CONSTRAINT OPTIMIZATION OF ROUTINE ASSIGNMENT PROBLEM

Sharif Tamjidur Rahman
Student Id: 011 133 054
Haider Ali
Student Id: 011 133 051
Fahmida Mim
Student Id: 011 141 012
Nafija Rahman Sharna
Student Id: 011 151 185
Mustary Ahmed
Student Id: 011 151 015

A project in the Department of Computer Science and Engineering presented in partial fulfillment of the requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

United International University

Dhaka, Bangladesh

June 2019

Declaration

We, Sharif Tamjidur Rahman, Haider Ali, Fahmida Mim, Nafija Rahman Sharna, Mustary Ahmed declare that this project titled and the work presented in it are all our own. We confirm that:

- This work was done wholly or mainly while in candidature for a BSc degree at United International University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at United International University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourselves jointly with others, we have made clear exactly what was done by others and what we have contributed by our self.

_____

Sharif Tamjidur Rahman, 011 133 054, Department of CSE

_____

Haider Ali, 011 133 051, Department of CSE

_____

Fahmida Mim, 011 141 012, Department of CSE

_____

Nafija Rahman Sharna, 011 151 185, Department CSE

_____

Mustary Ahmed, 011 151 015, Department of CSE

Certificate

I do hereby declare that the research works embodied in this thesis/project entitled "**CONSTRAINT OPTIMIZATION OF ROUTINE ASSIGNMENT PROBLEM**" is the outcome of an original work carried out by Sharif Tamjidur Rahman, Haider Ali, Fahmida Mim, Nafija Rahman Sharna and Mustary Ahmed under my supervision.

I further certify that the dissertation meets the requirements and the standard for the degree of BSc in Computer Science and Engineering.

_____

Associate Professor Dr. Swakkhar Shatabda, Department of CSE.

Abstract

We consider a simple form of a solve of a typical university problem which occur while registration of courses in university timetable.We involved three types of hard and soft constraints. We will provide an appropriate construction graph and matrix representation.The result will demonstrate that our approach is able to produce solutions that have lower penalty on all the similar problems in the literature.At first we will satisfy all the hard constraints.Then we will satisfy all the soft constraints.Hard and soft constraints are the part of cost function.Cost function will calculate penalty movements.And penalty movement will add a cost.We will try to find out the lowest cost solution.We will try to apply traveling salesman or hamiltonian path or minimum spanning tree or shortest path graph algorithm etc..After satisfying all soft constraint we will apply optimization technique.We will optimize the routine by using local search approach.Such as, simulated annealing or gradient descent or ant min max or ant colony algorithm etc.We will apply several algorithms each time and compare the result of the previous attempt.Then we will choose the best solution.Solution will auto generate and excel sheet of optimized routine.

Acknowledgement

We are very much thankful to our honorable faculties for rendering us the data we need. We also thank CSE 400: Final Year Design Project teacher Professor Dr. A K M Muzahidul Islam and our advisor Associate Professor Dr. Swakkhar Shatabda, Department of CSE.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

In this chapter the motivations and problem statement have been discussed. What is the problem which is faced and using what methods can be used to solve this have been organized in this report. So, this chapter reveals the problem and motivation behind the project.

1.1 Problem Statement

In every semester during registration of courses and scheduling United International University faces problems like section drop, excessive student, very less student etc. The task of scheduling is done manually so these problems occur. The solve of this problem should be like there will always be a system which will generate the routine automatically. Thus, there will be less problems like section drop because system will generate the best time table for the student. And the students will be sufficient so that there is no less student or excessive student in a section.

1.2 Motivation

The motivation behind this problem is to build an automated system which will save the human resources for more productive tasks because the existing solutions are not enough yet. So, there should be more optimized solution for scheduling. And, thus there will be no occurrences of section drop and time conflict.

1.3 Objective

Every work done has some objectives. Goals drive us towards success. This project has some strong objectives to fulfill.

- To identify algorithm(s), which work best for schedule optimization.
- To automate Teacher Assignment.
- To automate Room/Lab Assignment.

- To automate Scheduling.

1.4 Methodology

First, we have done the requirement analysis where we searched for the functional and non -functional requirements to develop this problem. After that we gathered up some research papers and journals and studied them and compared them. Different people used different algorithms having different datasets and constraints to solve this problem. Then we further analyzed our problem to find the scopes of development. We collected data from all the departments on the faculty lists, courses and sections and room usage. We organized the data in excel sheets and analyzed them. After that we developed the algorithm and developed the web application. After testing and verification we were ready to deploy the web tool.

1.5 Our Contribution

Different universities used different algorithms to solve their routine assignment problem because they have different constraints to bind them. We found out that Simulated annealing and Genetic algorithm is the best fit for our problem. So, we developed the algorithm in Python with our own soft and hard constraints and restrictions which serves our own dataset best. We collected data of all the departments showing the most usage of rooms and congestions which helps find out the problem better.

1.6 Organization of the Report

The report contains all the topics required to develop this project. First, a small introduction is given where the problem is described. Then comes the background and literature review where we describe our registration process and what problem it faces. There is data visualization to show the congestion of room and courses. Then there is a detailed literature review. After that the methodology is described. There is an SRS and multiple diagrams. Then we show our work progress and our website design and tools. Then there is conclusion and future work. At the end of the report there is references and appendix.

Chapter 2

Background and Literature Review

It's very important to know about the background of a project before understanding it. Backgrounds tell a lot about a project's motivations, goals and targets. In this chapter our literature reviews and background has been described.

2.1 Background

The background of this project is to find the best optimized solutions as many other projects have been done on this routine assignment problem. Every semester students and faculties of United International University face problems like section drop due to insufficient students, sometimes course drop also happens here. And sometimes there are section time conflicts. So students can not take their desired courses. Moreover, it is found that a student has 3 classes in a particular day which is quite pressure to him. And exam conflicts also happen here. On the other hand, our teachers do these tasks of scheduling manually so their important time and human resources are wasted. So our project is to find out an automated system where there will be no scheduling conflicts and human resources are saved.

2.1.1 The registration process

Every trimester the students have to go through a registration process for selecting courses. Primarily three people are involved in this process Figure 1. First, in the pre-advising process the Department Resource Planner fills up an excel sheet with course names, sections and time schedule Figure 2(a). Next in the section selection process the room numbers are assigned by the University Resource Planner shown in Figure 2(b). Finally, the Department Course Coordinator fills the Faculty/Teacher column shown in Figure 2(c) and completes the registration process.

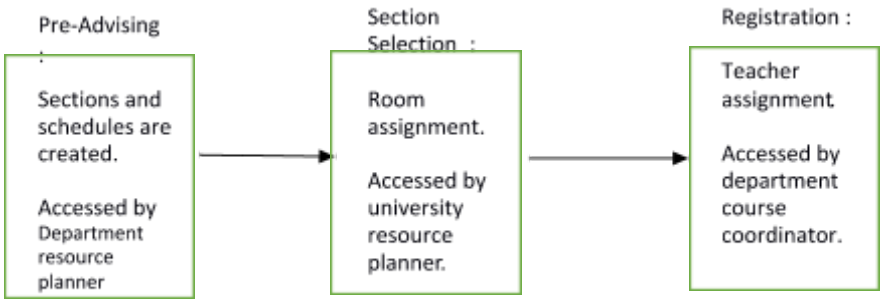
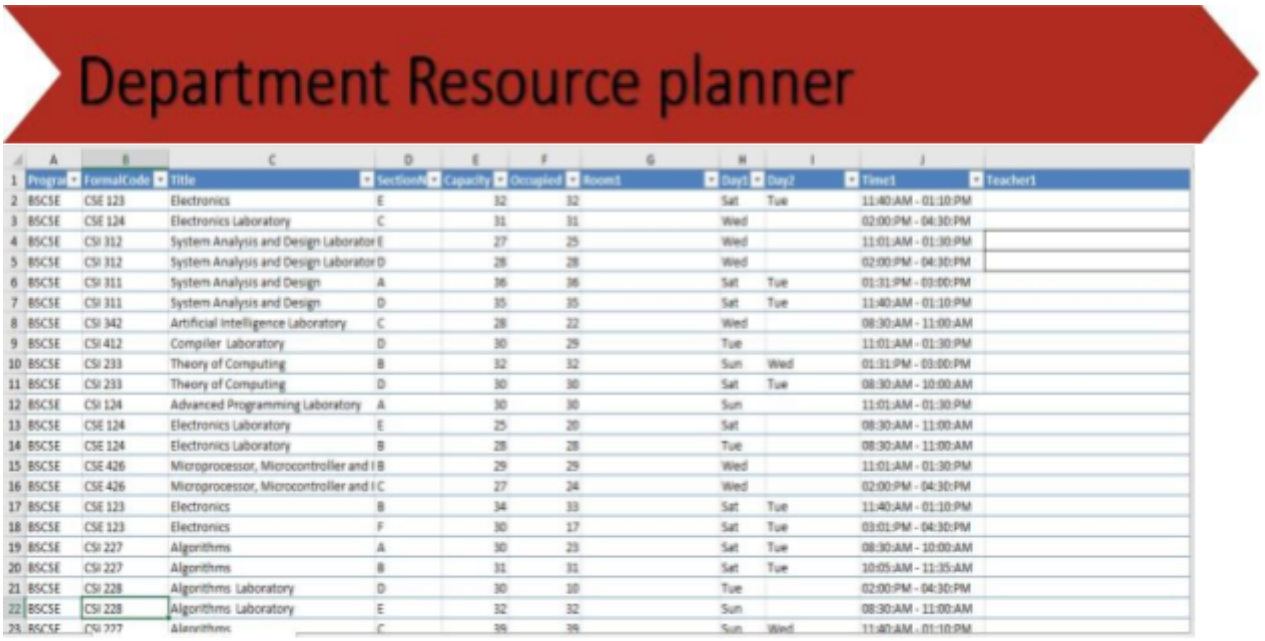
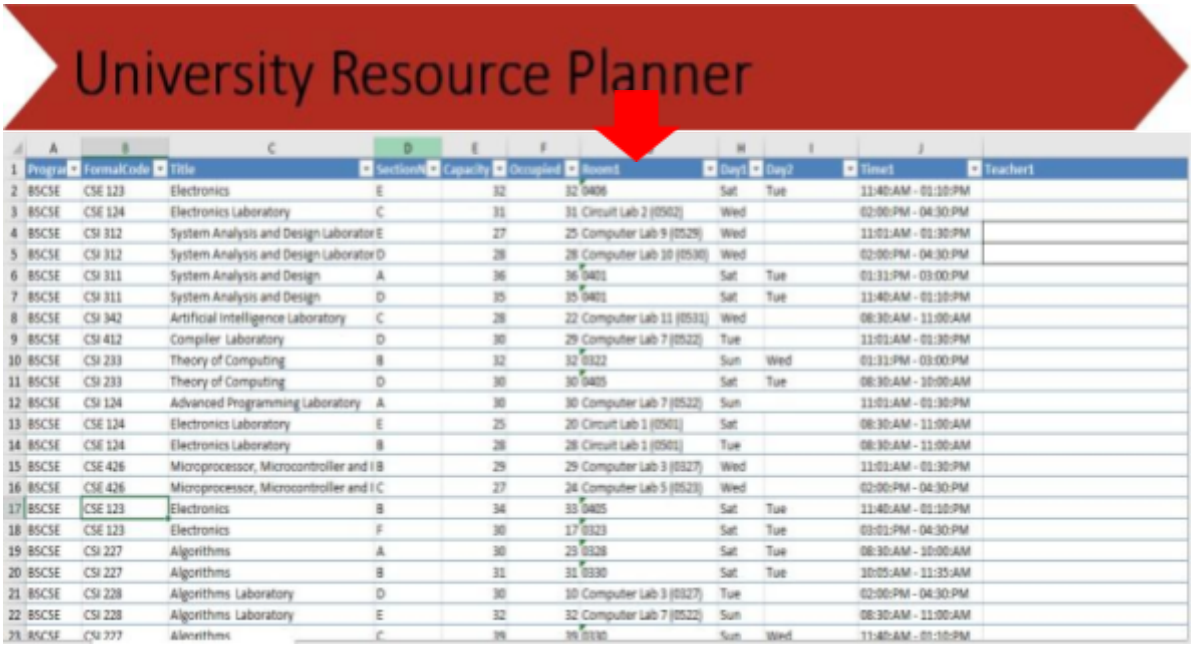


Figure 1: Block Diagram of registration Process.



