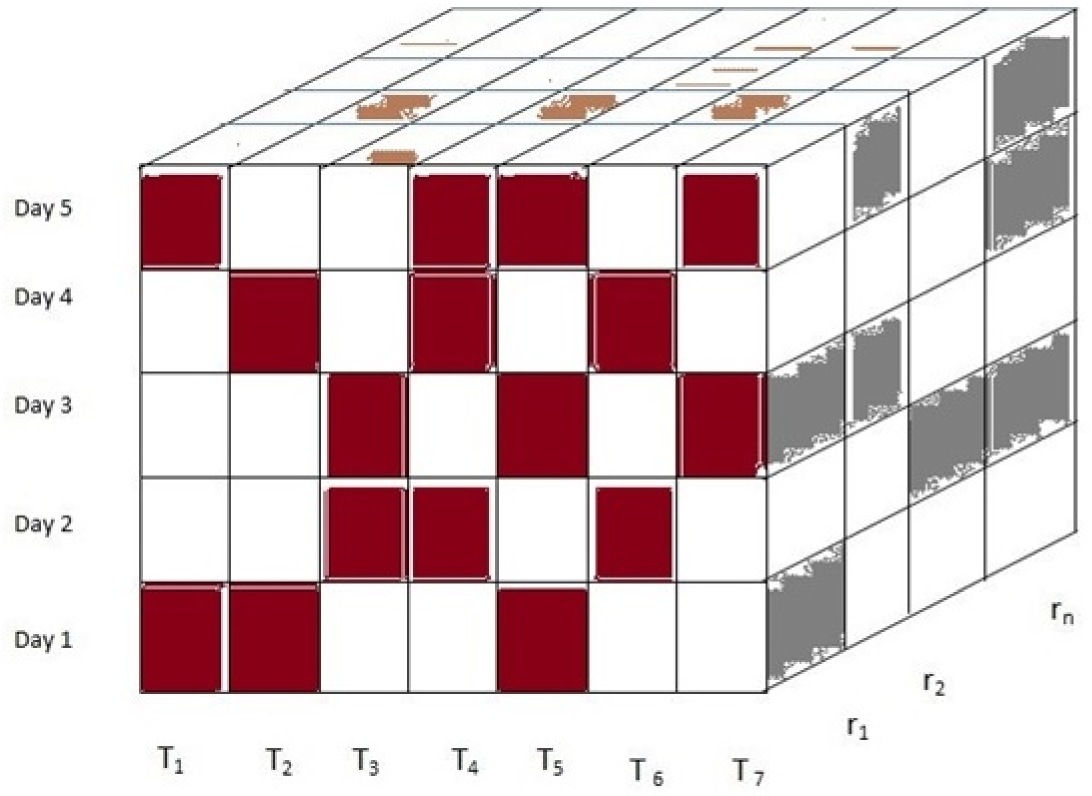
As a leading University in Bangladesh which is ensuring higher education, it has some unique facilities, rules and regulation . Some times rules are helpful for avoiding complexity. Sometimes it makes things more complicated in certain criteria. As, being private university government facilities are absent in here. So, apparently hundred percent of it's resources are provided and ensured by the authority and trusty board. As a new university, comparing to other government and traditional university like Dhaka University, BUET, Rajshahi University and many more universities, it does not have a great establish alumni association. So, Scholarship from outside university are very rare. As a result, University has limited options to arrange events like class, seminar and academic facilities outside the regular timing. Another Problem is BRAC University does not have any permanent campus and departments do not have buildings for their own operation. They are just overlapping each other for every aspects of resources.

### 4.1.1 Constraints for BRAC University

BRAC University is an open credit university where different credits courses are being offered to be taken by . But restrictions are defined as prerequisite. Students are free to take any courses according to the curriculum but they have to ensure that prerequisite for that particular course is being completed. On the other hand, authority holds the responsibility to make schedule such a way so that students do not have to mess up with the schedule due to prerequisite hazards. So, section number of higher courses actually depend on lower courses of university. So. according to the number of sections for particular course refers the need of instructors.

### 4.1.2 Multidimensional representation

The Scheduling problem is easy to describe in a multidimensional view. Here, resources like rooms and time slots can easily be identified. In a particular day for one particular slot all the rooms which are permitted for one department are available. So, One unit cube of the three dimensional view represents a unique allowable resource.[19]



Some particular components are :

- every courses are to be taught three hours per week. This time is divided into two equals time slots.
- one teacher is assign for each section of a particular course .
- lecturers should be available according to their designated time into assigned room.

### 4.1.3 Resources of BRAC University

To define the problem by set of resources and events,first we need to clarify each and every set of needed components. For $n$ number of courses if $m$ number of lecturers are available where $q$ numbers of rooms to be designated into several time slots than a schedule problem can be well defined. So, to define in mathematical expression lets assume the needed sets like:

Set of n Courses, $C = \{c_1, c_2, c_3, ..., c_n\}$
Set of Sections for corresponding courses, $S = \{s_1, s_2, s_3, ..., s_n\}$
Here, course maps to section.
Set of m Lecturers, $L = \{l_1, l_2, l_3, ..., l_m\}$
Set of q Rooms,$R = \{r_1, r_2, r_3, ..., r_q\}$

Set of $i * j$ Time slots,$T = \Sigma\{t_i, d_j\}$ Where $i = slots/day$; $j = days/week$ So the problem is defined with 5 tuples [6][4] [8][9]

$$f(x) =< C, S, L, R, T >$$

According to the definition we designed our chromosomes for the implementation of Genetic Algorithm to produce new routine.

# Chapter 5

# Problem Solving Approach

In an on going semester BRAC University needs to prepare a whole new schedule for upcoming semester. Most of the time maximum teachers remains unchanged. So, it is pretty much predictable that teachers are going to take more or less same courses in next semester.There are several causes for having changes in new schedule like

- The number of student increase in each semester.
- Arrival of Ramadan.
- Arrival of new teachers.
- Offering new courses.
- Number of rooms
- Recovery of leaving teacher

In making of new schedule different approaches are introduced by the person/s who is/are responsible for it. What if one single approach can solved all these headache and ensure a new time table which is acceptable at a high fitness scales? In this thesis, we try to represent such an approach in an organized manner so that the preparations of a new schedule can be well understood.

## 5.1 Genetic Algorithm Approach

### 5.1.1 Chromosome Encoding

As discussed before, to solve the scheduling problem using GA, we needed to encode the data set into sets of **Chromosomes** [11]. We defined each Chromosome with six parts.

| | | | | | |
|---|---|---|---|---|---|
| Instructor | Course | Section | Room | Day | Time |

For easier manipulation we merged days and Time into one as **Time Slot**.

| | | | | |
|---|---|---|---|---|
| Instructor | Course | Section | Room | Time Slot |

We use String encoding for the Chromosomes, like for Instructor we use his/her three length initial; for Course a six length course id; Section for any course would be a digit a per need; Rooms will be encoded as their name given by the university to identify according to building floor and room number; and time slot will be denoted according to their number. Now, we define different data sets according to the need for **Chromosome** encoding.[13][5]

## 5.2 Data Set

To do so first of all, we need to establish a parameter according to the problem definition which will be resonating to BRAC University curriculum for CSE. One of the most reliable sources for collecting the relative data set is the routine which is provided by the department before each semester's registration to the students. In these routines the facts are clearly indicated that which teacher is going to take which courses in which time and in which room the class will take place. for establishing the data set we need we collected ten consecutive routines from fall 2012 to fall 2015. From them eight routines have been discarded because most of the data from the earlier routines became unusable due to a lot of inevitable changes in administrative and objective structure of the department. Our most valuable data set has come from latest four semester(Fall 2014,Spring 2015,Summer 2015 and Fall 2015).

**Room Set**

After analysis of collected data we found a set of rooms from several buildings which are frequently used for different classes. We defined the set as $R$ which we have the cardinality $R_n$. It is need to be cleared that no lab room are included as we have not focused about lab scheduling of corresponding courses.[11][5]

**Instructor Set**

We also made an analysis on another resource which we need for scheduling is instructors. For instructors the defined set is I as we used in here. An analytic survey over CSE department of BRAC University is found that junior instructors are not allocated to take theory classes for certain period of time which varies instructor to instructor. So, again we "Redefine" our instructor set a functional instructor set, $I_f$ which has cardinality $I_M$. We clearly state that $I_f$ includes those instructor who are allocated one or more theory classes for one particular semester.[11][5]

**Course Set**

A predefined set which is designed by BRAC University Curriculum for CSE including necessary and elective courses; which we denote as $C$. Few years back where we were fresh graduate students, noticed that some of the necessary courses were being offered in attractive semester. In an explanation lets talk about the course "Database System" which course id is CSE 370 in BRAC University. If this course was being offered in Fall semester then it would be offered in Fall semester on corresponding year where summer semester is skipped to offer the course. As a growing university students numbers are counting up in each semester. Recently we found that this skip methodologies now in action only for some elective courses and not applicable for necessary courses. The cardinality of course set is defined by the BRAC University curriculum for CSE is $C_P$. [11][5]

**Time set**

In general BRAC University goes functional for five days a week for undergraduate program. Each day has theory class periods, starts at 8.00am and ends at 6.30pm, one hour twenty minutes span for each and after every class period, ten minutes break for next class. Counting One and half hour a unit time period for one theory class we got seven(7) slots for one day and $7 * 5 = 35$ time slots for a week [6] [11].

| Day/Time | T1 | T2 | T3 | T4 | T5 | T6 | T7 |
|---|---|---|---|---|---|---|---|
| Day 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Day 2 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Day 3 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| Day 4 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| Day 5 | 28 | 29 | 30 | 31 | 32 | 33 | 34 |

## 5.3 Probability Analysis

The chance of an event to occur for an experiment is the probability of that event. If an event can occur in $A$ number of ways and total number of outcomes is $B$ then probability of event $E$ is

$$P(E) = A/B$$

An important fact is, there are courses which can be taught by different number of instructors and there also can be courses which is taught by only one particular teacher. In that case

$$P(Instructor|Course) = 1$$

For the case of multiple instructors for one course which have multiple sections:-
lets assume that,
$a$ = number of sections
$b$ = number of instructors could be assigned

**Probability** of an **Instructor** to be assigned for a particular **Course** is :-

$$P(Instructor|Course) = f(a, b) = \sum_{i=a}^{1} \frac{1}{i_{C_1} * b_{C_1}}$$

We see that each courses could have multiple sections to allow all the students who want to take the course for that particular semester. If course $C_i$ has number of sections $S_i$ then total number of instances for the courses to be offered by the department for a particular semester is

$$N = \sum_{i=1}^{n} C_i * S_i, n \in R$$

As we have $N$ number of instances of courses to be assigned within 35 slots where we have $R_n$ number of rooms, we can allocate maximum $R_n$ course instances for one slot. A slot could be remain empty means no course instance is assigned on that particular slot. Another important point is one course instance should be allocated for three hours a week, which indicates two different time slots in 35 slots which are to be assigned. Under this circumstances total instances for courses becomes double means $2 * N$ is waiting for designated slots. So, for one slot the probability to be assigned for one course instance is

$$P(Course|Slot) = \frac{1}{(35 * R_n)_{C_1} * (2 * N)_{C_1}}$$

We experienced in our University that one or two rooms always kept free for any urgent need. For satisfying this constraint we kept three rooms reserved, so that if any clash happens only due to same room then it can be solved up to three clashes. At this pint the probability gets revised as

$$P(Course|Slot) = \frac{1}{(35 * (R_n - 3))_{C_1} * (2 * N)_{C_1}}$$

# Chapter 6

# Data Manipulation

To get result from raw data we need to collect corresponding data from BRAC University and store them into Database according to defined data set. We kept the facility to upload previous schedule and process them into raw data. When a previous routine is uploaded to the database in csv format our program then extract the necessary data and analyze them to insert into appropriate tables. Before that the whole set of data from the csv file is stored into a table called **Raw Data**. This Table is not actually relational but a virtual source for other tables which are dynamically enrich with necessary data from this **Raw Data** table. Some of the tables are manipulated in such a way that they are created at run time; if previously the table existed containing data the whole table get dropped before creating new one. In this way we avoided duplicate data.