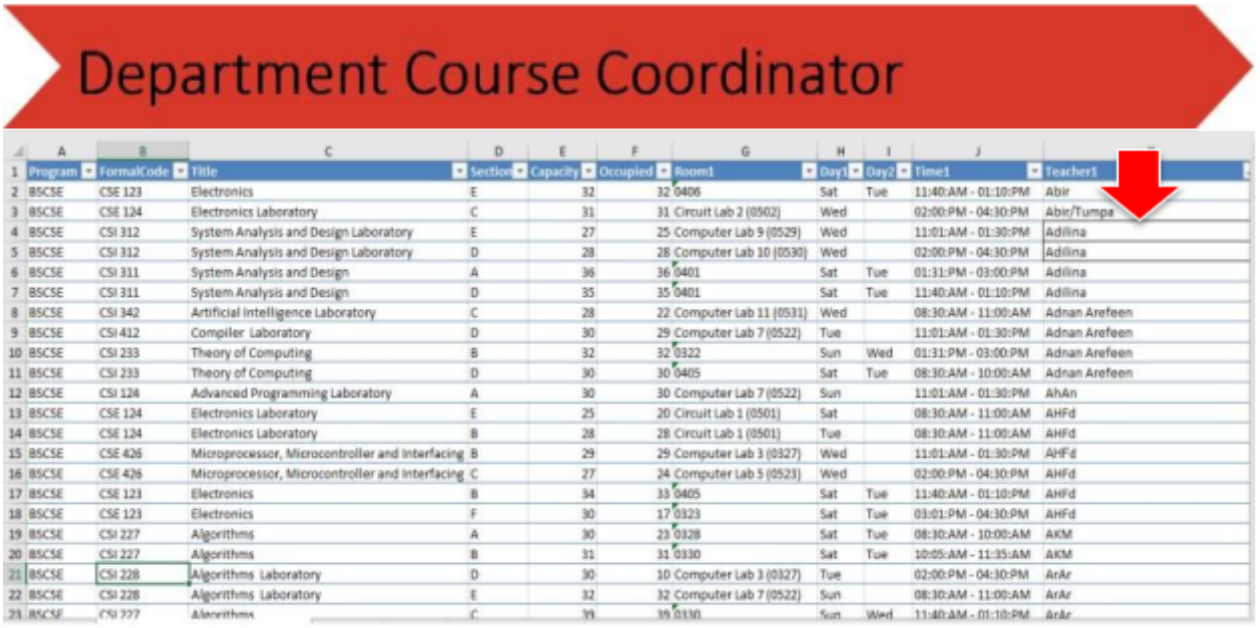
	A	B	C	D	E	F	G	H	I	J	
1	Progra	FormalCode	Title	SectionN	Capacity	Occupied	Room1	Day1	Day2	Time1	Teacher1
2	BSCSE	CSE 123	Electronics	E	32	32		Sat	Tue	11:40:AM - 01:10:PM	
3	BSCSE	CSE 124	Electronics Laboratory	C	31	31		Wed		02:00:PM - 04:30:PM	
4	BSCSE	CSI 312	System Analysis and Design Laborator	E	27	25		Wed		11:01:AM - 01:30:PM	
5	BSCSE	CSI 312	System Analysis and Design Laborator	D	28	28		Wed		02:00:PM - 04:30:PM	
6	BSCSE	CSI 311	System Analysis and Design	A	36	36		Sat	Tue	01:31:PM - 03:00:PM	
7	BSCSE	CSI 311	System Analysis and Design	D	35	35		Sat	Tue	11:40:AM - 01:10:PM	
8	BSCSE	CSI 342	Artificial Intelligence Laboratory	C	28	22		Wed		08:30:AM - 11:00:AM	
9	BSCSE	CSI 412	Compiler Laboratory	D	30	29		Tue		11:01:AM - 01:30:PM	
10	BSCSE	CSI 233	Theory of Computing	B	32	32		Sun	Wed	01:31:PM - 03:00:PM	
11	BSCSE	CSI 233	Theory of Computing	D	30	30		Sat	Tue	08:30:AM - 10:00:AM	
12	BSCSE	CSI 124	Advanced Programming Laboratory	A	30	30		Sun		11:01:AM - 01:30:PM	
13	BSCSE	CSE 124	Electronics Laboratory	E	25	20		Sat		08:30:AM - 11:00:AM	
14	BSCSE	CSE 124	Electronics Laboratory	B	28	28		Tue		08:30:AM - 11:00:AM	
15	BSCSE	CSE 426	Microprocessor, Microcontroller and I	B	29	29		Wed		11:01:AM - 01:30:PM	
16	BSCSE	CSE 426	Microprocessor, Microcontroller and I	C	27	24		Wed		02:00:PM - 04:30:PM	
17	BSCSE	CSE 123	Electronics	B	34	33		Sat	Tue	11:40:AM - 01:10:PM	
18	BSCSE	CSE 123	Electronics	F	30	17		Sat	Tue	03:01:PM - 04:30:PM	
19	BSCSE	CSI 227	Algorithms	A	30	23		Sat	Tue	08:30:AM - 10:00:AM	
20	BSCSE	CSI 227	Algorithms	B	31	31		Sat	Tue	10:05:AM - 11:35:AM	
21	BSCSE	CSI 228	Algorithms Laboratory	D	30	10		Tue		02:00:PM - 04:30:PM	
22	BSCSE	CSI 228	Algorithms Laboratory	E	32	32		Sun		08:30:AM - 11:00:AM	
23	BSCSE	CSI 227	Algorithms	C	39	39		Sun	Wed	11:40:AM - 01:10:PM	

Figure 2.1(a): Department Resource Planner Excel Sheet.



	A	B	C	D	E	F	G	H	I	J	
1	Progra	FormalCode	Title	SectionN	Capacity	Occupied	Room1	Day1	Day2	Time1	Teacher1
2	BSCSE	CSE 123	Electronics	E	32	32	0406	Sat	Tue	11:40:AM - 01:10:PM	
3	BSCSE	CSE 124	Electronics Laboratory	C	31	31	Circuit Lab 2 (0502)	Wed		02:00:PM - 04:30:PM	
4	BSCSE	CSI 312	System Analysis and Design Laborator	E	27	25	Computer Lab 9 (0529)	Wed		11:01:AM - 01:30:PM	
5	BSCSE	CSI 312	System Analysis and Design Laborator	D	28	28	Computer Lab 10 (0530)	Wed		02:00:PM - 04:30:PM	
6	BSCSE	CSI 311	System Analysis and Design	A	36	36	0401	Sat	Tue	01:31:PM - 03:00:PM	
7	BSCSE	CSI 311	System Analysis and Design	D	35	35	0401	Sat	Tue	11:40:AM - 01:10:PM	
8	BSCSE	CSI 342	Artificial Intelligence Laboratory	C	28	22	Computer Lab 11 (0531)	Wed		08:30:AM - 11:00:AM	
9	BSCSE	CSI 412	Compiler Laboratory	D	30	29	Computer Lab 7 (0522)	Tue		11:01:AM - 01:30:PM	
10	BSCSE	CSI 233	Theory of Computing	B	32	32	0322	Sun	Wed	01:31:PM - 03:00:PM	
11	BSCSE	CSI 233	Theory of Computing	D	30	30	0405	Sat	Tue	08:30:AM - 10:00:AM	
12	BSCSE	CSI 124	Advanced Programming Laboratory	A	30	30	Computer Lab 7 (0522)	Sun		11:01:AM - 01:30:PM	
13	BSCSE	CSE 124	Electronics Laboratory	E	25	20	Circuit Lab 1 (0501)	Sat		08:30:AM - 11:00:AM	
14	BSCSE	CSE 124	Electronics Laboratory	B	28	28	Circuit Lab 1 (0501)	Tue		08:30:AM - 11:00:AM	
15	BSCSE	CSE 426	Microprocessor, Microcontroller and I	B	29	29	Computer Lab 3 (0327)	Wed		11:01:AM - 01:30:PM	
16	BSCSE	CSE 426	Microprocessor, Microcontroller and I	C	27	24	Computer Lab 5 (0523)	Wed		02:00:PM - 04:30:PM	
17	BSCSE	CSE 123	Electronics	B	34	33	0405	Sat	Tue	11:40:AM - 01:10:PM	
18	BSCSE	CSE 123	Electronics	F	30	17	0323	Sat	Tue	03:01:PM - 04:30:PM	
19	BSCSE	CSI 227	Algorithms	A	30	23	0328	Sat	Tue	08:30:AM - 10:00:AM	
20	BSCSE	CSI 227	Algorithms	B	31	31	0330	Sat	Tue	10:05:AM - 11:35:AM	
21	BSCSE	CSI 228	Algorithms Laboratory	D	30	10	Computer Lab 3 (0327)	Tue		02:00:PM - 04:30:PM	
22	BSCSE	CSI 228	Algorithms Laboratory	E	32	32	Computer Lab 7 (0522)	Sun		08:30:AM - 11:00:AM	
23	BSCSE	CSI 227	Algorithms	C	39	39	0330	Sun	Wed	11:40:AM - 01:10:PM	

Figure 2.1(b): University Resource Planner Excel Sheet.



	A	B	C	D	E	F	G	H	I	J	
1	Program	FormalCode	Title	Section	Capacity	Occupied	Room1	Day1	Day2	Time1	Teacher1
2	BSCSE	CSE 123	Electronics	E	32	32	0406	Sat	Tue	11:40:AM - 01:10:PM	Abir
3	BSCSE	CSE 124	Electronics Laboratory	C	31	31	Circuit Lab 2 (0502)	Wed		02:00:PM - 04:30:PM	Abir/Tumpa
4	BSCSE	CSI 312	System Analysis and Design Laboratory	E	27	25	Computer Lab 9 (0529)	Wed		11:01:AM - 01:30:PM	Adilina
5	BSCSE	CSI 312	System Analysis and Design Laboratory	D	28	28	Computer Lab 10 (0530)	Wed		02:00:PM - 04:30:PM	Adilina
6	BSCSE	CSI 311	System Analysis and Design	A	36	36	0401	Sat	Tue	01:31:PM - 03:00:PM	Adilina
7	BSCSE	CSI 311	System Analysis and Design	D	35	35	0401	Sat	Tue	11:40:AM - 01:10:PM	Adilina
8	BSCSE	CSI 342	Artificial Intelligence Laboratory	C	28	22	Computer Lab 11 (0531)	Wed		08:30:AM - 11:00:AM	Adnan Arefeen
9	BSCSE	CSI 412	Compiler Laboratory	D	30	29	Computer Lab 7 (0522)	Tue		11:01:AM - 01:30:PM	Adnan Arefeen
10	BSCSE	CSI 233	Theory of Computing	B	32	32	0322	Sun	Wed	01:31:PM - 03:00:PM	Adnan Arefeen
11	BSCSE	CSI 233	Theory of Computing	D	30	30	0405	Sat	Tue	08:30:AM - 10:00:AM	Adnan Arefeen
12	BSCSE	CSI 124	Advanced Programming Laboratory	A	30	30	Computer Lab 7 (0522)	Sun		11:01:AM - 01:30:PM	AhAn
13	BSCSE	CSE 124	Electronics Laboratory	E	25	20	Circuit Lab 1 (0501)	Sat		08:30:AM - 11:00:AM	AHFd
14	BSCSE	CSE 124	Electronics Laboratory	B	28	28	Circuit Lab 1 (0501)	Tue		08:30:AM - 11:00:AM	AHFd
15	BSCSE	CSE 426	Microprocessor, Microcontroller and Interfacing	B	29	29	Computer Lab 3 (0327)	Wed		11:01:AM - 01:30:PM	AHFd
16	BSCSE	CSE 426	Microprocessor, Microcontroller and Interfacing	C	27	24	Computer Lab 5 (0523)	Wed		02:00:PM - 04:30:PM	AHFd
17	BSCSE	CSE 123	Electronics	B	34	33	0405	Sat	Tue	11:40:AM - 01:10:PM	AHFd
18	BSCSE	CSE 123	Electronics	F	30	17	0323	Sat	Tue	03:01:PM - 04:30:PM	AHFd
19	BSCSE	CSI 227	Algorithms	A	30	23	0328	Sat	Tue	08:30:AM - 10:00:AM	AKM
20	BSCSE	CSI 227	Algorithms	B	31	31	0330	Sat	Tue	10:05:AM - 11:35:AM	AKM
21	BSCSE	CSI 228	Algorithms Laboratory	D	30	10	Computer Lab 3 (0327)	Tue		02:00:PM - 04:30:PM	ArAr
22	BSCSE	CSI 228	Algorithms Laboratory	E	32	32	Computer Lab 7 (0522)	Sun		08:30:AM - 11:00:AM	ArAr
23	BSCSE	CSI 227	Algorithms	C	39	39	0330	Sun	Wed	11:40:AM - 01:10:PM	ArAr

Figure 2.1(c): Department Course Coordinator Excel Sheet.

2.2 Visualizing data after registration process

We will draw some histograms about the routines of BSCSE,BSEEE and BBA department.So that we can easily visualize the problems

2.2.1 Data visualized for BSCSE Department

Section and Room occupancy

We have visualized section per room for BSCSE program by histogram. So that we can notice over pressured room numbers.

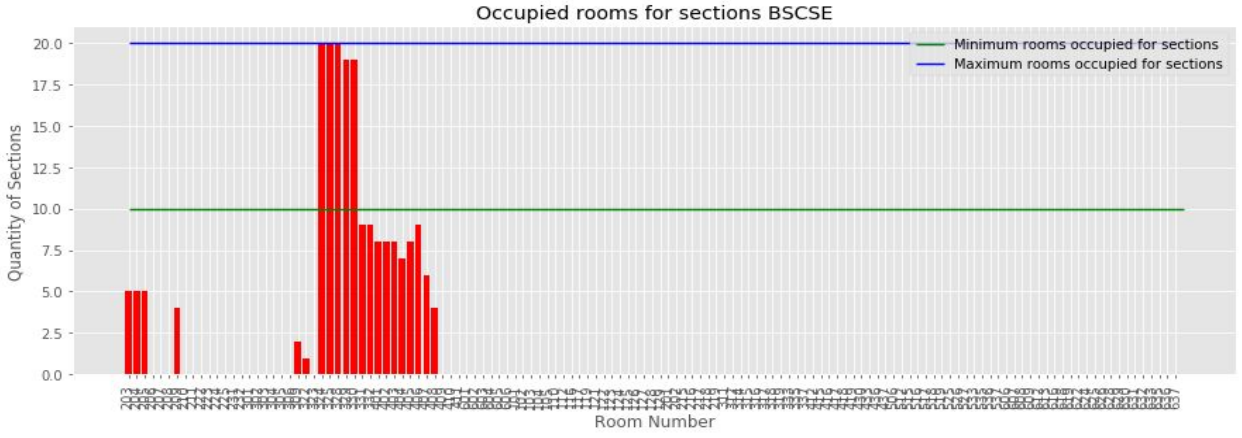


Figure 2.2(a) : Section and Room occupancy for BSCSE Department

Section and laboratory occupancy

We have visualized section per laboratory for BSCSE program by histogram. So that we can notice over pressured laboratory numbers.

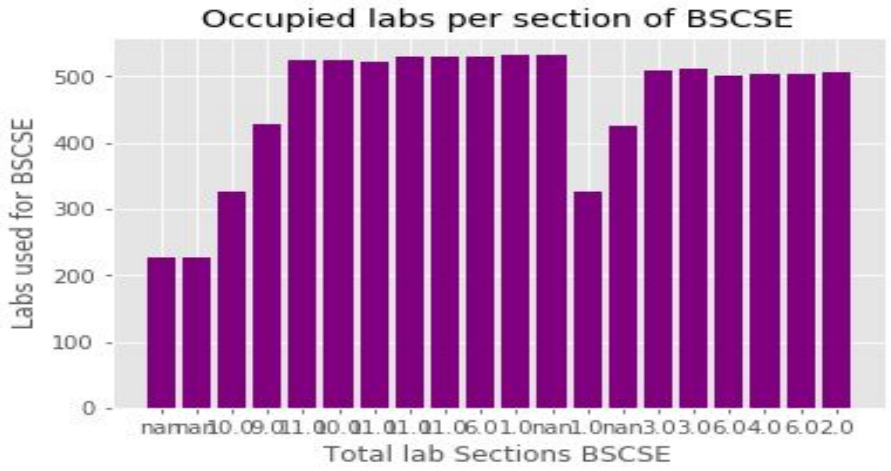


Figure 2.2(b) : Section and laboratory occupancy for BSCSE Department

Section of courses and Faculty occupancy

We have visualized course per faculties for BSCSE program by histogram.So that we can manage optimum routine for faculty members.

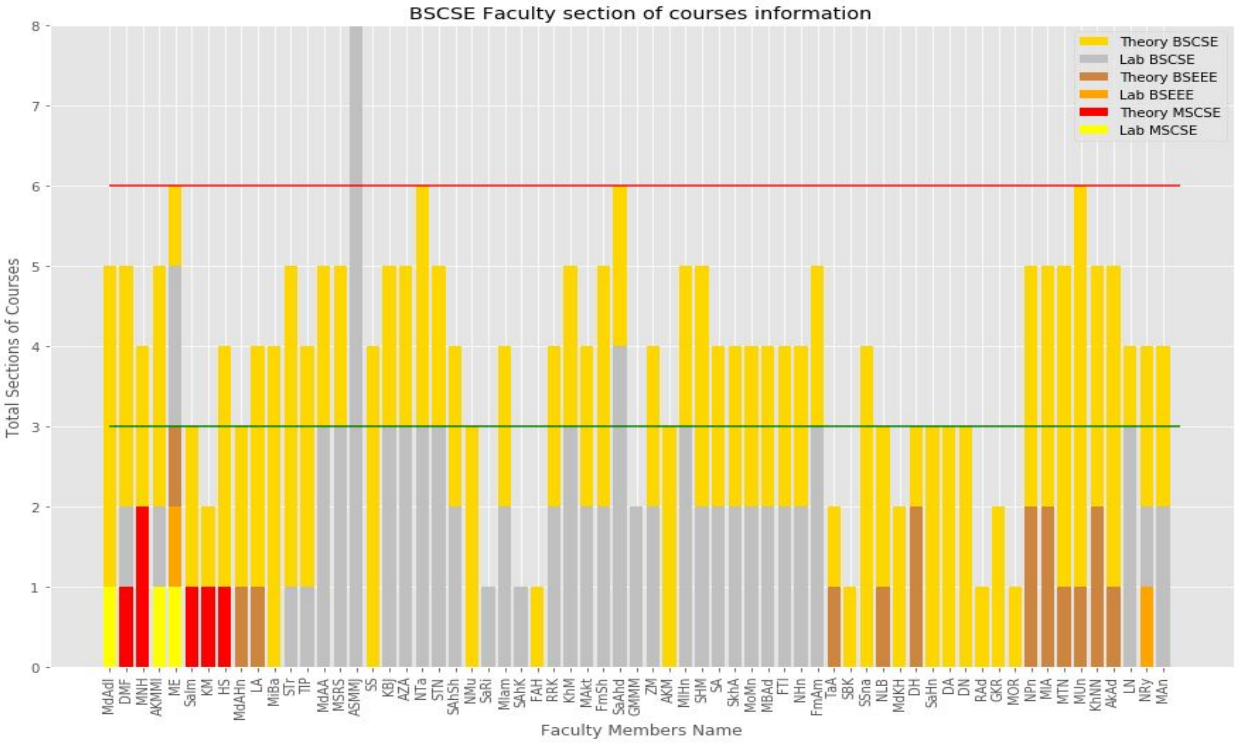


Figure 2.2(c) : Faculty and course occupancy for BSCSE Department

Faculty and Credits information

In this histogram we can identify each and every faculties taken courses.

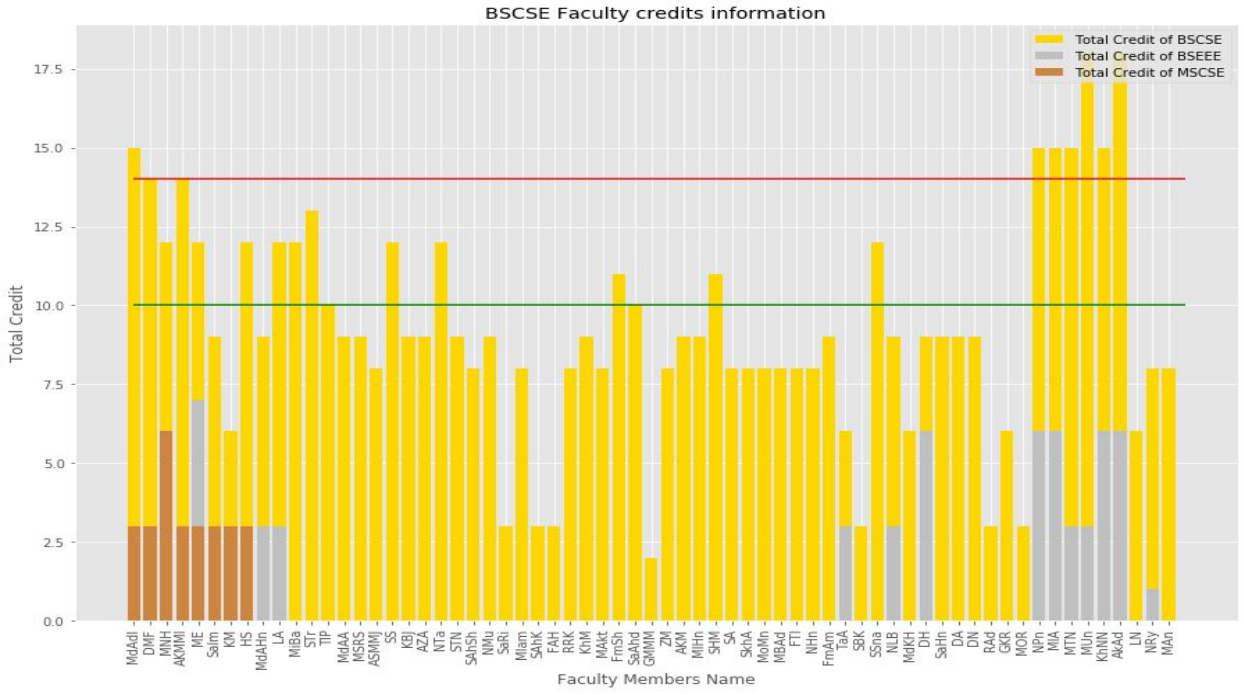


Figure 2.2(d) : Faculty and course occupancy for BSCSE Department

2.2.2 Data visualized for BSEEE Department

Section and Room occupancy

We have visualized section per room for BSEEE program by histogram.So that we can notice over pressured room numbers.

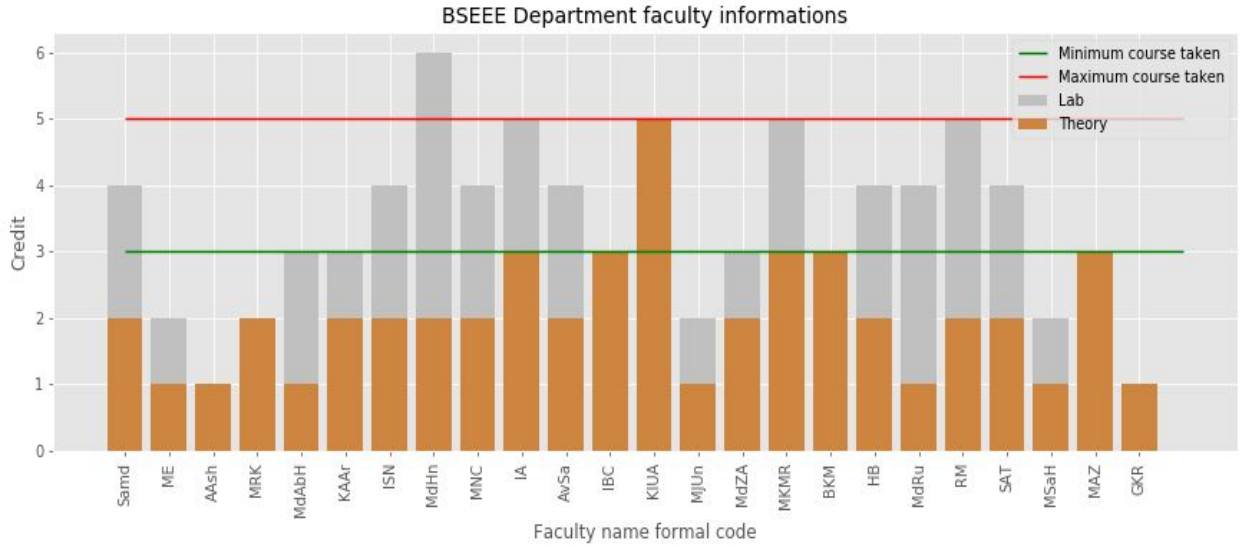


Figure 2.2(e) : Course and Room occupancy for BSEEE Department

Course and Faculty occupancy

We have visualized course per faculties for BSEEE program by histogram. So that we can manage optimum routine for faculty members.

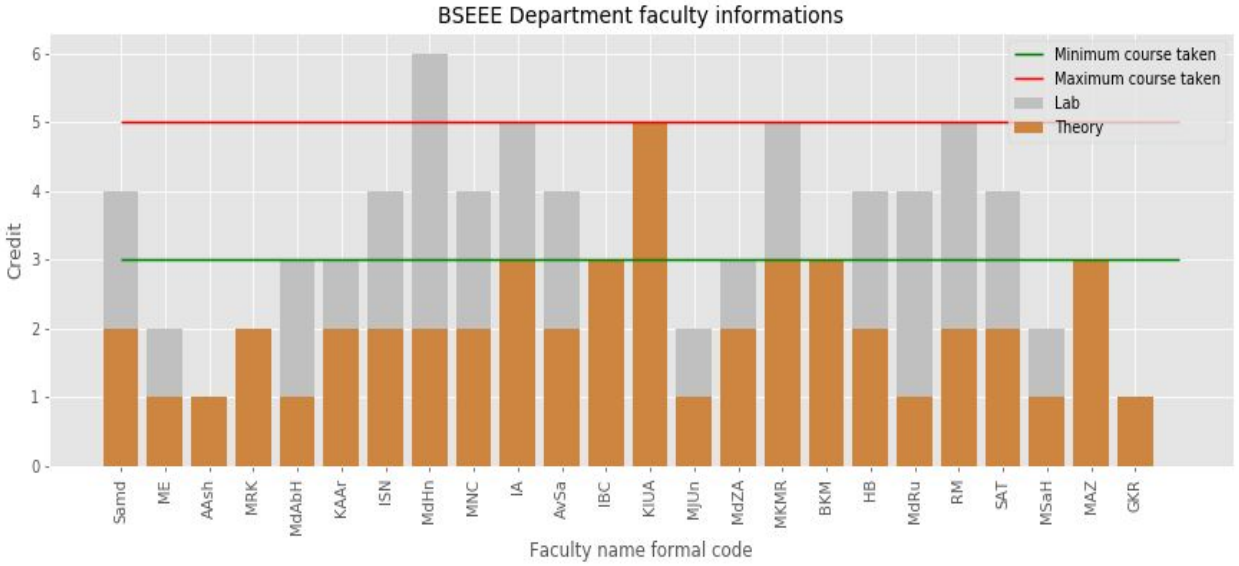


Figure 2.2(f) : Faculty and course occupancy for BSEEE Department

2.2.3 Data visualized for BBA Department

Section and Room occupancy

We have visualized section per room for BBA program by histogram. So that we can notice over pressured room numbers.

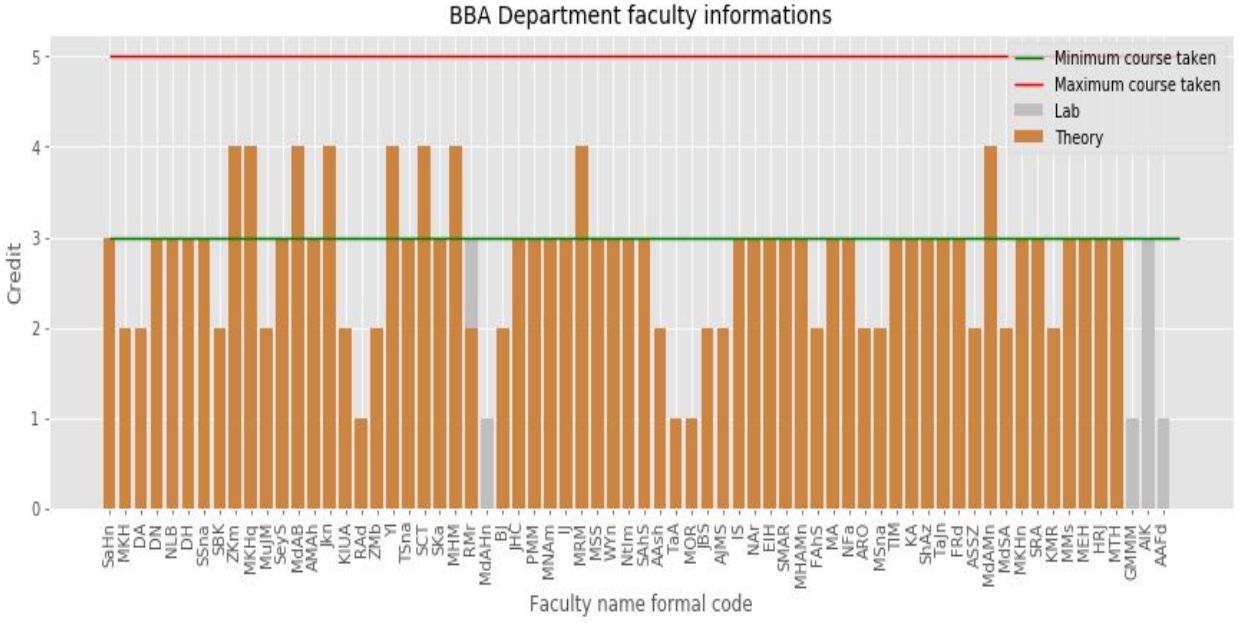


Figure 2.2(g) : Section and Room occupancy for BBA Department

Course and Room occupancy

We have visualized course per room for BSCSE program by histogram.So that we can notice over pressured room numbers.