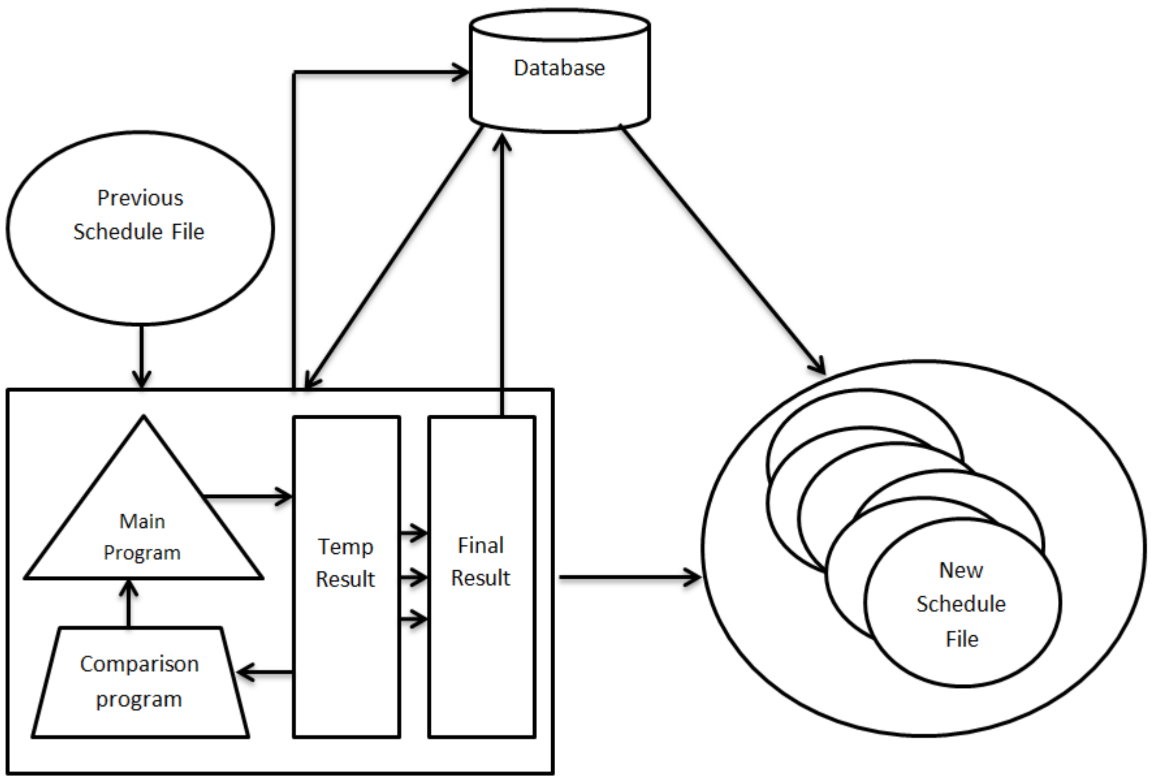
Along with database, we needed appropriate programming where our necessary algorithm(which is Genetic algorithm)is being implemented. We also needed programming for comparison purpose where different algorithms (Simulated Annealing and Q-Learning) are used. For these type of implementation where bunch of data sets are being handled within a single manipulation at one single time of the process we needed object oriented programming. Mainly **Encapsulation** property is required.

Finally , we would like to present our prepared schedule from the implementation results in different ways. Such as in csv format or using a traditional GUI(Graphical User Interface) designed as formal academic routine.

## 6.1 The Database

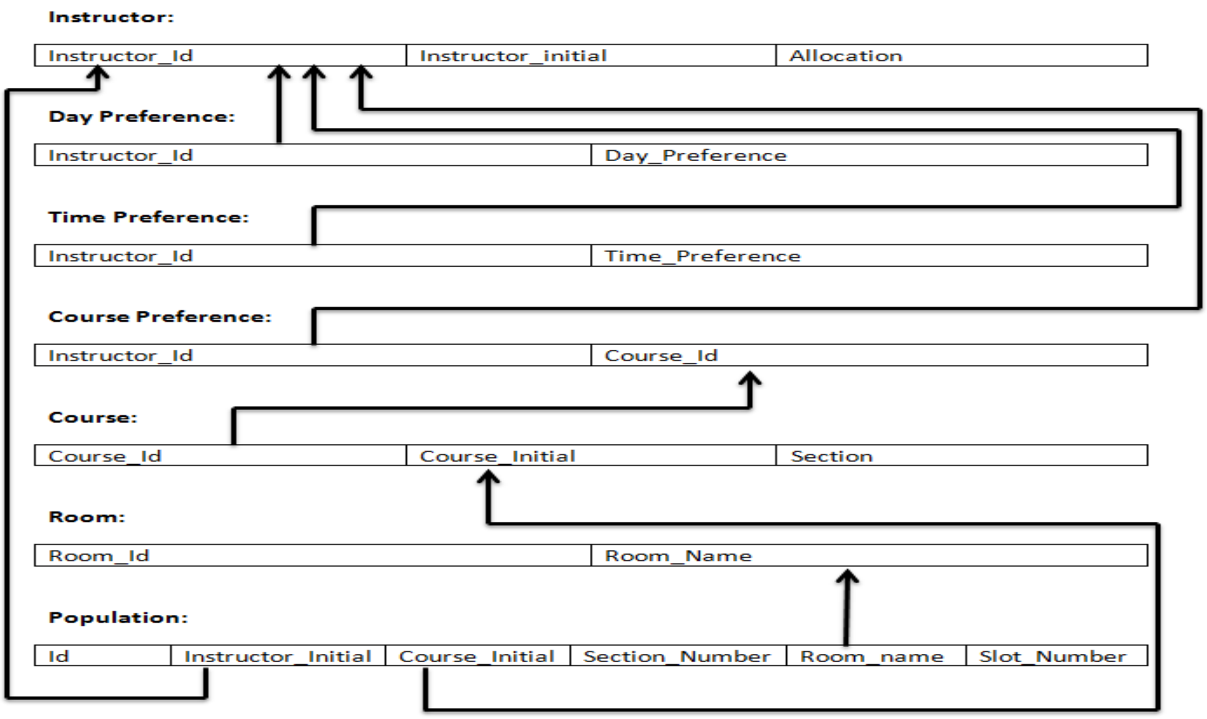
### 6.1.1 System Architecture

We used MySQL Database for necessary data storing purpose. When main program starts it provides data to designed data structure and after processing them when the program generates a temporary schedule which to be justified by comparison program several times to get a final result; then the result get stored again in database for further use and after getting a fair enough fitness for overall schedule a completely new schedule file will be generated.



### 6.1.2 Database Schema

Relational tables are in action to reduce redundancy and faster access to the data. Some tables are predefined and some tables are temporary, as they are dynamically created and dropped. Population table is completely dependent to other table to be created. When the main program starts at that moment previously crated population table gets dropped and completely new table gets crated for new population from where new results come out.



## 6.2 Program

The whole task was divided into different modules where each and every module has different functions to perform. From parsing previous routine into raw data to generating a brand new routine every step works as a interconnected dependent network.the steps can be shown in a listed manner like:

- Data Analysis
  - Parse previous Schedule
  - Fetch into raw data
  - Categorize and store in different tables
- Random Population Generation
  - Select Course
  - Select Random Instructor
  - Assign appropriate Section number
  - Select Random Room
  - Select random Slot

| Instructor | Course | Section | Room | Day | Time |
|---|---|---|---|---|---|

- Fitness Evaluation[19][5]
  - Course Fitness (increasing Order Course ID)
  - Teacher Fitness (Priority Value)[5]
  - Slot Fitness (anti-proportional to Slot Number)

$$fitness = C_{value} * P_{value} * (35 - S_{value}).$$

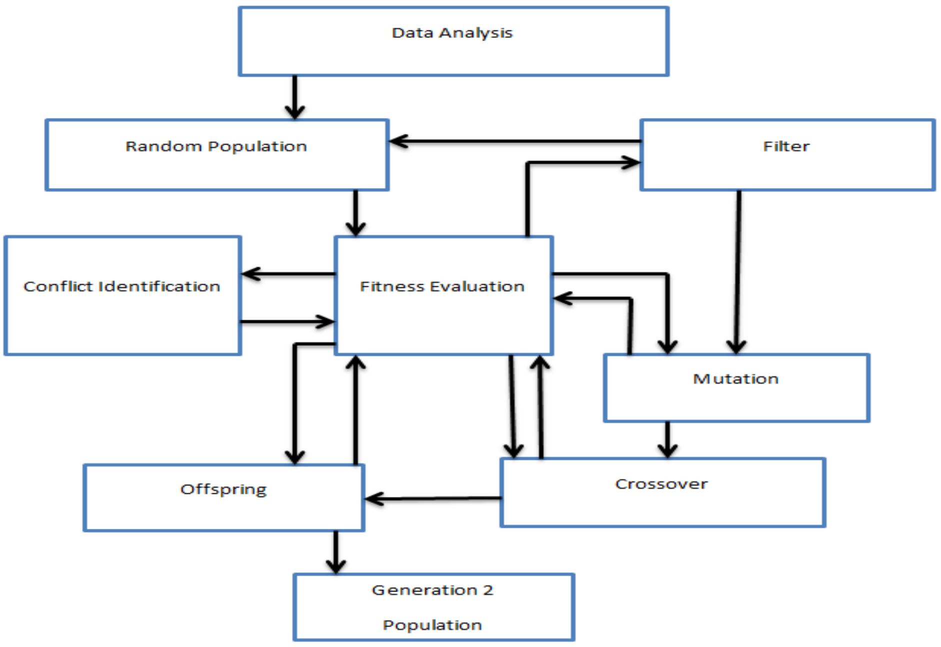
- Conflict Identification
  - Teacher Conflict // same teacher assigned for more than one courses in

same slots

  - Room Conflict // more than one class allocated in one room in same slots

- GA
  - Filter //based on fitness and Conflicts
  - Mutation // if conflict is only for room in same slot
  - Crossover // if conflict is for same teach in same slot
  - Offspring Selection // based on fitness evaluation after Mutation and Crossover

- Second Generation Population
  - Calculate total fitness
  - Compare with previously generated new Schedule fitness
  - Update new Schedule if new fitness is higher than previous one



### 6.2.1 Filter Mechanism

Filtering process filters all fit enough chromosomes on the basis of Instructors. It only checks if one particular teacher is assigned twice or more in single time slot. If such hard constraint violation is identified, the conflicted chromosomes can not pass through the filtering process.

### 6.2.2 Crossover Mechanism

Single point Crossover is used in this paper.After filtering process Crossover process goes for the left conflicted chromosomes. At first it checks either the conflict is for Instructor or not, if for instructor then the crossover point is only the instructor part. If the conflict is not for Instructor it goes for crossover exit point.

### 6.2.3 mutation Mechanism

After **Filter** and **Crossover** process the main program(GA) checks whether any room conflict is present or not. If room constraint violation number is less than or equal to present available room number then Mutation process only mutates room part of the Chromosomes. Otherwise the process seeks for available rooms in other slots and mutates room part and slot part of that corresponding Chromosome accordingly.

# Chapter 7

# Result Representation

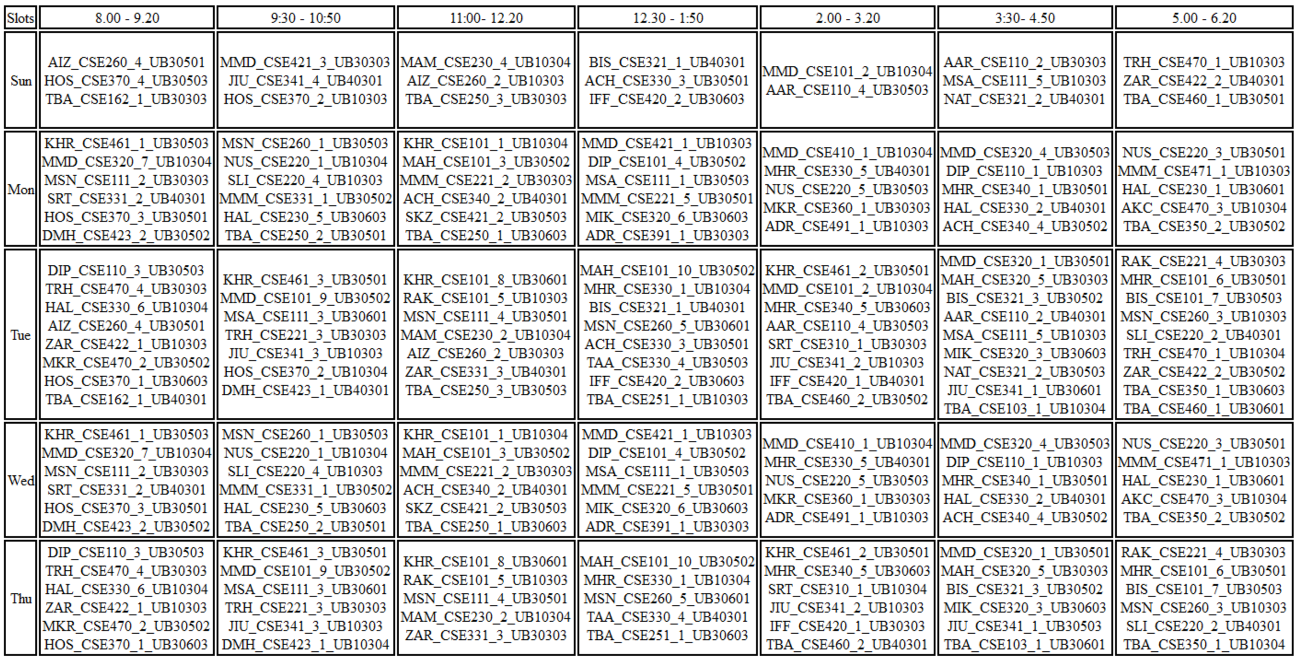
When the program runs for the very first time a new schedule file is produced and it happens in shorter time comparatively as the result has a fitness value which is to compare only with itself. From the second time run of the program and onward *runtime* increase. Every time the fitness of a newly generated schedule is higher than the last kept schedule only then the new one gets saved as a new schedule.

## 7.1 New Schedule

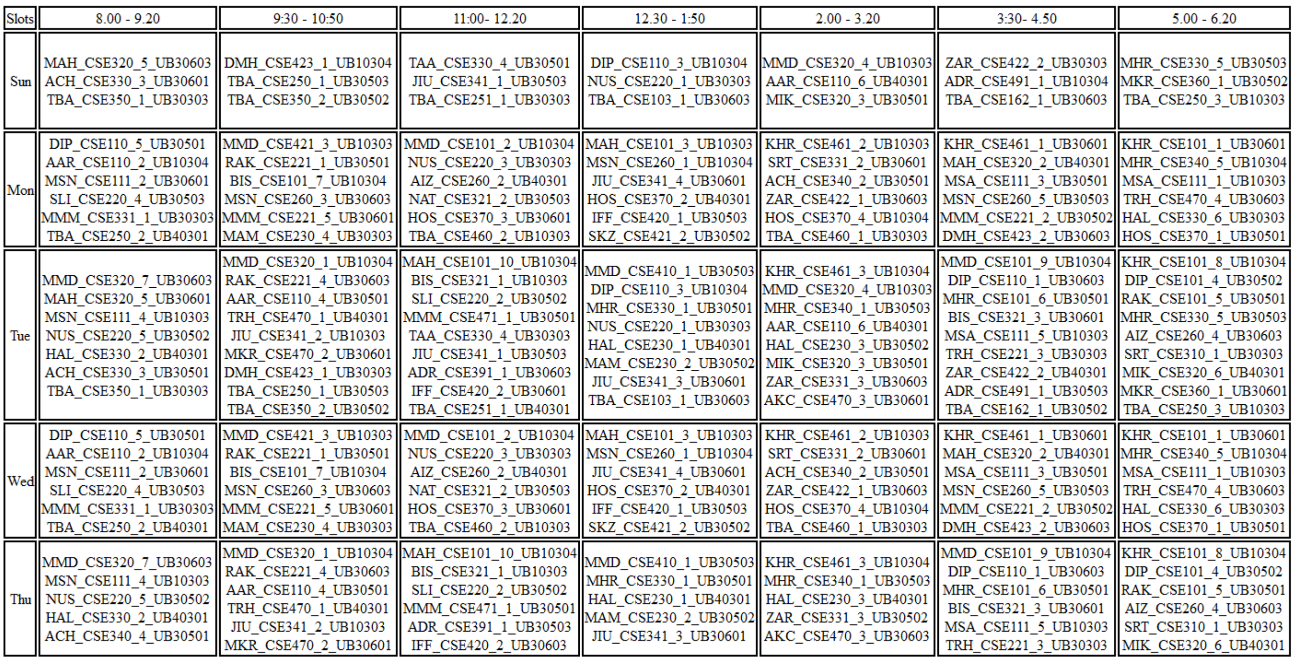
The schedule which is generated by the procedure discussed earlier can be represented in a tabular format. The results are shown in this paper are obtained from 15 times execution of the main program. Each time the program generates a better fitness schedule than the previous one it takes more execution time. If we gather the run time in a table then the comparison comes clear.

| **No.** | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | E10 | E11 | E12 | E13 | E14 | E15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Runtime** | 31.7 | 15.4 | 16.2 | 28.8 | 16.4 | 16.2 | 16.5 | 16.6 | 16.5 | 29.5 | 22.9 | 18.9 | 16.03 | 16.03 | 15.3 |

From the execution time we can have a prediction on each generation either it is a good one, can bee accepted or comparatively bad one and be rejected. In the tabular format, we tried to show a detail information for a class where slots, courses, instructors, sections are arranged according to their correspondence.



| Slots | 8.00 - 9.20 | 9:30 - 10:50 | 11:00- 12.20 | 12.30 - 1:50 | 2.00 - 3.20 | 3:30- 4.50 | 5.00 - 6.20 |
|---|---|---|---|---|---|---|---|
| Sun | AIZ_CSE260_4_UB30501<br>HOS_CSE370_4_UB30503<br>TBA_CSE162_1_UB30303 | MMD_CSE421_3_UB30303<br>JIU_CSE341_4_UB40301<br>HOS_CSE370_2_UB10303 | MAM_CSE230_4_UB10304<br>AIZ_CSE260_2_UB10303<br>TBA_CSE250_3_UB30303 | BIS_CSE321_1_UB40301<br>ACH_CSE330_3_UB30501<br>IFF_CSE420_2_UB30603 | MMD_CSE101_2_UB10304<br>AAR_CSE110_4_UB30503 | AAR_CSE110_2_UB30303<br>MSA_CSE111_5_UB10303<br>NAT_CSE321_2_UB40301 | TRH_CSE470_1_UB10303<br>ZAR_CSE422_2_UB40301<br>TBA_CSE460_1_UB30501 |
| Mon | KHR_CSE461_1_UB30503<br>MMD_CSE320_7_UB10304<br>MSN_CSE111_2_UB30303<br>SRT_CSE331_2_UB40301<br>HOS_CSE370_3_UB30501<br>DMH_CSE423_2_UB30502 | MSN_CSE260_1_UB30503<br>NUS_CSE220_1_UB10304<br>SLI_CSE220_4_UB10303<br>MMM_CSE331_1_UB30502<br>HAL_CSE230_5_UB30603<br>TBA_CSE250_2_UB30501 | KHR_CSE101_1_UB10304<br>MAH_CSE101_3_UB30502<br>MMM_CSE221_2_UB30303<br>ACH_CSE340_2_UB40301<br>SKZ_CSE421_2_UB30503<br>TBA_CSE250_1_UB30603 | MMD_CSE421_1_UB10303<br>DIP_CSE101_4_UB30502<br>MSA_CSE111_1_UB30503<br>MMM_CSE221_5_UB30501<br>MIK_CSE320_6_UB30603<br>ADR_CSE391_1_UB30303 | MMD_CSE410_1_UB10304<br>MHR_CSE330_5_UB40301<br>NUS_CSE220_5_UB30503<br>MKR_CSE360_1_UB30303<br>ADR_CSE491_1_UB10303 | MMD_CSE320_4_UB30503<br>DIP_CSE110_1_UB10303<br>MHR_CSE340_1_UB30501<br>HAL_CSE330_2_UB40301<br>ACH_CSE340_4_UB30502 | NUS_CSE220_3_UB30501<br>MMM_CSE471_1_UB10303<br>HAL_CSE230_1_UB30601<br>AKC_CSE470_3_UB10304<br>TBA_CSE350_2_UB30502 |
| Tue | DIP_CSE110_3_UB30503<br>TRH_CSE470_4_UB30303<br>HAL_CSE330_6_UB10304<br>AIZ_CSE260_4_UB30501<br>ZAR_CSE422_1_UB10303<br>MKR_CSE470_2_UB30502<br>HOS_CSE370_1_UB30603<br>TBA_CSE162_1_UB40301 | KHR_CSE461_3_UB30501<br>MMD_CSE101_9_UB30502<br>MSA_CSE111_3_UB30601<br>TRH_CSE221_3_UB30303<br>JIU_CSE341_3_UB10303<br>HOS_CSE370_2_UB10304<br>DMH_CSE423_1_UB40301 | KHR_CSE101_8_UB30601<br>RAK_CSE101_5_UB10303<br>MSN_CSE111_4_UB30501<br>MAM_CSE230_2_UB10304<br>AIZ_CSE260_2_UB30303<br>ZAR_CSE331_3_UB40301<br>TBA_CSE250_3_UB30503 | MAH_CSE101_10_UB30502<br>MHR_CSE330_1_UB10304<br>BIS_CSE321_1_UB40301<br>MSN_CSE260_5_UB30601<br>ACH_CSE330_3_UB30501<br>TAA_CSE330_4_UB30503<br>IFF_CSE420_2_UB30603<br>TBA_CSE251_1_UB10303 | KHR_CSE461_2_UB30501<br>MMD_CSE101_2_UB10304<br>MHR_CSE340_5_UB30603<br>AAR_CSE110_4_UB30503<br>SRT_CSE310_1_UB30303<br>JIU_CSE341_2_UB10303<br>IFF_CSE420_1_UB40301<br>TBA_CSE460_2_UB30502 | MMD_CSE320_1_UB30501<br>MAH_CSE320_5_UB30303<br>BIS_CSE321_3_UB30502<br>AAR_CSE110_2_UB40301<br>MSA_CSE111_5_UB10303<br>MIK_CSE320_3_UB30603<br>NAT_CSE321_2_UB30503<br>JIU_CSE341_1_UB30601<br>TBA_CSE103_1_UB10304 | RAK_CSE221_4_UB30303<br>MHR_CSE101_6_UB30501<br>BIS_CSE101_7_UB30503<br>MSN_CSE260_3_UB10303<br>SLI_CSE220_2_UB40301<br>TRH_CSE470_1_UB10304<br>ZAR_CSE422_2_UB30502<br>TBA_CSE350_1_UB30603<br>TBA_CSE460_1_UB30601 |
| Wed | KHR_CSE461_1_UB30503<br>MMD_CSE320_7_UB10304<br>MSN_CSE111_2_UB30303<br>SRT_CSE331_2_UB40301<br>HOS_CSE370_3_UB30501<br>DMH_CSE423_2_UB30502 | MSN_CSE260_1_UB30503<br>NUS_CSE220_1_UB10304<br>SLI_CSE220_4_UB10303<br>MMM_CSE331_1_UB30502<br>HAL_CSE230_5_UB30603<br>TBA_CSE250_2_UB30501 | KHR_CSE101_1_UB10304<br>MAH_CSE101_3_UB30502<br>MMM_CSE221_2_UB30303<br>ACH_CSE340_2_UB40301<br>SKZ_CSE421_2_UB30503<br>TBA_CSE250_1_UB30603 | MMD_CSE421_1_UB10303<br>DIP_CSE101_4_UB30502<br>MSA_CSE111_1_UB30503<br>MMM_CSE221_5_UB30501<br>MIK_CSE320_6_UB30603<br>ADR_CSE391_1_UB30303 | MMD_CSE410_1_UB10304<br>MHR_CSE330_5_UB40301<br>NUS_CSE220_5_UB30503<br>MKR_CSE360_1_UB30303<br>ADR_CSE491_1_UB10303 | MMD_CSE320_4_UB30503<br>DIP_CSE110_1_UB10303<br>MHR_CSE340_1_UB30501<br>HAL_CSE330_2_UB40301<br>ACH_CSE340_4_UB30502 | NUS_CSE220_3_UB30501<br>MMM_CSE471_1_UB10303<br>HAL_CSE230_1_UB30601<br>AKC_CSE470_3_UB10304<br>TBA_CSE350_2_UB30502 |
| Thu | DIP_CSE110_3_UB30503<br>TRH_CSE470_4_UB30303<br>HAL_CSE330_6_UB10304<br>ZAR_CSE422_1_UB10303<br>MKR_CSE470_2_UB30502<br>HOS_CSE370_1_UB30603 | KHR_CSE461_3_UB30501<br>MMD_CSE101_9_UB30502<br>MSA_CSE111_3_UB30601<br>TRH_CSE221_3_UB30303<br>JIU_CSE341_3_UB10303<br>DMH_CSE423_1_UB10304 | KHR_CSE101_8_UB30601<br>RAK_CSE101_5_UB10303<br>MSN_CSE111_4_UB30501<br>MAM_CSE230_2_UB10304<br>ZAR_CSE331_3_UB30303 | MAH_CSE101_10_UB30502<br>MHR_CSE330_1_UB10304<br>MSN_CSE260_5_UB30601<br>TAA_CSE330_4_UB40301<br>TBA_CSE251_1_UB30603 | KHR_CSE461_2_UB30501<br>MHR_CSE340_5_UB30603<br>SRT_CSE310_1_UB10304<br>JIU_CSE341_2_UB10303<br>IFF_CSE420_1_UB30303<br>TBA_CSE460_2_UB40301 | MMD_CSE320_1_UB30501<br>MAH_CSE320_5_UB30303<br>BIS_CSE321_3_UB30502<br>MIK_CSE320_3_UB30603<br>JIU_CSE341_1_UB30503<br>TBA_CSE103_1_UB30601 | RAK_CSE221_4_UB30303<br>MHR_CSE101_6_UB30501<br>BIS_CSE101_7_UB30503<br>MSN_CSE260_3_UB10303<br>SLI_CSE220_2_UB40301<br>TBA_CSE350_1_UB10304 |