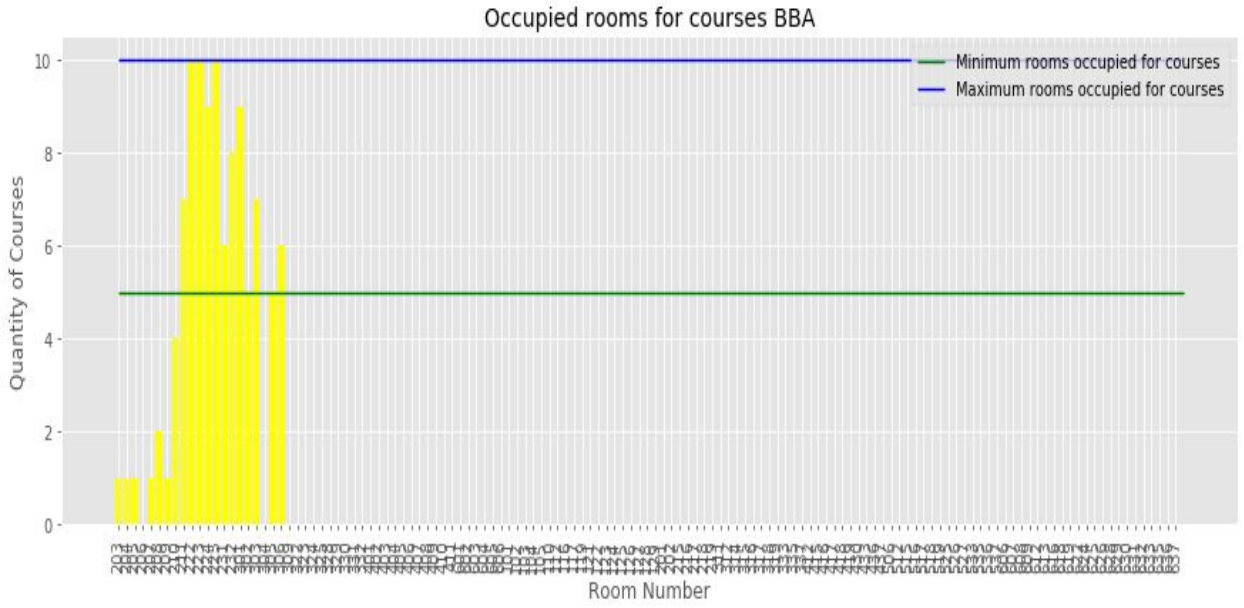


Figure 2.2(h) : Course and Room occupancy for BBA Department

Course and Faculty occupancy

We have visualized course per faculties for BBA program by histogram. So that we can manage optimum routine for faculty members.

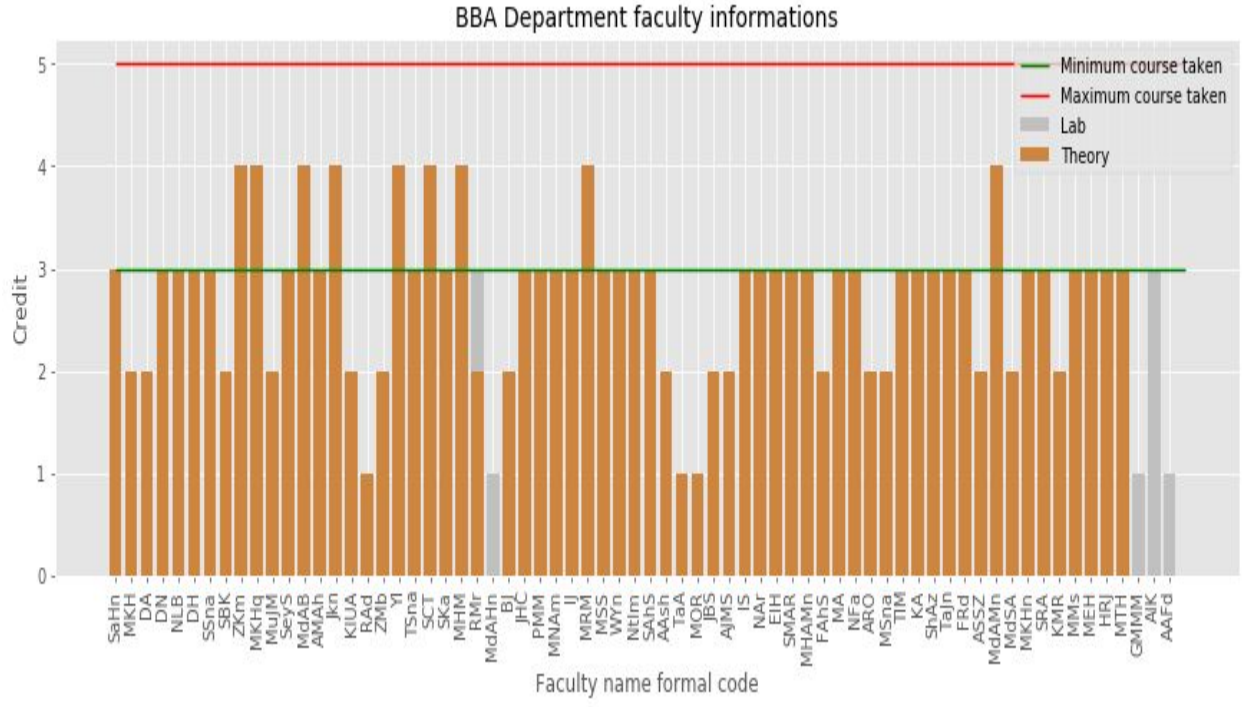


Figure 2.2(i) : Faculty and course occupancy for BBA Department

2.2.4 All Department Lab and Course visualization

Here Figure 2.2(l), we have visualized the lab and course information for all departments so that we can get a better idea about the congestion in the labs and which labs need more attention.

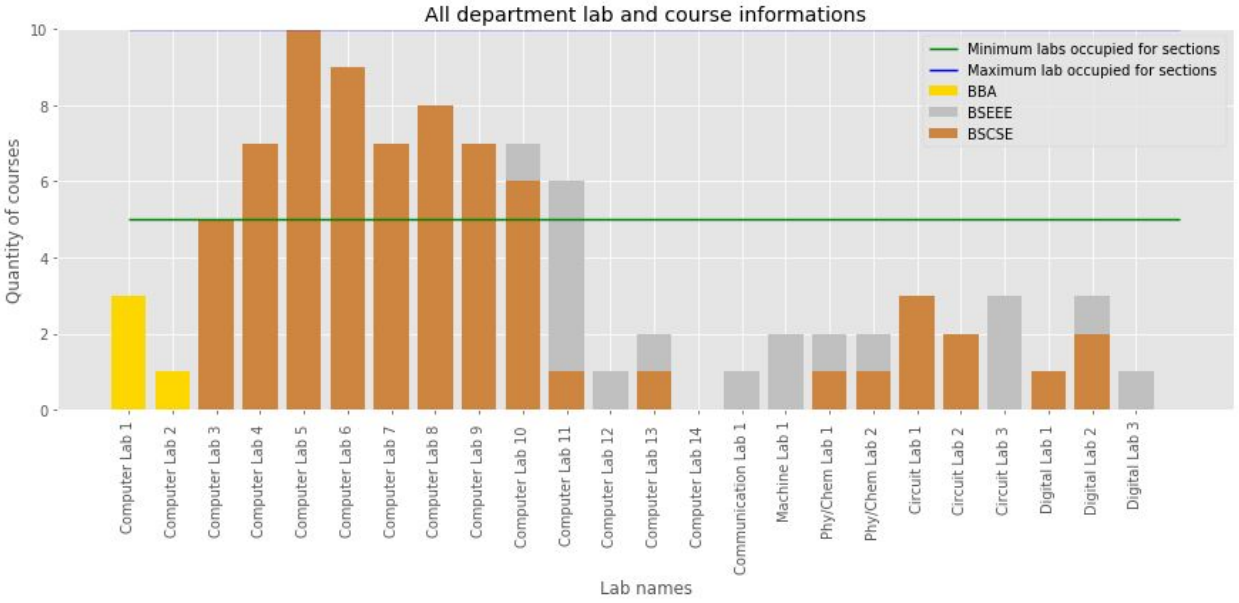


Figure 2.2(j) : Lab and course visualization for All Department

2.2.5 All Department Lab and Section Visualization

Here Figure 2.2(m), we have visualized the lab and section information for all departments so that we can get a better idea about the congestion in the labs and which labs need more attention.

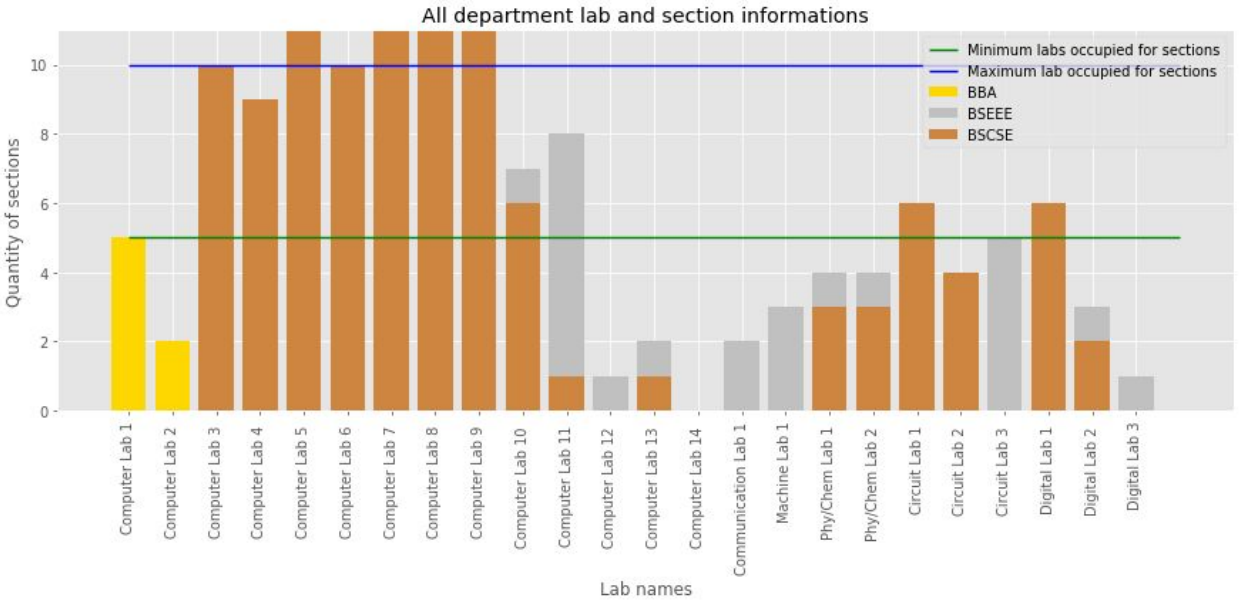


Figure 2.2(k) : Lab and Section visualization for All Departments

2.3 Visualization Summary

We can easily notice that,For the all departments some rooms,some faculties,has excessive pressure. So that,And some faculties and rooms has less pressure. So that we will solve the situation by optimizing the entire routine.

2.4 Literature Review

Here we have listed Table 1 and briefly described all the literatures and papers we have studied for this project. There are in total 19 papers organized in the Table 1 below.

Table 1 : Literature Review

Sl	Paper Title	Algorithm	Publisher	Year	Key Factors
1	Using a Randomized Iterative Improvement Algorithm with Composite Neighbourhood Structures for the University Course Timetabling Problem [1]	Randomized Iterative Improvement Algorithm	Springer	2007	1. Composite neighborhood structure 2. Hard and soft constraints 3. Large problem and graph based hyper heuristic
2	A MAX-MIN Ant System For the University Course Timetabling Problem [2]	MAX-MIN Ant System	Springer	2002	1. 3 types of hard and 3 types of soft constraints 2. Appropriate construction graph 3. Pheromone matrix
3	A hybrid evolutionary approach to the university course timetabling problem [3]	Randomised Iterative Improvement Algorithm	IEEE	2007	1. Evolutionary operator: mutation 2. Population generation and solution 3. Hybrid evolutionary approach
4	Parallel Ant Colony Optimization on the University Course-Faculty Timetabling Problem in MSU-IIT [4]	Ant colony optimization,max-min ant system, parallel algorithm	GSTF Journal of Computing	2018	1. Distributed application 2. Search space graph, parallel and distributed architecture 3. Search parameter calibration 4. Pheromone Trail Mode

Literature Review Continue

Sl	Paper Title	Algorithm	Publisher	Year	Key Factors
5	Modelling the university malaysia pahang examination timetabling problem [5]	Randomised Iterative Improvement Algorithm	IJSECS	2018	1. Examination timetabling 2. Constraints Mathematical model
6	Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem [6]	Simulated Annealing with Reheating	Journal of the Operational Research Society	2018	1. Two stage approach 2. Improved reheating and learning 3. Reinforcement learning and suitable neighborhood 4. Tabu search with sampling and perturbation (TSSP)
7	A Generic Object Oriented Constraint Based Model for University Course Timetabling [7]	Tabu search, simulated annealing, genetic algorithms, constraint programming, graph coloring	Springer	2000	1. Soft constraint, Hard constraint 2. Search Methods: 3. Iterative broadening (IB) 4. Depth-bounded discrepancy search (DDS) 5. Constraint programming object-oriented 6. Object-Oriented Modeling
8	A memetic algorithm for University Course Timetabling [8]	Memetic algorithm for local search (evolutionary computation, tabu search, simulated annealing)	metaheuristic ics.net	2004	1. Metaheuristics 2. NP hard 3. Re-interpretation and improvement 4. Evolution process: 5. Steady-State evolution model

Literature Review Continue

Sl	Paper Title	Algorithm	Publisher	Yea r	Key Factors
9	Adaptive Selection of Heuristics for Improving Exam Timetables. [9]	hyper heuristic approach with hybridised low level heuristics	Springer	2014	1. Hyper-heuristics 2. kempe chain heuristic 3. time slot swapping heuristic 4. low-level heuristic
10	Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art [10]	Ant-colony system,MAX MIN Ant system	Springer	2003	1. Compare Metaheuristics 2. Reference random restart local search 3. NP hard
11	Hardness Prediction for the University Course Timetabling Problem [11]	MAX-MIN Ant system,Ant Colony Optimization	Springer	2004	1. Statistical model: a.Normal linear models b.Casting our problem into a normal model framework 2. Metaheuristics 3. particular hard instance 4. Fitting the Model: a. 6-fold cross-validation b. Simulated Annealing
12	Hyper-heuristics: a survey of the state of the art [12]	Hyper-heuristics, Evolutionary computation, combinatorial Optimisation, Scheduling, Metaheuristics, machine learning	Taylor & Francis	2013	1. Hyper-heuristics, metaheuristics 2. Combinatorial optimization, scheduling 3. Evolutionary computation

Literature Review Continue

Sl	Paper Title	Algorithm	Publisher	Yea r	Key Factors
13	Memetic Algorithms for Solving University Course Timetabling Problem [13]	Graph coloring with hill climbing approach, hybrid genetic algorithm	Ieeeexplo re . ieee.org	2011	1. Course timetabling 2. Memetic algorithm 3. graph colouring
14	Multi-Criteria Soft Constraints in Timetabling [14]	Constraint Satisfaction problem	unitime.o rg	2005	1. Soft constraint 2. Weight: weighted constraint 3. criteria, multi criteria
15	University Course Timetabling Using Constraint Handling Rules [15]	Constraint handling rules by performing soft constraint propagation	Taylor & Francis	2000	1. Partial constraint satisfaction. 2. Concise finite domain solver 3. Active soft constraints 4. Constraint satisfaction: 5. Constraint Logic Programming 6. (CLP)
16	University Course Timetabling with Probability Collectives [16]	Probability collective	Calhoun. np s .edu	2008	1. Novel approach 2. Agent based timetabling 3. Post enrollment timetabling solution

Literature Review Continue

Sl	Paper Title	Algorithm	Publisher	Yea r	Key Factors
17	University Course Timetabling with Soft Constraints [17]	Constraint logic programming	Springer	2002	1. Soft constraints, weighted constraints 2. Constraint propagation 3. Logical constraints 4. Constraint logic programming
18	A Simulated Annealing for Tahmidi Course Timetabling [18]	Simulated annealing	Elsevier	2013	1. Fitness function 2. Hard and soft constraints 3. Scheduling, timetabling
19	Metaheuristics for University Course Timetabling [19]	Grouping Genetic Algorithm (GGA), Iterated Heuristic Search algorithm	Springer	2007	1. Evolutionary algorithm (Iterated Heuristic Search algorithm) 2. Simulated annealing (Iterated Heuristic Search algorithm) 3. Relaxations, Search space

Brief Discussion:

We have reviewed 19 papers about university schedule optimization problem. There are m16 thesis papers and 3 journals. For routine optimization they have used basically 4 process.