| Slots | 8.00 - 9.20 | 9:30 - 10:50 | 11:00- 12.20 | 12.30 - 1:50 | 2.00 - 3.20 | 3:30- 4.50 | 5.00 - 6.20 |
|---|---|---|---|---|---|---|---|
| Sun | MAH_CSE320_5_UB30603<br>ACH_CSE330_3_UB30601<br>TBA_CSE350_1_UB30303 | DMH_CSE423_1_UB10304<br>TBA_CSE250_1_UB30503<br>TBA_CSE350_2_UB30502 | TAA_CSE330_4_UB30501<br>JIU_CSE341_1_UB30503<br>TBA_CSE251_1_UB30303 | DIP_CSE110_3_UB10304<br>NUS_CSE220_1_UB30303<br>TBA_CSE103_1_UB30603 | MMD_CSE320_4_UB10303<br>AAR_CSE110_6_UB40301<br>MIK_CSE320_3_UB30501 | ZAR_CSE422_2_UB30303<br>ADR_CSE491_1_UB10304<br>TBA_CSE162_1_UB30603 | MHR_CSE330_5_UB30503<br>MKR_CSE360_1_UB30502<br>TBA_CSE250_3_UB10303 |
| Mon | DIP_CSE110_5_UB30501<br>AAR_CSE110_2_UB10304<br>MSN_CSE111_2_UB30601<br>SLI_CSE220_4_UB30503<br>MMM_CSE331_1_UB30303<br>TBA_CSE250_2_UB40301 | MMD_CSE421_3_UB10303<br>RAK_CSE221_1_UB30501<br>BIS_CSE101_7_UB10304<br>MSN_CSE260_3_UB30603<br>MMM_CSE221_5_UB30601<br>MAM_CSE230_4_UB30303 | MMD_CSE101_2_UB10304<br>NUS_CSE220_3_UB30303<br>AIZ_CSE260_2_UB40301<br>NAT_CSE321_2_UB30503<br>HOS_CSE370_3_UB30601<br>TBA_CSE460_2_UB10303 | MAH_CSE101_3_UB10303<br>MSN_CSE260_1_UB10304<br>JIU_CSE341_4_UB30601<br>HOS_CSE370_2_UB40301<br>IFF_CSE420_1_UB30503<br>SKZ_CSE421_2_UB30502 | KHR_CSE461_2_UB10303<br>SRT_CSE331_2_UB30601<br>ACH_CSE340_2_UB30501<br>ZAR_CSE422_1_UB30603<br>HOS_CSE370_4_UB10304<br>TBA_CSE460_1_UB30303 | KHR_CSE461_1_UB30601<br>MAH_CSE320_2_UB40301<br>MSA_CSE111_3_UB30501<br>MSN_CSE260_5_UB30503<br>MMM_CSE221_2_UB30502<br>DMH_CSE423_2_UB30603 | KHR_CSE101_1_UB30601<br>MHR_CSE340_5_UB10304<br>MSA_CSE111_1_UB10303<br>TRH_CSE470_4_UB30603<br>HAL_CSE330_6_UB30303<br>HOS_CSE370_1_UB30501 |
| Tue | MMD_CSE320_7_UB30603<br>MAH_CSE320_5_UB30601<br>MSN_CSE111_4_UB10303<br>NUS_CSE220_5_UB30502<br>HAL_CSE330_2_UB40301<br>ACH_CSE330_3_UB30501<br>TBA_CSE350_1_UB30303 | MMD_CSE320_1_UB10304<br>RAK_CSE221_4_UB30603<br>AAR_CSE110_4_UB30501<br>TRH_CSE470_1_UB40301<br>JIU_CSE341_2_UB10303<br>MKR_CSE470_2_UB30601<br>DMH_CSE423_1_UB30303<br>TBA_CSE250_1_UB30503<br>TBA_CSE350_2_UB30502 | MAH_CSE101_10_UB10304<br>BIS_CSE321_1_UB10303<br>SLI_CSE220_2_UB30502<br>MMM_CSE471_1_UB30501<br>TAA_CSE330_4_UB30303<br>JIU_CSE341_1_UB30503<br>ADR_CSE391_1_UB30603<br>IFF_CSE420_2_UB30601<br>TBA_CSE251_1_UB40301 | MMD_CSE410_1_UB30503<br>DIP_CSE110_3_UB10304<br>MHR_CSE330_1_UB30501<br>NUS_CSE220_1_UB30303<br>HAL_CSE230_1_UB40301<br>MAM_CSE230_2_UB30502<br>JIU_CSE341_3_UB30601<br>TBA_CSE103_1_UB30603 | KHR_CSE461_3_UB10304<br>MMD_CSE320_4_UB10303<br>MHR_CSE340_1_UB30503<br>AAR_CSE110_6_UB40301<br>HAL_CSE230_3_UB30502<br>MIK_CSE320_3_UB30501<br>ZAR_CSE331_3_UB30603<br>AKC_CSE470_3_UB30601 | MMD_CSE101_9_UB10304<br>DIP_CSE110_1_UB30603<br>MHR_CSE101_6_UB30501<br>BIS_CSE321_3_UB30601<br>MSA_CSE111_5_UB10303<br>TRH_CSE221_3_UB30303<br>ZAR_CSE422_2_UB40301<br>ADR_CSE491_1_UB30503<br>TBA_CSE162_1_UB30502 | KHR_CSE101_8_UB10304<br>DIP_CSE101_4_UB30502<br>RAK_CSE101_5_UB30501<br>MHR_CSE330_5_UB30503<br>AIZ_CSE260_4_UB30603<br>SRT_CSE310_1_UB30303<br>MIK_CSE320_6_UB40301<br>MKR_CSE360_1_UB30601<br>TBA_CSE250_3_UB10303 |
| Wed | DIP_CSE110_5_UB30501<br>AAR_CSE110_2_UB10304<br>MSN_CSE111_2_UB30601<br>SLI_CSE220_4_UB30503<br>MMM_CSE331_1_UB30303<br>TBA_CSE250_2_UB40301 | MMD_CSE421_3_UB10303<br>RAK_CSE221_1_UB30501<br>BIS_CSE101_7_UB10304<br>MSN_CSE260_3_UB30603<br>MMM_CSE221_5_UB30601<br>MAM_CSE230_4_UB30303 | MMD_CSE101_2_UB10304<br>NUS_CSE220_3_UB30303<br>AIZ_CSE260_2_UB40301<br>NAT_CSE321_2_UB30503<br>HOS_CSE370_3_UB30601<br>TBA_CSE460_2_UB10303 | MAH_CSE101_3_UB10303<br>MSN_CSE260_1_UB10304<br>JIU_CSE341_4_UB30601<br>HOS_CSE370_2_UB40301<br>IFF_CSE420_1_UB30503<br>SKZ_CSE421_2_UB30502 | KHR_CSE461_2_UB10303<br>SRT_CSE331_2_UB30601<br>ACH_CSE340_2_UB30501<br>ZAR_CSE422_1_UB30603<br>HOS_CSE370_4_UB10304<br>TBA_CSE460_1_UB30303 | KHR_CSE461_1_UB30601<br>MAH_CSE320_2_UB40301<br>MSA_CSE111_3_UB30501<br>MSN_CSE260_5_UB30503<br>MMM_CSE221_2_UB30502<br>DMH_CSE423_2_UB30603 | KHR_CSE101_1_UB30601<br>MHR_CSE340_5_UB10304<br>MSA_CSE111_1_UB10303<br>TRH_CSE470_4_UB30603<br>HAL_CSE330_6_UB30303<br>HOS_CSE370_1_UB30501 |
| Thu | MMD_CSE320_7_UB30603<br>MSN_CSE111_4_UB10303<br>NUS_CSE220_5_UB30502<br>HAL_CSE330_2_UB40301<br>ACH_CSE340_4_UB30501 | MMD_CSE320_1_UB10304<br>RAK_CSE221_4_UB30603<br>AAR_CSE110_4_UB30501<br>TRH_CSE470_1_UB40301<br>JIU_CSE341_2_UB10303<br>MKR_CSE470_2_UB30601 | MAH_CSE101_10_UB10304<br>BIS_CSE321_1_UB10303<br>SLI_CSE220_2_UB30502<br>MMM_CSE471_1_UB30501<br>ADR_CSE391_1_UB30503<br>IFF_CSE420_2_UB30603 | MMD_CSE410_1_UB30503<br>MHR_CSE330_1_UB30501<br>HAL_CSE230_1_UB40301<br>MAM_CSE230_2_UB30502<br>JIU_CSE341_3_UB30601 | KHR_CSE461_3_UB10304<br>MHR_CSE340_1_UB30503<br>HAL_CSE230_3_UB40301<br>ZAR_CSE331_3_UB30502<br>AKC_CSE470_3_UB30603 | MMD_CSE101_9_UB10304<br>DIP_CSE110_1_UB30603<br>MHR_CSE101_6_UB30501<br>BIS_CSE321_3_UB30601<br>MSA_CSE111_5_UB10303<br>TRH_CSE221_3_UB30303 | KHR_CSE101_8_UB10304<br>DIP_CSE101_4_UB30502<br>RAK_CSE101_5_UB30501<br>AIZ_CSE260_4_UB30603<br>SRT_CSE310_1_UB30303<br>MIK_CSE320_6_UB40301 |

If we represent the schedule in a raw manner including all related data like chromosome fitness, chromosome number and all other, then the representation would be like below:

| **Chromosome** | **Instructor** | **Course** | **Section** | **Room** | **Slot** | **Fitness** |
|---|---|---|---|---|---|---|
| 1 | MAH | CSE320 | 5 | UB30603 | 0 | 3185 |
| 2 | ACH | CSE330 | 3 | UB30601 | 0 | 10500 |
| 3 | TBA | CSE350 | 1 | UB30303 | 0 | 0 |
| 4 | DMH | CSE423 | 1 | UB10304 | 1 | 18360 |
| 5 | TBA | CSE250 | 1 | UB30503 | 1 | 0 |
| 6 | TBA | CSE350 | 2 | UB30502 | 1 | 0 |
| 7 | TAA | CSE330 | 4 | UB30501 | 2 | 495 |
| 8 | JIU | CSE341 | 1 | UB30503 | 2 | 11880 |
| 9 | TBA | CSE251 | 1 | UB30303 | 2 | 0 |
| 10 | DIP | CSE110 | 3 | UB10304 | 3 | 96 |
| 11 | NUS | CSE220 | 1 | UB30303 | 3 | 384 |
| 12 | TBA | CSE103 | 1 | UB30603 | 3 | 0 |
| 13 | MMD | CSE320 | 4 | UB10303 | 4 | 1612 |
| 14 | AAR | CSE110 | 6 | UB40301 | 4 | 1860 |
| 15 | MIK | CSE320 | 3 | UB30501 | 4 | 2821 |
| 16 | ZAR | CSE422 | 2 | UB30303 | 5 | 15600 |
| 17 | ADR | CSE491 | 1 | UB10304 | 5 | 19200 |
| 18 | TBA | CSE162 | 1 | UB30603 | 5 | 0 |
| 19 | MHR | CSE330 | 5 | UB30503 | 6 | 2610 |
| 20 | MKR | CSE360 | 1 | UB30502 | 6 | 1160 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 21 | TBA | CSE250 | 3 | UB10303 | 6 | 0 |
| 22 | DIP | CSE110 | 5 | UB30501 | 7 | 42 |
| 23 | AAR | CSE110 | 2 | UB10304 | 7 | 1680 |
| 24 | MSN | CSE111 | 2 | UB30601 | 7 | 168 |
| 25 | SLI | CSE220 | 4 | UB30503 | 7 | 168 |
| 26 | MMM | CSE331 | 1 | UB30303 | 7 | 1344 |
| 27 | TBA | CSE250 | 2 | UB40301 | 7 | 0 |
| 28 | MMD | CSE421 | 3 | UB10303 | 8 | 2700 |
| 29 | RAK | CSE221 | 1 | UB30501 | 8 | 189 |
| 30 | BIS | CSE101 | 7 | UB10304 | 8 | 39 |
| 31 | MSN | CSE260 | 3 | UB30603 | 8 | 429 |
| 32 | MMM | CSE221 | 5 | UB30601 | 8 | 567 |
| 33 | MAM | CSE230 | 4 | UB30303 | 8 | 2080 |
| 34 | MMD | CSE101 | 2 | UB10304 | 9 | 48 |
| 35 | NUS | CSE220 | 3 | UB30303 | 9 | 312 |
| 36 | AIZ | CSE260 | 2 | UB40301 | 9 | 286 |
| 37 | NAT | CSE321 | 2 | UB30503 | 9 | 728 |
| 38 | HOS | CSE370 | 3 | UB30601 | 9 | 5040 |
| 39 | TBA | CSE460 | 2 | UB10303 | 9 | 0 |
| 40 | MAH | CSE101 | 3 | UB10303 | 10 | 77 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 41 | MSN | CSE260 | 1 | UB10304 | 10 | 363 |
| 42 | JIU | CSE341 | 4 | UB30601 | 10 | 3960 |
| 43 | HOS | CSE370 | 2 | UB40301 | 10 | 10500 |
| 44 | IFF | CSE420 | 1 | UB30503 | 10 | 600 |
| 45 | SKZ | CSE421 | 2 | UB30502 | 10 | 12500 |
| 46 | KHR | CSE461 | 2 | UB10303 | 11 | 13920 |
| 47 | SRT | CSE331 | 2 | UB30601 | 11 | 160 |
| 48 | ACH | CSE340 | 2 | UB30501 | 11 | 3400 |
| 49 | ZAR | CSE422 | 1 | UB30603 | 11 | 12480 |
| 50 | HOS | CSE370 | 4 | UB10304 | 11 | 4200 |
| 51 | TBA | CSE460 | 1 | UB30303 | 11 | 0 |
| 52 | KHR | CSE461 | 1 | UB30601 | 12 | 13340 |
| 53 | MAH | CSE320 | 2 | UB40301 | 12 | 819 |
| 54 | MSA | CSE111 | 3 | UB30501 | 12 | 720 |
| 55 | MSN | CSE260 | 5 | UB30503 | 12 | 297 |
| 56 | MMM | CSE221 | 2 | UB30502 | 12 | 483 |
| 57 | DMH | CSE423 | 2 | UB30603 | 12 | 12420 |
| 58 | KHR | CSE101 | 1 | UB30601 | 13 | 440 |
| 59 | MHR | CSE340 | 5 | UB10304 | 13 | 816 |
| 60 | MSA | CSE111 | 1 | UB10303 | 13 | 1760 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 61 | TRH | CSE470 | 4 | UB30603 | 13 | 240 |
| 62 | HAL | CSE330 | 6 | UB30303 | 13 | 360 |
| 63 | HOS | CSE370 | 1 | UB30501 | 13 | 9240 |
| 64 | MMD | CSE320 | 7 | UB30603 | 14 | 364 |
| 65 | MAH | CSE320 | 5 | UB30601 | 14 | 3185 |
| 66 | MSN | CSE111 | 4 | UB10303 | 14 | 252 |
| 67 | NUS | CSE220 | 5 | UB30502 | 14 | 252 |
| 68 | HAL | CSE330 | 2 | UB40301 | 14 | 945 |
| 69 | ACH | CSE330 | 3 | UB30501 | 14 | 10500 |
| 70 | TBA | CSE350 | 1 | UB30303 | 14 | 0 |
| 71 | MMD | CSE320 | 1 | UB10304 | 15 | 1040 |
| 72 | RAK | CSE221 | 4 | UB30603 | 15 | 140 |
| 73 | AAR | CSE110 | 4 | UB30501 | 15 | 360 |
| 74 | TRH | CSE470 | 1 | UB40301 | 15 | 180 |
| 75 | JIU | CSE341 | 2 | UB10303 | 15 | 7200 |
| 76 | MKR | CSE470 | 2 | UB30601 | 15 | 360 |
| 77 | DMH | CSE423 | 1 | UB30303 | 15 | 18360 |
| 78 | TBA | CSE250 | 1 | UB30503 | 15 | 0 |
| 79 | TBA | CSE350 | 2 | UB30502 | 15 | 0 |
| 80 | MAH | CSE101 | 10 | UB10304 | 16 | 35 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 81 | BIS | CSE321 | 1 | UB10303 | 16 | 210 |
| 82 | SLI | CSE220 | 2 | UB30502 | 16 | 30 |
| 83 | MMM | CSE471 | 1 | UB30501 | 16 | 1767 |
| 84 | TAA | CSE330 | 4 | UB30303 | 16 | 495 |
| 85 | JIU | CSE341 | 1 | UB30503 | 16 | 11880 |
| 86 | ADR | CSE391 | 1 | UB30603 | 16 | 8360 |
| 87 | IFF | CSE420 | 2 | UB30601 | 16 | 456 |
| 88 | TBA | CSE251 | 1 | UB40301 | 16 | 0 |
| 89 | MMD | CSE410 | 1 | UB30503 | 17 | 368 |
| 90 | DIP | CSE110 | 3 | UB10304 | 17 | 96 |
| 91 | MHR | CSE330 | 1 | UB30501 | 17 | 1620 |
| 92 | NUS | CSE220 | 1 | UB30303 | 17 | 384 |
| 93 | HAL | CSE230 | 1 | UB40301 | 17 | 432 |
| 94 | MAM | CSE230 | 2 | UB30502 | 17 | 640 |
| 95 | JIU | CSE341 | 3 | UB30601 | 17 | 1440 |
| 96 | TBA | CSE103 | 1 | UB30603 | 17 | 0 |
| 97 | KHR | CSE461 | 3 | UB10304 | 18 | 9860 |
| 98 | MMD | CSE320 | 4 | UB10303 | 18 | 1612 |
| 99 | MHR | CSE340 | 1 | UB30503 | 18 | 306 |
| 100 | AAR | CSE110 | 6 | UB40301 | 18 | 1860 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 101 | HAL | CSE230 | 3 | UB30502 | 18 | 72 |
| 102 | MIK | CSE320 | 3 | UB30501 | 18 | 2821 |
| 103 | ZAR | CSE331 | 3 | UB30603 | 18 | 5440 |
| 104 | AKC | CSE470 | 3 | UB30601 | 18 | 510 |
| 105 | MMD | CSE101 | 9 | UB10304 | 19 | 8 |
| 106 | DIP | CSE110 | 1 | UB30603 | 19 | 6 |
| 107 | MHR | CSE101 | 6 | UB30501 | 19 | 12 |
| 108 | BIS | CSE321 | 3 | UB30601 | 19 | 672 |
| 109 | MSA | CSE111 | 5 | UB10303 | 19 | 1280 |
| 110 | TRH | CSE221 | 3 | UB30303 | 19 | 112 |
| 111 | ZAR | CSE422 | 2 | UB40301 | 19 | 15600 |
| 112 | ADR | CSE491 | 1 | UB30503 | 19 | 19200 |
| 113 | TBA | CSE162 | 1 | UB30502 | 19 | 0 |
| 114 | KHR | CSE101 | 8 | UB10304 | 20 | 20 |
| 115 | DIP | CSE101 | 4 | UB30502 | 20 | 1 |
| 116 | RAK | CSE101 | 5 | UB30501 | 20 | 15 |
| 117 | MHR | CSE330 | 5 | UB30503 | 20 | 2610 |
| 118 | AIZ | CSE260 | 4 | UB30603 | 20 | 11 |
| 119 | SRT | CSE310 | 1 | UB30303 | 20 | 180 |
| 120 | MIK | CSE320 | 6 | UB40301 | 20 | 1365 |