1.Local Search (Simulated Annealing. Genetic Algorithm, Ant min max, Ant colony optimization, tabu search, graph coloring) [2] [4] [7] [8] [11] [18]
2. Constraint Satisfaction (Focused on soft constraints) [14] [15] [17]
3. Mathematical tool(probability) [16]
4. Randomized Iterative Improvement Algorithm [1][3][5]

They have used some other techniques:

1. Evolutionary Computation [12]
2. Combinatorial optimization [12]
3. Scheduling [12]
4. Hyper Heuristics [9] [12]

Some Advancements:

1. Grouping genetic algorithm (GGA) [19]
2. Advanced Reheating for simulated annealing [6]
3. Graph coloring algorithm with hill climbing approach [13]
4. Iterated Heuristic Search Algorithm [19]

2.5 Literature Review Summary

We have learnt about different algorithms reviewing 19 papers so far. They contain a lot of information and algorithms. In the Table 2 below, we have summarized the main theme of the papers and categorized them with respect to the algorithms they used to solve routine optimization problem.

Table 2: Literature Review Summary.

Sl	Paper Title	Algorithm
1	a. Parallel Ant Colony Optimization on the University Course-Faculty Timetabling Problem in MSU-IIT. b. A MAX-MIN Ant System for the University Course Timetabling Problem. c. Hardness Prediction for the University Course Timetabling Problem. d. Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art.	a. Ant colony optimization, b. max-min ant system, c. parallel algorithm
2	a. Multi-Criteria Soft Constraints in Timetabling. b. Timetabling Using Constraint Handling Rules. c. University Course Timetabling Using Constraint Handling Rules.	a. Constraint logic programming b. rules by performing soft constraint propagation c. Constraint Satisfaction problem

	d. University Course Timetabling with Soft Constraints.	

Literature Review Summary Continue

Sl	Paper Title	Algorithm
3	University Course Timetabling with Probability Collectives **.**	Probability collective
4	a. Adaptive Selection of Heuristics for Improving Exam Timetables. b. Hyper-heuristics: a survey of the state of the art.	hyper heuristic approach with hybridizes low-level heuristics
5	a. Using a Randomized Iterative Improvement Algorithm with Composite Neighborhood Structures for the University Course Timetabling Problem. b. A hybrid evolutionary approach to the university course timetabling problem. c. Modelling the university malaysia pahang examination timetabling problem.	Randomized Iterative Improvement Algorithm.
6	a. Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem. b. A Generic Object Oriented Constraint Based Model for University Course Timetabling. c. A Simulated Annealing for Tahmidi Course Timetabling. d. Metaheuristics for University Course Timetabling.	a. Grouping Genetic. b. Algorithm (GGA). c. Iterated Heuristic Search algorithm. d.Simulated Annealing with Reheating.
7	a.A memetic algorithm for University Course Timetabling. b.Memetic Algorithms for Solving University Course Timetabling Problem.	a. Memetic algorithm for local search (evolutionary computation, tabu search, simulated annealing). b. Graph coloring with hill climbing approach, hybrid genetic algorithm.

We have chosen Iterative Neighborhood algorithm, Simulated annealing algorithm, Genetic algorithm for our project. The reasons behind choosing these algorithms are described below.

Why simulated Annealing Algorithm:

We will try to optimize the dataset by the motivation of metallurgy and materials science because its modification is easy by improved reheating.

Chapter 3

Methodology

In this chapter we are going to discuss about our methodology. Methodology is actually the process for completing the project and necessary tasks for giving it a conclusion. So the methodology will be elaborately discussed here.

3.1 Software Requirements Specification

This document lays out a project plan for the development of "Constraint Optimization of Routine Assignment Problem". The intended readers of this document are current and future developers working on "Constraint Optimization of Routine Assignment Problem". The plan will include, but is not restricted to a summary of the system functionality, the scope of the project from the perspective of the "Constraint Optimization of Routine Assignment Problem" team (me, my team members and my mentor), scheduling and delivery estimates, project risks and how those risks will be mitigated, the process by which we will develop the project will be recorded throughout the project.

3.1.1 Overview

Manual systems put pressure on people to be correct in all details of their work at all times, the problem being that people aren't perfect. With manual systems the level of service is dependent on individuals. We aim to develop an application that would enable them to invest their time to do other productive works.

3.1.2 Purpose

The purpose of SRS Table 3 is to present a detailed description of constraint routine optimization. It will explain the purpose and features of our system used here and how the system will work, what type of algorithms will be used here and how this system will be operated. This document is intended for both stakeholders and developers of the system.

3.1.3 Scope

In our system our university timetable scheduling can be hazard free. Every semester our university resource planners Table 3, faculties have to face a lot of hazards while scheduling our course classes and timetables. Moreover, their time is wasted to these irrelevant tasks while their time could be used in more productive tasks. As the scheduling is done manually so many conflicts occur here like section drop, over populated sections etc. To solve this problem, we came up with the idea of an automated system. Here our system will automatically generate classroom and time for courses. And this system will also assign teacher as per their preference. This system will solve the scheduling problems every semester and save our department resource planner's time for more beneficial tasks.

3.1.4 Goals

After the completion of this project we aspire to fulfill some specific goals. Some of the goals are listed below.

- No course drop, section drop.
- No overloaded sections.
- No insufficient students in a section.
- No conflict in examination time.
- No conflict in class schedule.
- No human Resource wastage due to manual process.

Table 3: Terms and definitions.

Term	Definition
SRS	Software requirement specification
Stakeholders	Anyone who has interaction and interest in this project
Department course coordinator	Someone who assigns teachers to courses
University Resource Planner	Someone who assigns rooms
Department Resource planner	Someone who creates sections and schedules

3.2 Overall Description

Here we have described the overall process elaborately to provide as much as information about our project we can in an organized way.

3.2.1 Users

There will be mainly three users of our web tool who are named Department course coordinator, University Resource Planner, Department Resource planner Table 3.

3.2.2 Functionality

It is important to understand how our web tool will function. Down below we have listed the functionality of our web tool.

• User would be able to assign teacher automatically.
• Course will be assigned automatically to teachers according to their preferences.
• No Section or course drop.
• No conflict in exam time.
• No conflict in class schedule.

3.2.3 Constraints

The system has several constraints to abide by. These constraints make it harder to optimize but following these constraints will make the routine more useful and efficient.

Hard Constraints

Hard constraints are those which can not be violated at any cost. The constraints mentioned below are our hard constraints.

• There cannot be two classes of the same faculty at the same time.
• There cannot be two classes in the same room at the same time.
• A faculty cannot have several classes at the same time.
• Two faculties cannot have the same section of the same course.

Soft Constraints

Soft constraints are temperamental, so we can ignore them sometimes but we will try our best to fulfill as many as soft constraints we can. Some soft constraints are listed below

Soft constraints for faculty optimization

A faculty should not have two consecutive class.
A faculty should not have more than three classes in a single day.
A faculty should not have more than two classes and a lab in a single day.
A faculty should not have more than one class and two labs in a single day.
A faculty should not have more than three labs in a single day.

A faculty should not have two consecutive laboratory classes.
A faculty should not take less than ten credit of courses in a single semester.
A faculty should not take more than fourteen credit of courses in a single semester.
A faculty has to join four day at work in a week.

Soft constraints for laboratory optimization:

Spl,Apl,Data structure course,Assembly,CSE 1110,Programming contest should be held in same labs.
Ai,Pattern,Simulation,Programming contest should be held in same labs.
OOP,Algorithm,Mobile Application development,CDIP should be held in same labs.
CN,Cloud,OS,Compiler,Cisco should be held in same lab.
SAD,Software,Web,Capstone,Graphics,CDIP,DBMS should be in same labs.
Labs minimum running section : 4(Of various types of course).
Labs maximum running section : 7(Of various types of course).
4 sections of each course will be held in 4 different days of working day in a week.

Soft constraints for room optimization:

BSCSE's class will primarily held of 4th floor.(Both departmental and non departmental)
BSEEE's class will primarily held on 3rd floor.(Both departmental and non departmental)
BBA's class will primarily held on 2nd floor.(Both departmental and non departmental)
Rooms minimum section limit 10.
Rooms maximum section limit 20.

Soft constraints for section optimization:

3 to 4 section needs to be held in different time slots and days to cover up four days of working hours of week.

Soft constraints for time slot optimization:

10.00 to 1.30 will get the maximum priority for each and every course.
8.30 to 10.00 and 3.00 to 4.30 will ge minimum priority for each and every course.

3.2.4 Platform

Our project will be launched as a Web-based application which will be accessed by a web browser which has an Internet connection.

3.2.5 Development and Responsibility

We would be developing the software and we are responsible for the creation of related interfaces, server connections and support.

3.2.6 Software Architecture

We will provide five services for all departments.Services are given below.
1. Manual routine edit
2. Faculty routine optimization
3. Room number optimization
4. Time slot optimization
5. Laboratory optimization

We will use four database.We will use excel sheet for database.Figure 3.2(a) describes the entire software architecture.

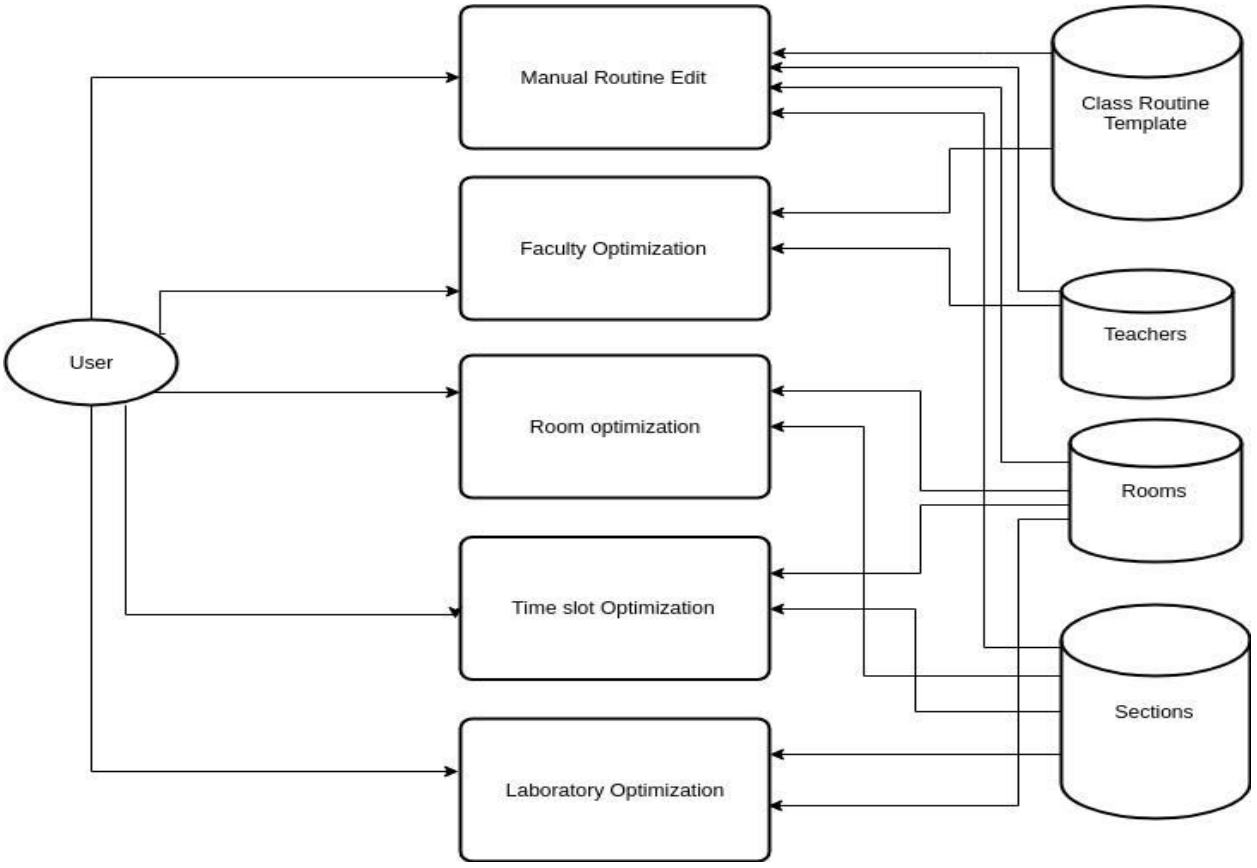


Figure 3.2(a) : Software Architecture.

3.2.7 ERP Solution diagram

Enterprise resource planning for routine optimization and management.For four automated systems for four optimization processes services.Figure 3.2(b) describes the entire process.

Figure 3.2(b) : ERP Solution diagram.

3.2.8 Requirement Analysis : Solution Designing Process

At first semester of our capstone project. We initially generate a concept of class routine optimization and management. Then gather resources for the project. Collect some papers and read about them. Know the basic needs of the users and analysis the entire system. Then at our second semester of capstone project. We make a plan,make a design and try to implement our design by making a website based routine management and optimization system. Figure 3.2(c) describes the entire process.