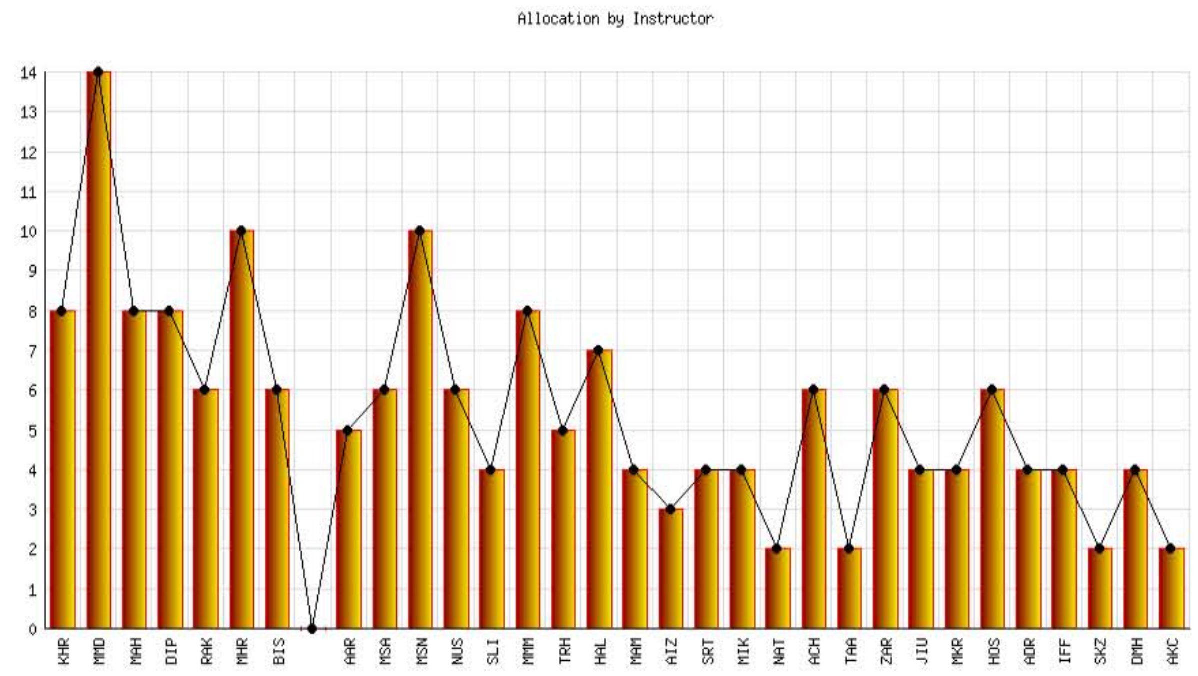
| | | | | | | |
|---|---|---|---|---|---|---|
| 121 | MKR | CSE360 | 1 | UB30601 | 20 | 1160 |
| 122 | TBA | CSE250 | 3 | UB10303 | 20 | 0 |
| 123 | DIP | CSE110 | 5 | UB30501 | 21 | 42 |
| 124 | AAR | CSE110 | 2 | UB10304 | 21 | 1680 |
| 125 | MSN | CSE111 | 2 | UB30601 | 21 | 168 |
| 126 | SLI | CSE220 | 4 | UB30503 | 21 | 168 |
| 127 | MMM | CSE331 | 1 | UB30303 | 21 | 1344 |
| 128 | TBA | CSE250 | 2 | UB40301 | 21 | 0 |
| 129 | MMD | CSE421 | 3 | UB10303 | 22 | 2700 |
| 130 | RAK | CSE221 | 1 | UB30501 | 22 | 189 |
| 131 | BIS | CSE101 | 7 | UB10304 | 22 | 39 |
| 132 | MSN | CSE260 | 3 | UB30603 | 22 | 429 |
| 133 | MMM | CSE221 | 5 | UB30601 | 22 | 567 |
| 134 | MAM | CSE230 | 4 | UB30303 | 22 | 2080 |
| 135 | MMD | CSE101 | 2 | UB10304 | 23 | 48 |
| 136 | NUS | CSE220 | 3 | UB30303 | 23 | 312 |
| 137 | AIZ | CSE260 | 2 | UB40301 | 23 | 286 |
| 138 | NAT | CSE321 | 2 | UB30503 | 23 | 728 |
| 139 | HOS | CSE370 | 3 | UB30601 | 23 | 5040 |
| 140 | TBA | CSE460 | 2 | UB10303 | 23 | 0 |

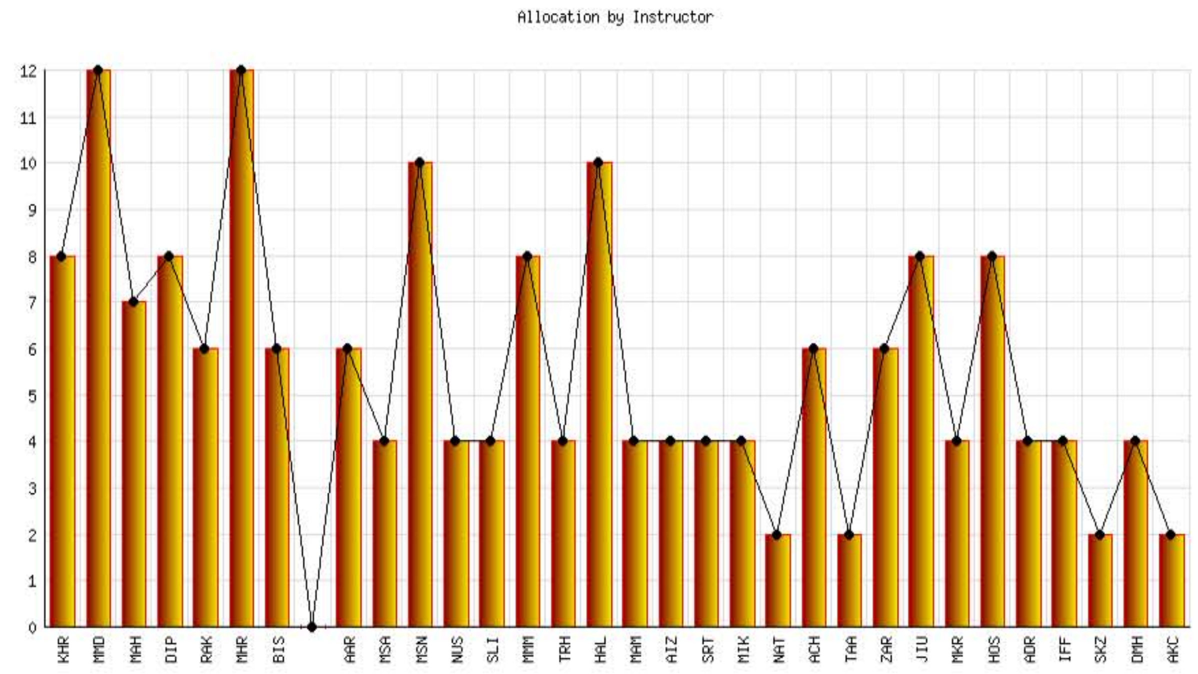
| | | | | | | |
|---|---|---|---|---|---|---|
| 141 | MAH | CSE101 | 3 | UB10303 | 24 | 77 |
| 142 | MSN | CSE260 | 1 | UB10304 | 24 | 363 |
| 143 | JIU | CSE341 | 4 | UB30601 | 24 | 3960 |
| 144 | HOS | CSE370 | 2 | UB40301 | 24 | 10500 |
| 145 | IFF | CSE420 | 1 | UB30503 | 24 | 600 |
| 146 | SKZ | CSE421 | 2 | UB30502 | 24 | 12500 |
| 147 | KHR | CSE461 | 2 | UB10303 | 25 | 13920 |
| 148 | SRT | CSE331 | 2 | UB30601 | 25 | 160 |
| 149 | ACH | CSE340 | 2 | UB30501 | 25 | 3400 |
| 150 | ZAR | CSE422 | 1 | UB30603 | 25 | 12480 |
| 151 | HOS | CSE370 | 4 | UB10304 | 25 | 4200 |
| 152 | TBA | CSE460 | 1 | UB30303 | 25 | 0 |
| 153 | KHR | CSE461 | 1 | UB30601 | 26 | 13340 |
| 154 | MAH | CSE320 | 2 | UB40301 | 26 | 819 |
| 155 | MSA | CSE111 | 3 | UB30501 | 26 | 720 |
| 156 | MSN | CSE260 | 5 | UB30503 | 26 | 297 |
| 157 | MMM | CSE221 | 2 | UB30502 | 26 | 483 |
| 158 | DMH | CSE423 | 2 | UB30603 | 26 | 12420 |
| 159 | KHR | CSE101 | 1 | UB30601 | 27 | 440 |
| 160 | MHR | CSE340 | 5 | UB10304 | 27 | 816 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 161 | MSA | CSE111 | 1 | UB10303 | 27 | 1760 |
| 162 | TRH | CSE470 | 4 | UB30603 | 27 | 240 |
| 163 | HAL | CSE330 | 6 | UB30303 | 27 | 360 |
| 164 | HOS | CSE370 | 1 | UB30501 | 27 | 9240 |
| 165 | MMD | CSE320 | 7 | UB30603 | 28 | 364 |
| 166 | MSN | CSE111 | 4 | UB10303 | 28 | 252 |
| 167 | NUS | CSE220 | 5 | UB30502 | 28 | 252 |
| 168 | HAL | CSE330 | 2 | UB40301 | 28 | 945 |
| 169 | ACH | CSE340 | 4 | UB30501 | 28 | 2380 |
| 170 | MMD | CSE320 | 1 | UB10304 | 29 | 1040 |
| 171 | RAK | CSE221 | 4 | UB30603 | 29 | 140 |
| 172 | AAR | CSE110 | 4 | UB30501 | 29 | 360 |
| 173 | TRH | CSE470 | 1 | UB40301 | 29 | 180 |
| 174 | JIU | CSE341 | 2 | UB10303 | 29 | 7200 |
| 175 | MKR | CSE470 | 2 | UB30601 | 29 | 360 |
| 176 | MAH | CSE101 | 10 | UB10304 | 30 | 35 |
| 177 | BIS | CSE321 | 1 | UB10303 | 30 | 210 |
| 178 | SLI | CSE220 | 2 | UB30502 | 30 | 30 |
| 179 | MMM | CSE471 | 1 | UB30501 | 30 | 1767 |
| 180 | ADR | CSE391 | 1 | UB30503 | 30 | 8360 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 181 | IFF | CSE420 | 2 | UB30603 | 30 | 456 |
| 182 | MMD | CSE410 | 1 | UB30503 | 31 | 368 |
| 183 | MHR | CSE330 | 1 | UB30501 | 31 | 1620 |
| 184 | HAL | CSE230 | 1 | UB40301 | 31 | 432 |
| 185 | MAM | CSE230 | 2 | UB30502 | 31 | 640 |
| 186 | JIU | CSE341 | 3 | UB30601 | 31 | 1440 |
| 187 | KHR | CSE461 | 3 | UB10304 | 32 | 9860 |
| 188 | MHR | CSE340 | 1 | UB30503 | 32 | 306 |
| 189 | HAL | CSE230 | 3 | UB40301 | 32 | 72 |
| 190 | ZAR | CSE331 | 3 | UB30502 | 32 | 5440 |
| 191 | AKC | CSE470 | 3 | UB30603 | 32 | 510 |
| 192 | MMD | CSE101 | 9 | UB10304 | 33 | 8 |
| 193 | DIP | CSE110 | 1 | UB30603 | 33 | 6 |
| 194 | MHR | CSE101 | 6 | UB30501 | 33 | 12 |
| 195 | BIS | CSE321 | 3 | UB30601 | 33 | 672 |
| 196 | MSA | CSE111 | 5 | UB10303 | 33 | 1280 |
| 197 | TRH | CSE221 | 3 | UB30303 | 33 | 112 |
| 198 | KHR | CSE101 | 8 | UB10304 | 34 | 20 |
| 199 | DIP | CSE101 | 4 | UB30502 | 34 | 1 |
| 200 | RAK | CSE101 | 5 | UB30501 | 34 | 15 |
| 201 | AIZ | CSE260 | 4 | UB30603 | 34 | 11 |
| 202 | SRT | CSE310 | 1 | UB30303 | 34 | 180 |
| 203 | MIK | CSE320 | 6 | UB40301 | 34 | 1365 |

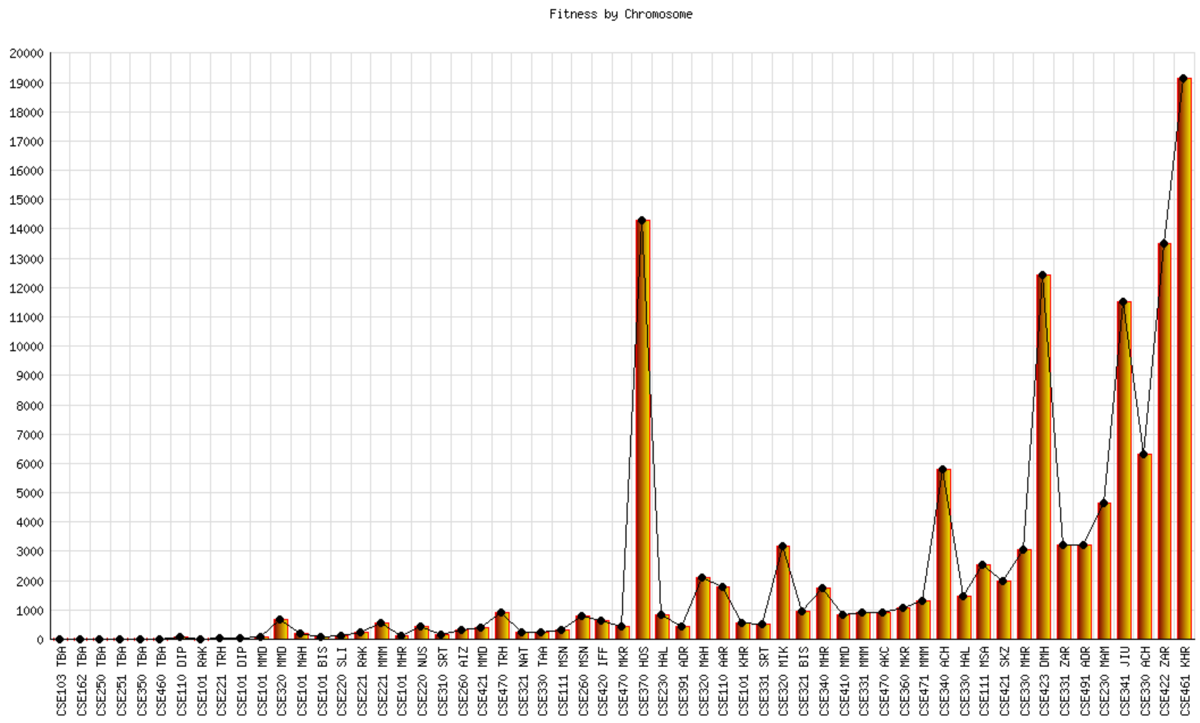
**Course Allocation Per Instructor**

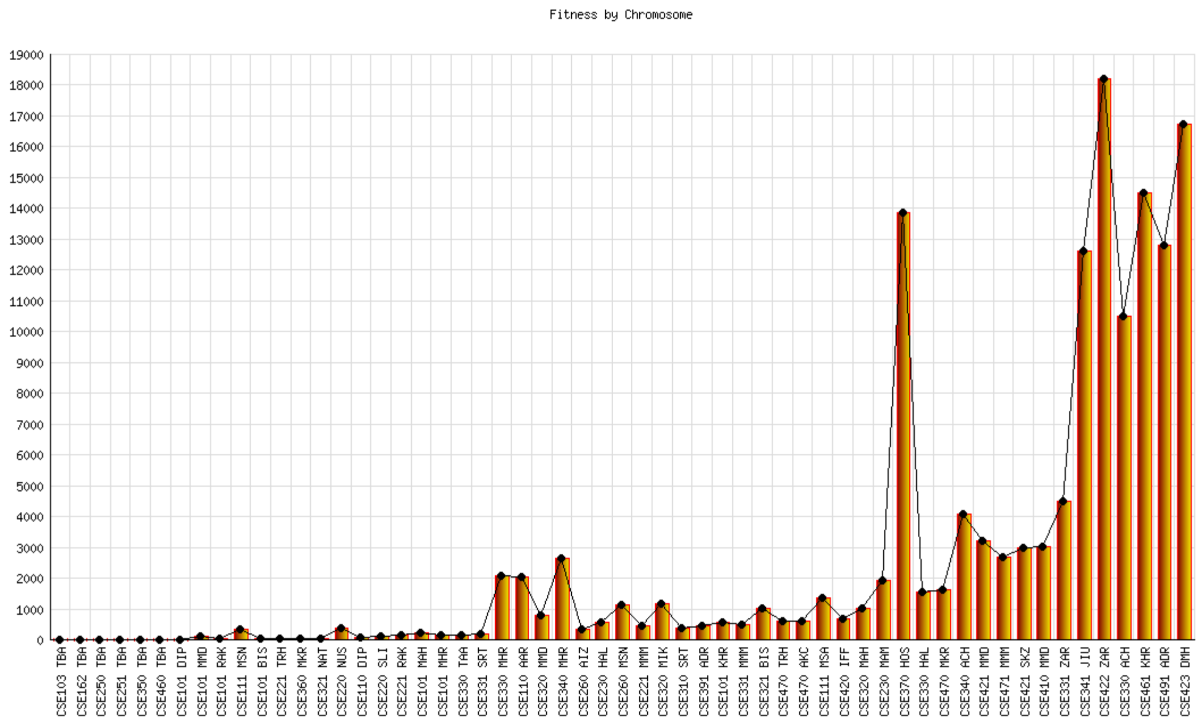




In this figure, each and every teacher are represented in $x-$ axis and number of courses allocated to them plotted in $y-$ axis. In two different graph we can see one particular teacher can have different number of courses which are randomly allocated to them. The courses for which no teachers are assigned for them the count is 0.

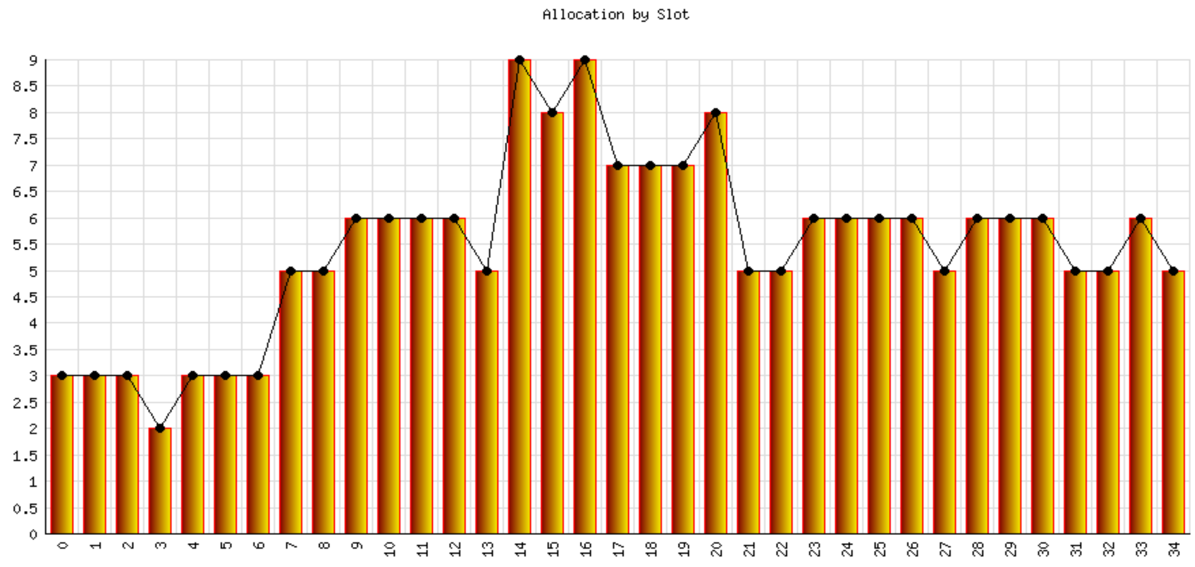
## Chromosomes Fitness

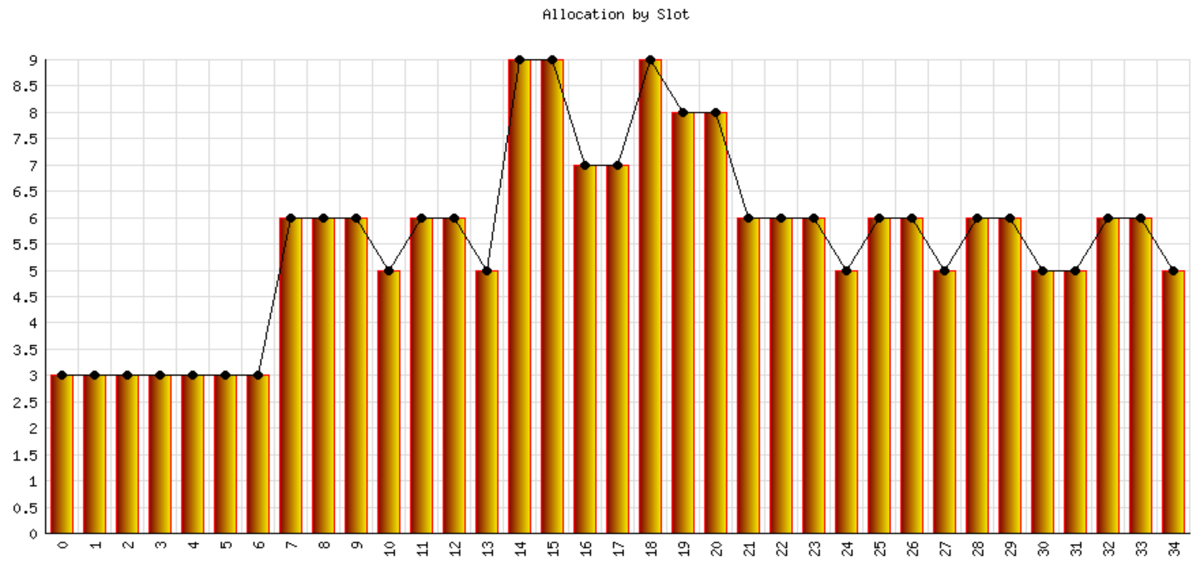




A fitness for chromosomes indicates the probability to be selected for next phase of the routine from a randomly generated population. The graph is plotted such a way that $x-$ axis represented particular course allocated to teacher and $y-$ axis represents fitness value for them. If the new population has higher total fitness regarding the old one only then the resulted routine will be updated.