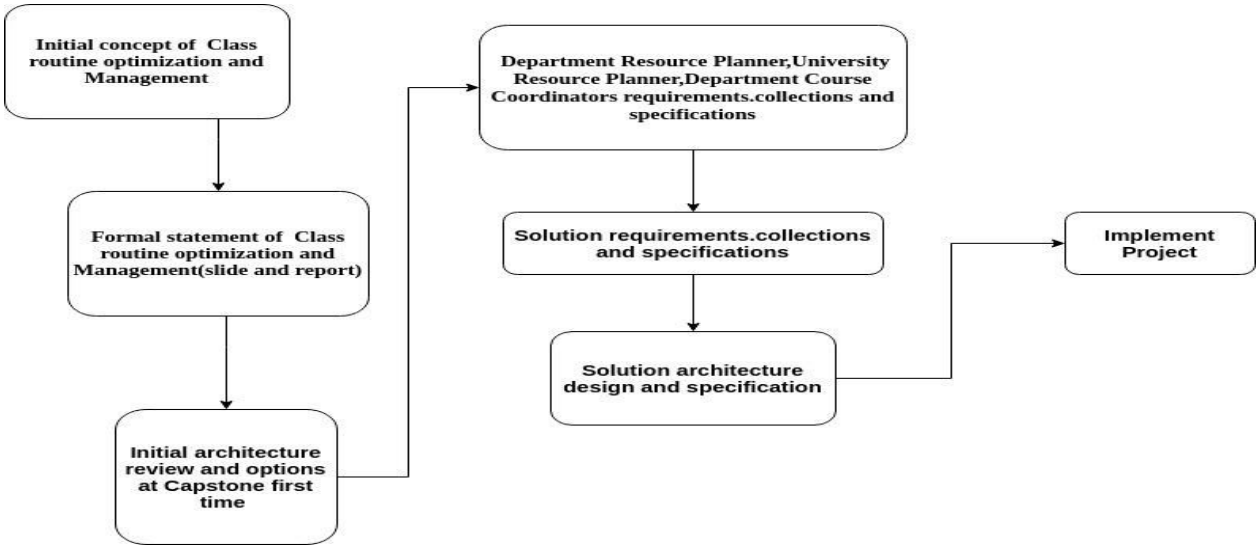


Figure 3.2(c) : Requirement Analysis : Solution Designing Process.

3.2.9 3 tier model

We applied 3 tier model approach at Figure 3.2(d) for routine optimization and management system. Interface services,routine optimization and management logic and database access service are our 3 tiers for the system. Four database for four excel sheets,web user interface and logic for optimization processes.

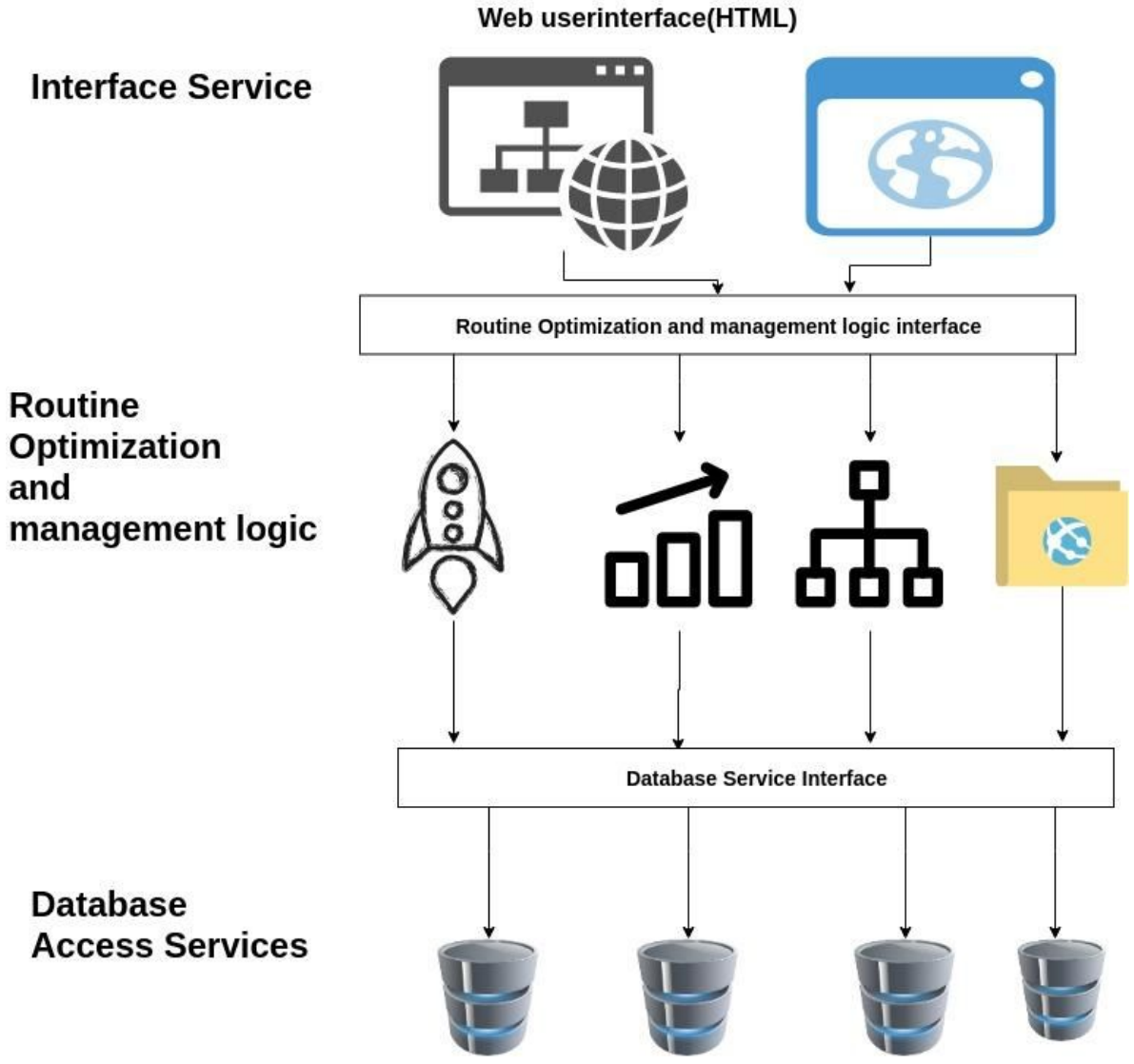


Figure 3.2(d) : 3 tier model.

3.2.10 Activity Diagram

At the activity diagram at Figure 3.2(e). We show how a user will use and experience the five services of the system. By creating an account and user can easily make routine, optimize routine, view routine, manually edit routine.

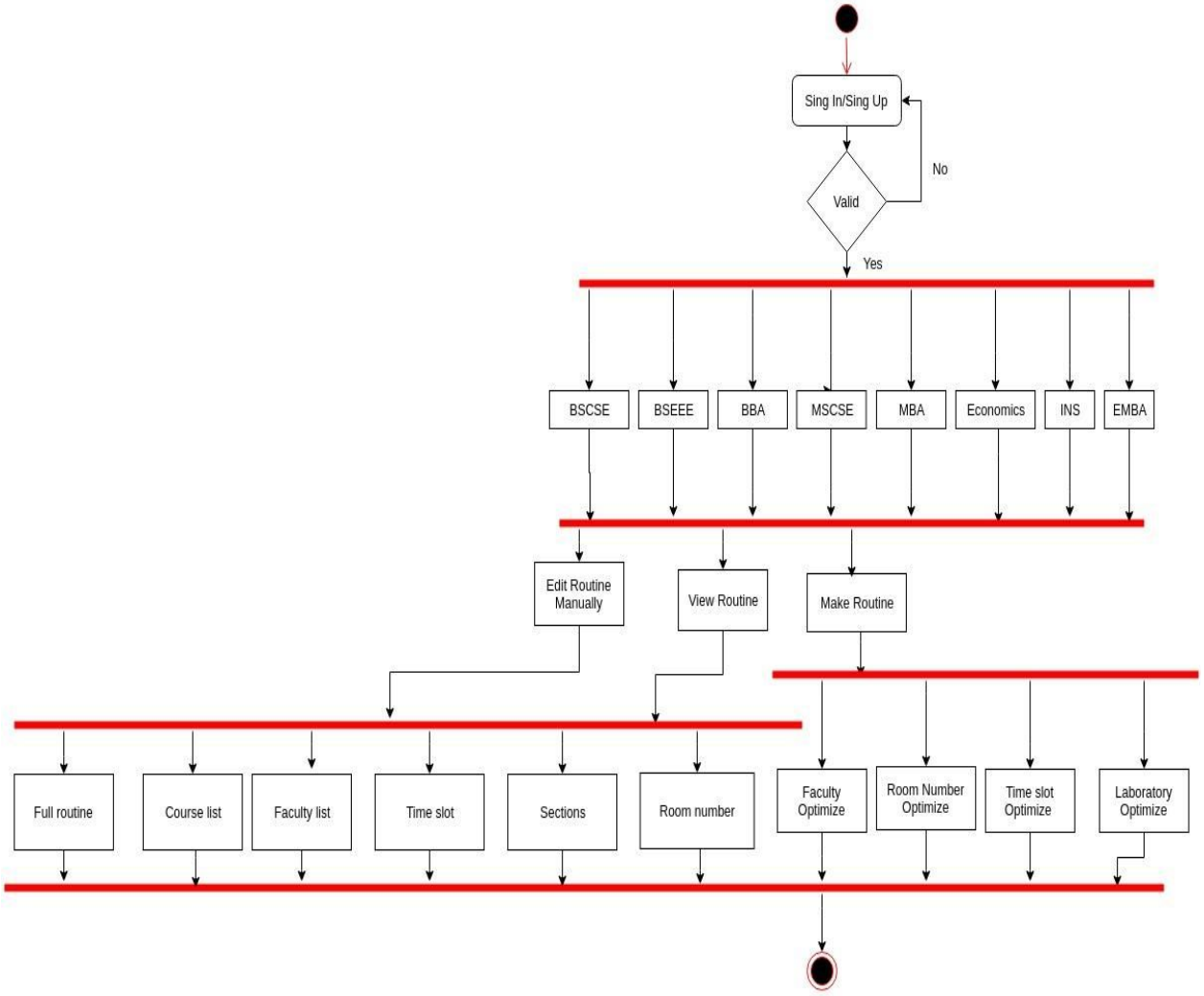


Figure 3.2(e) : Activity diagram.

3.2.11 Data Flow Diagram

At the data flow diagram at Figure 3.2(f) for 0 level and at Figure 3.2(g) for level 1. We show how the data passes and how the process works in the system.

Data flow diagram level 0:

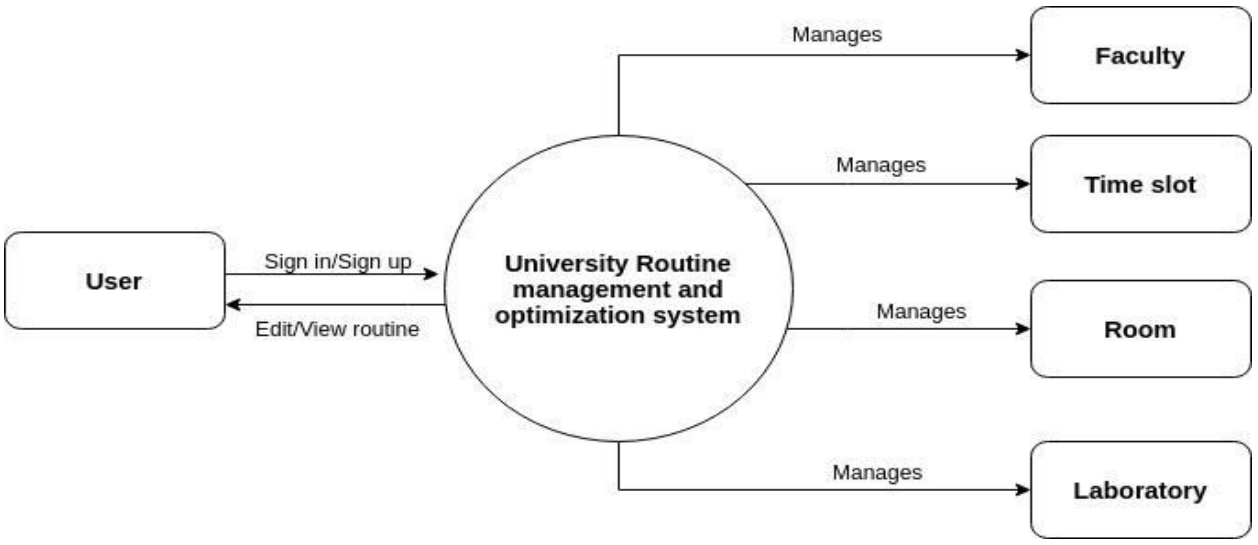


Figure 3.2(f) : Data flow diagram (level 0).

Data flow diagram level 1:

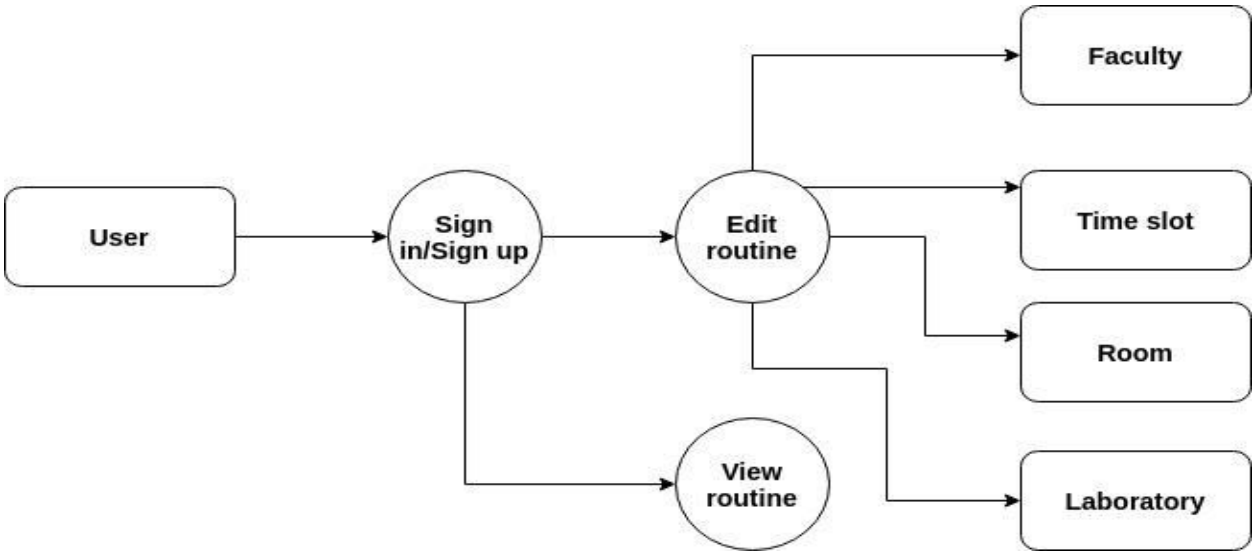


Figure 3.2(g) : Data flow diagram (level 1).