**Number of Courses Assigned Per Slot**





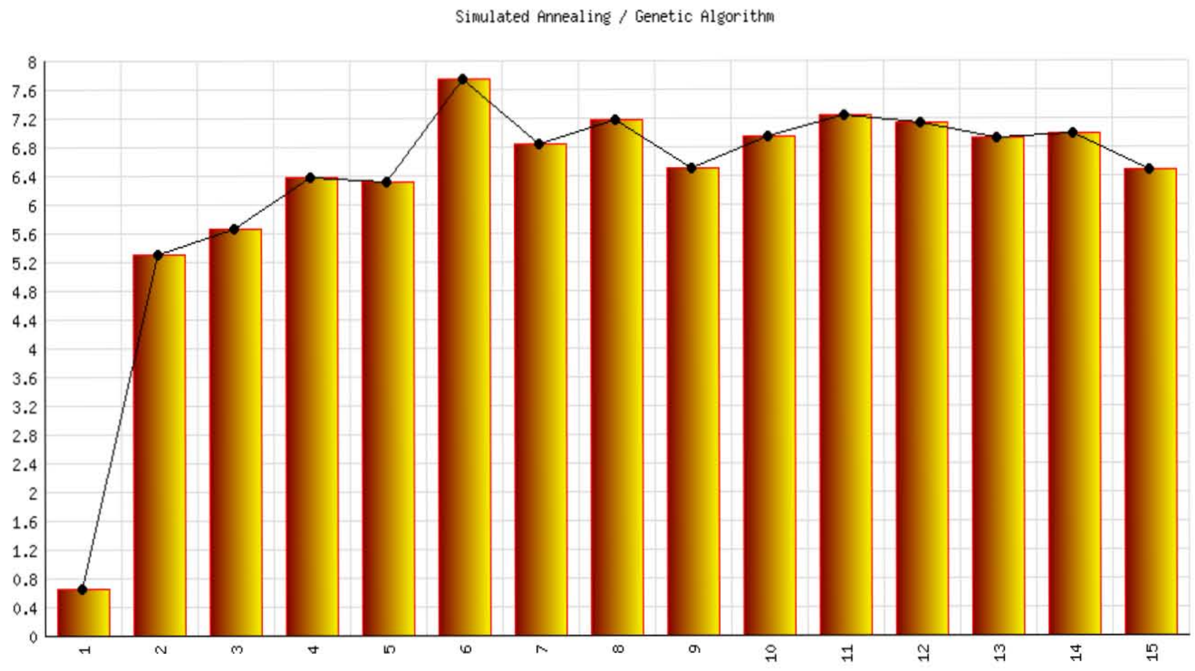
Along with $X$ axis slots are plotted and along with $Y$ axis number of allocations are plotted in the figure. This graph helps us to find out the demand of a slot according to the corresponding count. Also it becomes very useful for modifying mutation design of the main program.

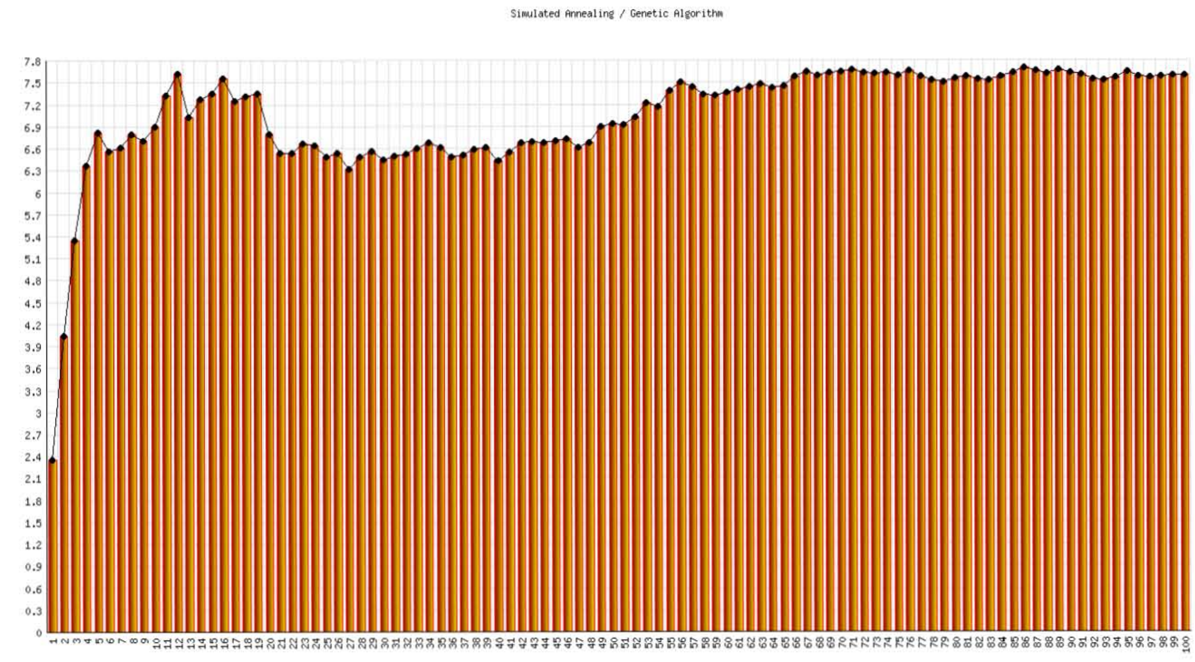
## 7.2 Result Comparison

Results from Simulated Annealing gets lower fitness than the Genetic Algorithm generated Schedule most of the time. For comparing the results two different fitness were calculated. One from Genetic Algorithm generated result and another from Simulated Annealing generated result. Finally the comparing fitness will be calculated by using formula:-

$$C_f = \frac{(GA_{fitness} - SA_{fitness})}{GA_{fitness}} * 100\%$$

If comparing fitness $C_f$ is positive then Genetic Algorithm has produced better result than the Simulated Annealing process, and if the $C_f$ is negative then Simulated Annealing is better in Scheduling.for every routine generation Genetic Algorithm and Simulated annealing has run simultaneously same number of times times as Genetic Algorithm count and in about more than 100 times run Simulated Annealing gives one better result than Genetic Algorithm.


Simulated Annealing / Genetic Algorithm


Simulated Annealing / Genetic Algorithm

# Chapter 8

# Future Work

Our study was only over Computer Science and Engineering Department of BRAC University to find and optimal solution of course scheduling. There are lots of scope over all the departments to be included under one schedule generation project. We are eager to implement many other features in our next work. Some are like:

- For further optimization and better fitness and hybrid model can be proposed.
- The shortcomings after further analysis is to be overcome.
- As we overlooked lab and exam scheduling the scope is open for these two categories to be integrated.
- Enrich User Interface for better performance and prepared version for remote devices.
- Secure data by decentralizing and distributing in different databases.

# Chapter 9

# Conclusion

In this paper, our main focus was to design a procedure, based on Genetic algorithm for generating a schedule for BRAC University CSE Department automatically. There are a lot of works done on this type of scheduling before but the main difference from them is, our work is only for BRAC University CSE department where we used our BRAC life experience and modified the algorithm accordingly to get a better and feasible schedule. We also developed our selection process in such unique way that the main program would need to run crossover and mutation process less amount of times, which increases the convergence rate of new schedules and decrease the new schedule generation time. Finally we tried to compare our results with other heuristic algorithms to ensure the acceptance of the newly generated schedule.

# Reference

[1] Al-Betar, M. A., and Khader, A. T. A harmony search algorithm for university course timetabling. *Annals of Operations Research 194*, 1 (2012), 3–31.

[2] Aycan, E. *Solving the course scheduling problem by constraint programming and simulated annealing.* 2008.

[3] Boyan, J. A., and L.Littman, M. Packet routing in dynamically changing networks: A reinforcement learning approach. *Advances in neural information processing systems* (1994), 671–671.

[4] Bratković, Z., Herman, T., Omrčen, V., Čupić, M., and Jakobović, D. University course timetabling with genetic algorithm: A laboratory excercises case study. In *Evolutionary Computation in Combinatorial Optimization.* Springer, 2009, pp. 240–251.

[5] Chen, R.-M., and Shih, H.-F. Solving university course timetabling problems using constriction particle swarm optimization with local search. *Algorithms 6*, 2 (2013), 227–244.

[6] Colorni, A., Dorigo, M., and Maniezzo, V. A genetic algorithm to solve the timetable problem. *Politecnico di Milano, Milan, Italy TR* (1992), 90–060.

[7] Cooper, T. B., and Kingston, J. H. *The complexity of timetable construction problems.* Springer, 1996.

[8] Jain, A., Jain, D. S., and Chandre, D. P. Formulation of genetic algorithm to generate good quality course timetable. *International Journal of Innovation, Management and Technology 1*, 3 (2010).

[9] Jat, S. N., and Yang, S. A guided search genetic algorithm for the university course timetabling problem.

[10] Jha, S. K. Exam timetabling problem using genetic algorithm. *International Journal of Research in Engineering and Technology* (2014).

[11] Kohshori, M. S., and Abadeh, M. S. Hybrid genetic algorithms for university course timetabling. *International Journal of Computer Science issues* (2012).

[12] Lakshmi, R., Vivekanandhan, K., and Brintha, R. A new biological operator in genetic algorithm for class scheduling problem. *International Journal of Computer Applications (0975 8887) Vol 60* (2012), 6–11.

[13] LARGET, H. Genetic algorithms used in timetable management.

[14] Lewis, R., Paechter, B., and Rossi-Doria, O. *Metaheuristics for university course timetabling*. Springer, 2007.

[15] Lukas, S., Aribowo, A., and Muchri, M. *Solving Timetable Problem by Genetic Algorithm and Heuristic Search Case Study: Universitas Pelita Harapan Timetable*. INTECH Open Access Publisher, 2012.

[16] Nuntasen, N., and Innet, S. Application of genetic algorithm for solving university timetabling problems: A case study of thai universities. *UTCC Engineering Research Papers* (2007).

[17] Qin, G., and Ma, H. An intelligent course scheduling model based on genetic algorithm. *TELKOMNIKA Indonesian Journal of Electrical Engineering 12*, 4 (2014), 2985–2994.

[18] Setyaningsih, F. A. System application of genetic algorithm for scheduling optimization study using java (case study: Department of computer system untan). *IPTEK Journal of Proceedings Series 1*, 1 (2014).

[19] Sigl, B., Golub, M., and Mornar, V. Solving timetable scheduling problem using genetic algorithms. In *Proc. of the 25th int. conf. on information technology interfaces* (2003), pp. 519–524.

[20] Watkins, C. J., and Dayan, P. Q-learning. *Machine learning 8*, 3-4 (1992), 279–292.

[21] West, J. M., and Antonio, J. K. A genetic algorithm approach to scheduling communications for a class of parallel space-time adaptive processing algorithms. In *Parallel and Distributed Processing.* Springer, 2000, pp. 855–861.

[22] Wren, A. *Scheduling, timetabling and rostering –A special relationship. In Practice and Theory of Automated Timetabling.* 1996.

[18] [16] [1] [5] [14] [12] [21] [17] [10]