3.2.12 Class Diagram

Class diagram at Figure 3.2(h) shows us about the entire system.It's a blueprint.It shows each and every individual sub systems of 5 services of routine management and optimization.

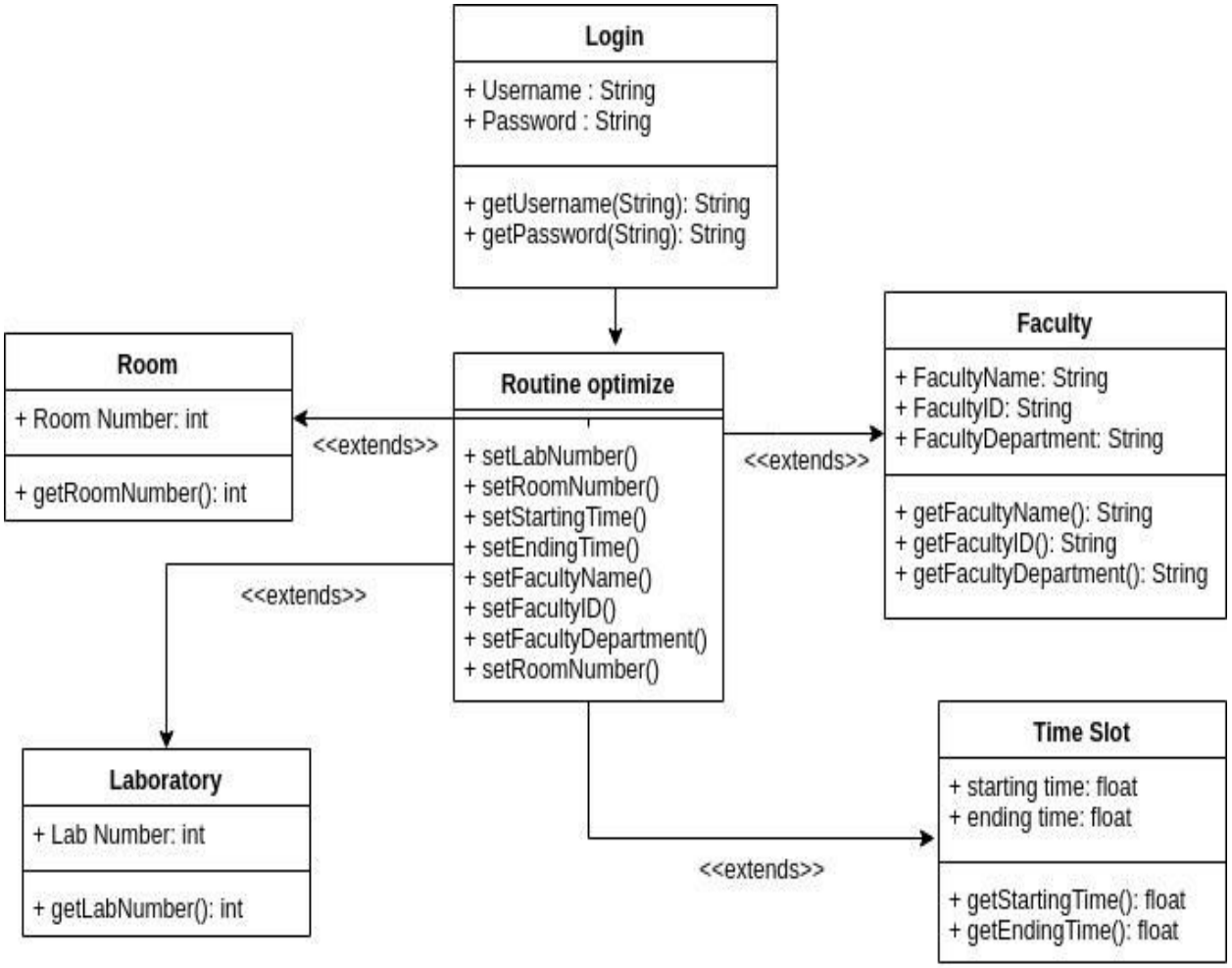


Figure 3.2(h) : Class Diagram.

3.2.13 Sequence Diagram

By sequence diagram at Figure 3.2(i).We are showing how an user will will perform his work.And how sequentially the system performs.

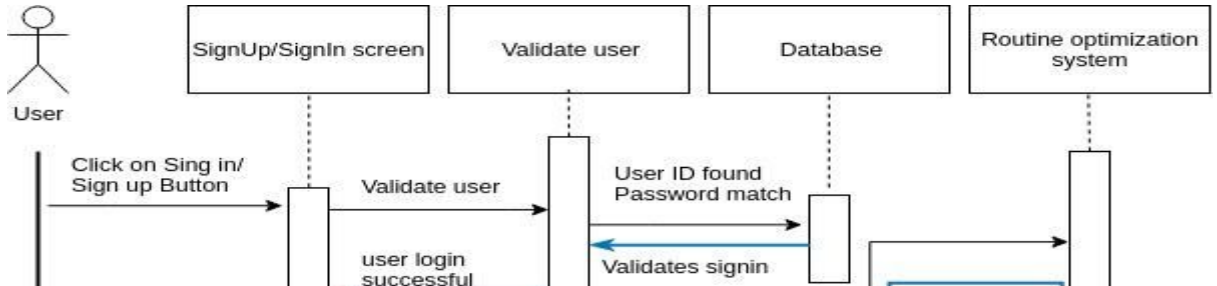


Figure 3.2(i) : Sequence Diagram.

3.2.14 Deployment Diagram

At deployment diagram at Figure 3.2(j) we can see that,our 3 tier processes. How the subsystems of four optimization process actually connected.

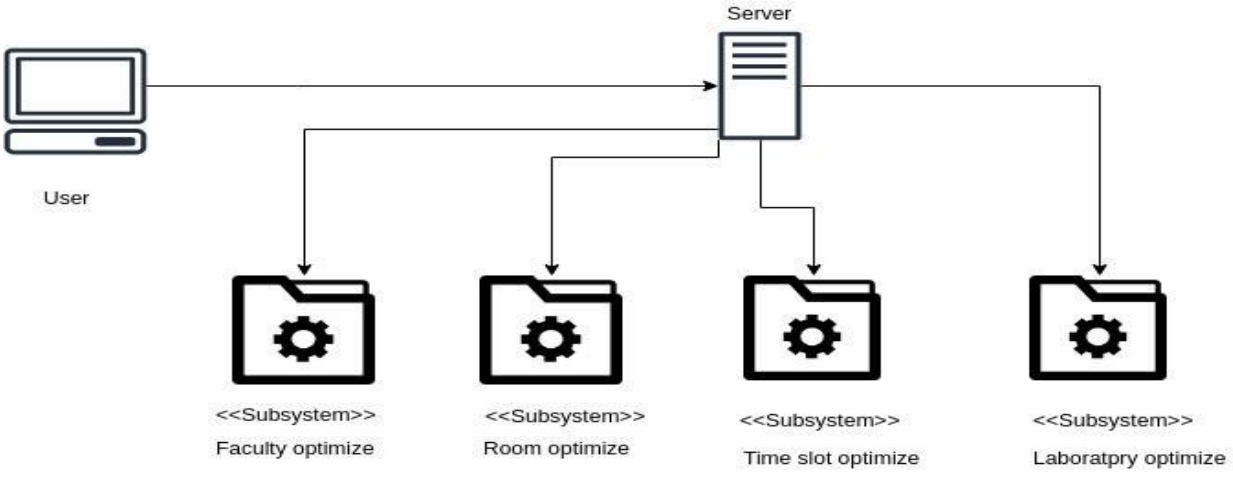


Figure 3.2(j) : Deployment diagram.

3.2.15 Use case Diagram

The use case diagram shows the relationship between users and the major processes of our system. The three users will interact with in total seven major processes shown in Figure 3.2(k).

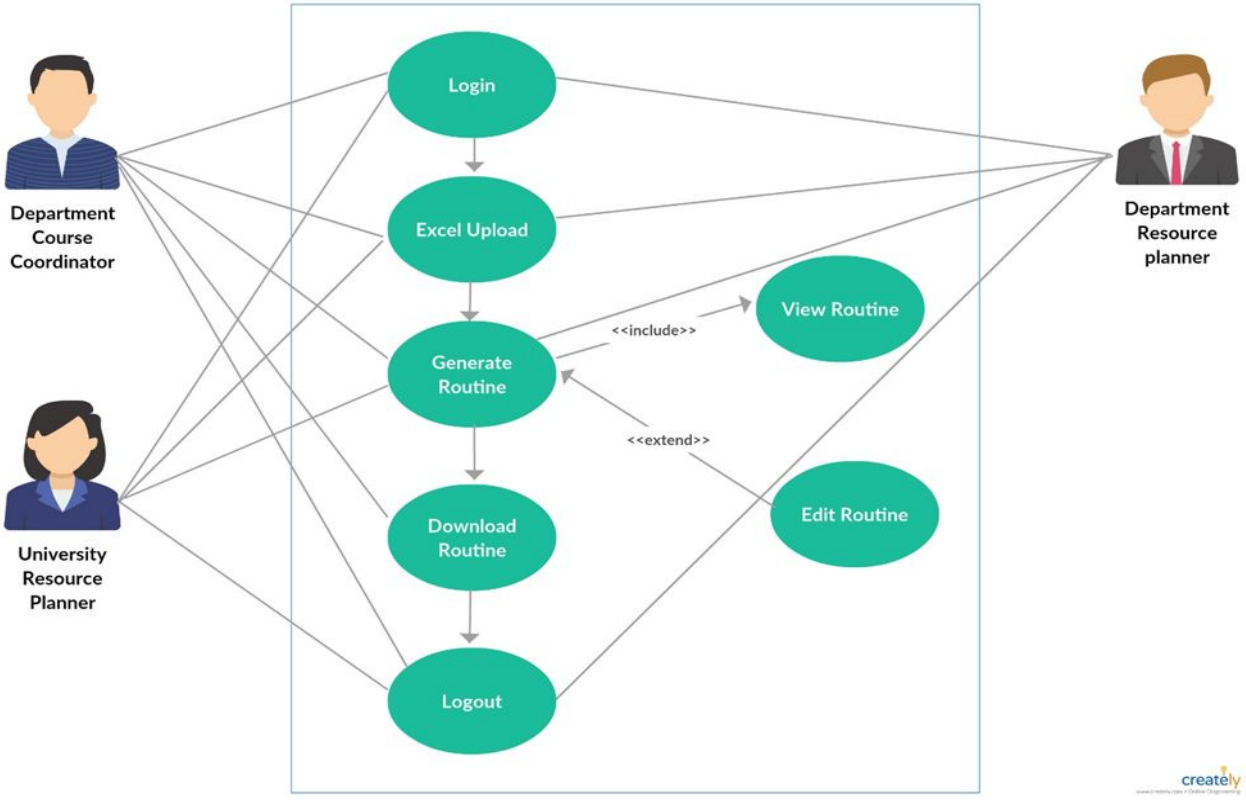


Figure 3.2(k): Use Case diagram of Routine Assignment Problem.

3.2.16 Functional Requirements

The functional requirement is describing the behavior of the system as it relates to the system's functionality. A finely designed system's usability is always satisfactory. This routine management systems portal and each of its pages are very much easy to use. User will find it comfortable and some good set of directives are given to guide the user doing different set of activities. The category of user interfaces depends on the privileges given to the users. All the basic user functions that the user can perform are shown at the homepage and they are just one click away to access those functions. Making it user friendly is one of our prime goals and there will be a feedback screen for the user if there are any issues to address.

3.2.17 Use Case: Excel Upload

Figure 3.2(l): Use case for Department Resource Planner.

Brief Description

The Department Resource Planner Figure 3.2(l) creates an Excel file and inputs the time slots and courses. After that he uploads the file into the system.

Initial Step-By-Step Description

Department Resource Planner uses the web tool to upload the excel file he created. This file goes through the system and is accessed by others.

1. Department Resource Planner logs into the system with his id and password.
2. He creates excel file with time slots and course schedule.
3. He uploads the excel file into to system.
4. If the file is uploaded successfully a confirmation message pops up.
5. He logs out of the system.

Figure 3.2(m): Use case for University Resource Planner.

Brief Description

The University Resource Planner Uploads Figure 3.2(m) the room numbers excel file into the system.

Initial Step-By-Step Description

The University Resource Planner uses this system to input the room numbers for each department. This is another important part of the registration process.

1. University Resource Planner logs into the system.
2. She uploads the excel file containing room numbers for each department.
3. She logs out of the system.

Figure 3.2(n): Use case for Department Course Coordinator.

Brief Description

Department Course Coordinator Figure 3.2(n) is the person who assigns faculty members to each section of each course. The courses and time schedules are already in the excel file at this point. His job is to provide the current faculty names and the system will generate the routine using predefined constraints.

Initial Step-By-Step Description

The Department Course Coordinator does the final step of this process. He inputs all the faculty names which goes as an input of the final optimized routine.

1. Department Course Coordinator logs in into the system using id and password.
2. He uploads the excel file containing the available faculty name for upcoming trimester.
3. Upon successful upload a pop up will appear.

3.2.18 Use Case: Generate Routine

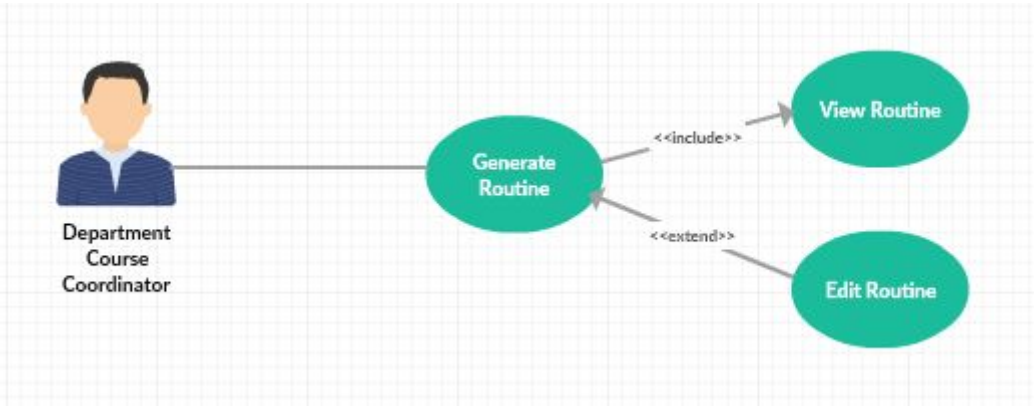


Figure 3.2(o): Use case for process Generate Routine.

Brief Description

The generate process includes viewing of the generated routine Figure 3.2(o). User can also edit the routine in extended process. He can make changes permitted by the constraints which doesn't create conflict. It allows to customize the final routine to Department Course Coordinator preferences.

Initial Step-By-Step Description

In this process the final optimized routine is generated and the output is given as an excel file. The users can download the file from the web tool.

1. Department Course Coordinator will click on Generate.
2. An optimized routine will be created following the constraints.
3. He will then have an option to view the routine.
4. He can also edit the routine in Edit option.

3.2.19 Requirements Specification

Let us discuss the non-functional requirements of our routine management system.

System Properties

The system properties are listed below which describes which tools were used to make this project and what are the requirements to run this tool.

a) A web based routine management portal runs on the internet.
b) Runs on Python server.
c) HTML/CSS, Bootstrap, Javascript based UI design.

Storage Requirements

The storage requirements for the project are described down below.

a) Different contents uploaded to the system are preserved.
b) Minimum and maximum requirements of disk space are considered with care.
c) Unnecessary and obsolete contents are removed to free the disk space.

Accessibility

Here we have stated the means by which someone can access this web tool.

a) This system is accessible from mobile/desktop/laptop devices.
b) A stable Internet connection is required.

Capacity, current and forecast

The capacity and limit of usage of the web tool is stated below.

a) The time of minimum and maximum use of the system are taken into account.
b) During the registration and before the examination period uses of the system are very high.
c) At Max load period additional CPU and Memory arrangements will be made to cope with the load.

Documentation

Proper documentation makes it easier to use any system. We will provide visual directions and instructions to use this web tool. Also,

a) User guidelines to use the system with screenshots will be provided.
b) User privileges and access management guidelines will be added in the documentation.

Availability

The system will be available to its specific users as stated below.

a) A Valid user account can access this system 24/7.
b) A minimum amount of downtime will be taken during maintenance period.

Scalability

It is our wish to build a tool that is flexible and fluid to use and to update, also can be implemented with other tools and websites. So, we can ensure that,

a) This system will be designed and developed modular basis.
b) A new module with additional functionality can be added with ease.

Privacy

Privacy is a major concern for web-based tools that are used in big organizations and institutions. We will ensure that the privacy of everyone involved is maintained. We will also ensure that,

a) User's privacy will be maintained with strict measures.
b) No user can use the system beyond the privileges given to him/her.

Security

We are hoping to develop the system with highest security measures so there are no data leaks and hackings. So, we have ensured,

a) Our routine manages web portal will use Hypertext Transfer Protocol Secure (HTTPS) to disallow security attacks.
b) No plain text content will be passed in or out.

3.2.20 Technical Process

Following would be the languages we would like to use for the development of our application within the stipulated time period:

Front-end development: HTML, CSS, Bootstrap, Javascript.

Back-end development:
• Programming Language :Python, php 7.
• IDE :Anaconda, Spyder , Sublimetext.
• First initiative: Applying Random search algorithm in local search approach to find the best optimized solution for Faculty List.

3.3 Methodology Flow

In Figure 8 we have shown the method flow. First, we did requirement analysis and then literature review was done. After understanding the problem with the help of literature review we searched for algorithms to use for our specific problem Then we will try to develop the algorithm and do the coding. Once the system is built we will test it thoroughly and verify it. Finally we will implement the tool for the users to use and get an optimized routine effortlessly.

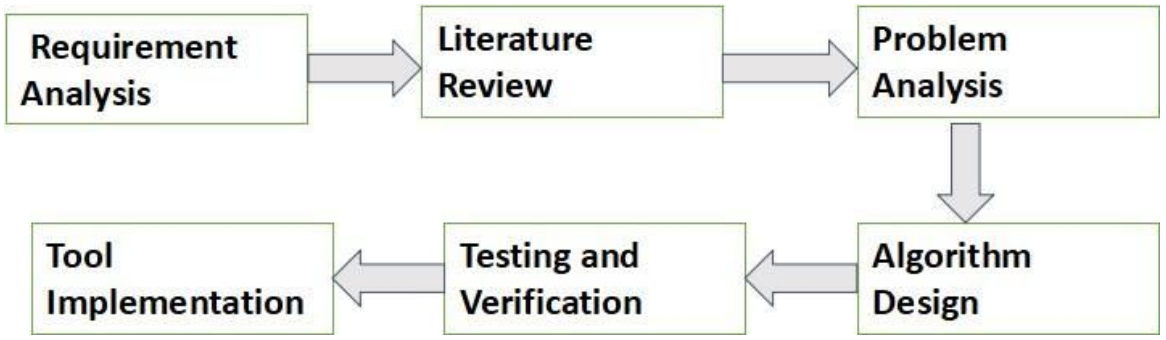


Figure 3.3(a): Method Flow.

Description:

Here we have described the process we followed to develop this project. First, we researched about the requirements to build this system. To do that we reviewed a lot of papers and literature journals. After that we started to modify the algorithm for our problem and designed the algorithm. After testing and verification we will implement the web tool to generate optimized routine.

3.3.1 Cost Function

We have satisfied some hard constraints and soft constraints to make cost function.Cost function will return a cost according to constraint satisfaction.

3.3.2 Diagram of constraint satisfaction

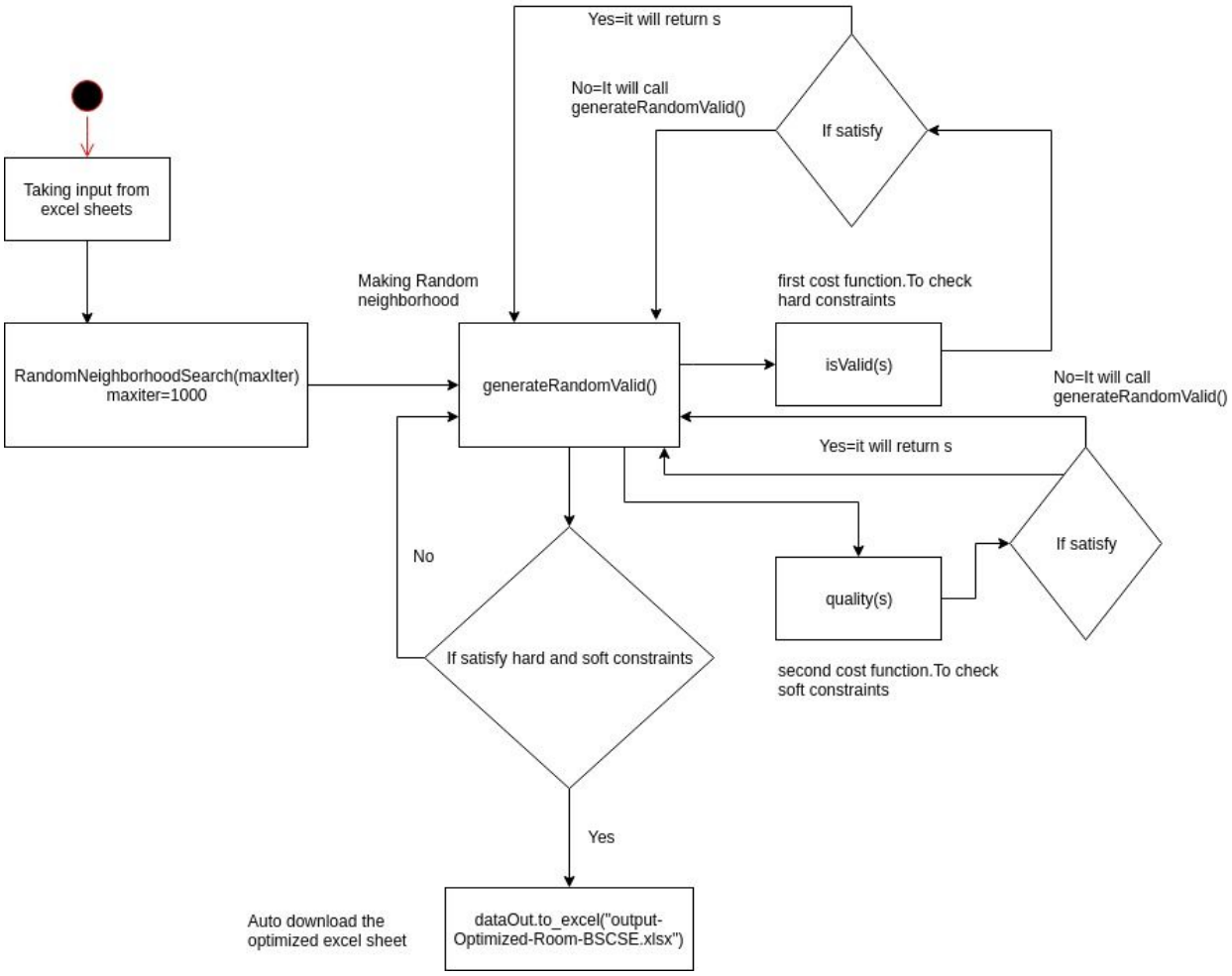


Figure 3.3(b): Diagram of constraint satisfaction

Requirement Analysis – System Analysis

At first, we will have to study the system. How it works, what type of problems are faced here etc. So that is why we will have to collect the information and data regarding this system.

Literature Review – Google scholar

Then we studied our literatures from Google Scholar where various types of scheduling system were optimized. We found many solutions and then we found out the lacking in the literature. So, we decided exactly where we are going to work and how our algorithm and method will proceed.

Problem Analysis – Objectives

After doing a literature review, we found our working platform. So, we found where we need to improve and redesign the algorithm. So, the objective is to find the most optimized solution. And the objective is to automate the scheduling and to save human resource.

Algorithm Design

We have been designing an algorithm which will give us the best solution for our problem. And this solution will be better than the previous ones.

Simulated Annealing.

Testing and Verification – Check with existing data

Then we are going to make our project and make a prototype at the starting. We will check the prototype with the existing data and then make amendments. Thus our final implementation will proceed.

Tool Implementation – Website using php7

Our back end will be made by using php7. We will use HTM5L, CSS3, Bootstrap 3, java script.

Chapter 4

Application

We expect to find an optimized routine in an excel sheet. After uploading the excel sheets of constraints the generate button will be clicked and then our algorithm will generate a new excel sheet with optimized routine. Then resource planner can download it.

1. We have used bootstrap 3 and javascript for user interface.
2. We have used php7 for back end development of our website.
3. We have used python 3.7.2 for our web optimization engine.
4. We have used python library matplotlib for data visualization.
5. We have used python library pandas for data processing.

4.1 Home page

At Figure 5 we can see the homepage of the website. Where user can see the options of load file, view file and manually edit file. User can choose any options from navigation bar.

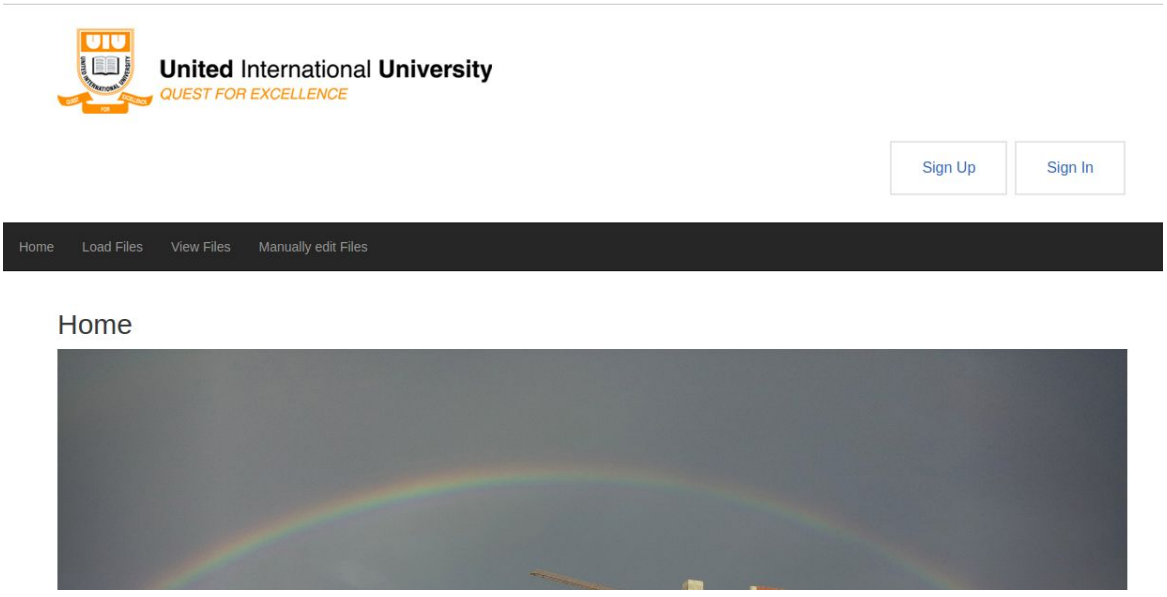


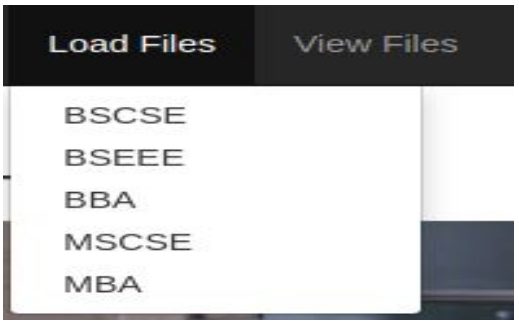
Figure 4.1: Home page.

4.2 Load, view and edit files

From the navigation bar user can choose option to load, view or edit any file.

4.2.1 Load file

User can choose excel sheet of routine to optimize for BSCSE, BSEEE, MSCSE, MSEEE, BBA,MBA,ECONOMICS and INS department.By clicking the “generate” button.user will get an optimized version of routine of the uploaded excel sheet.



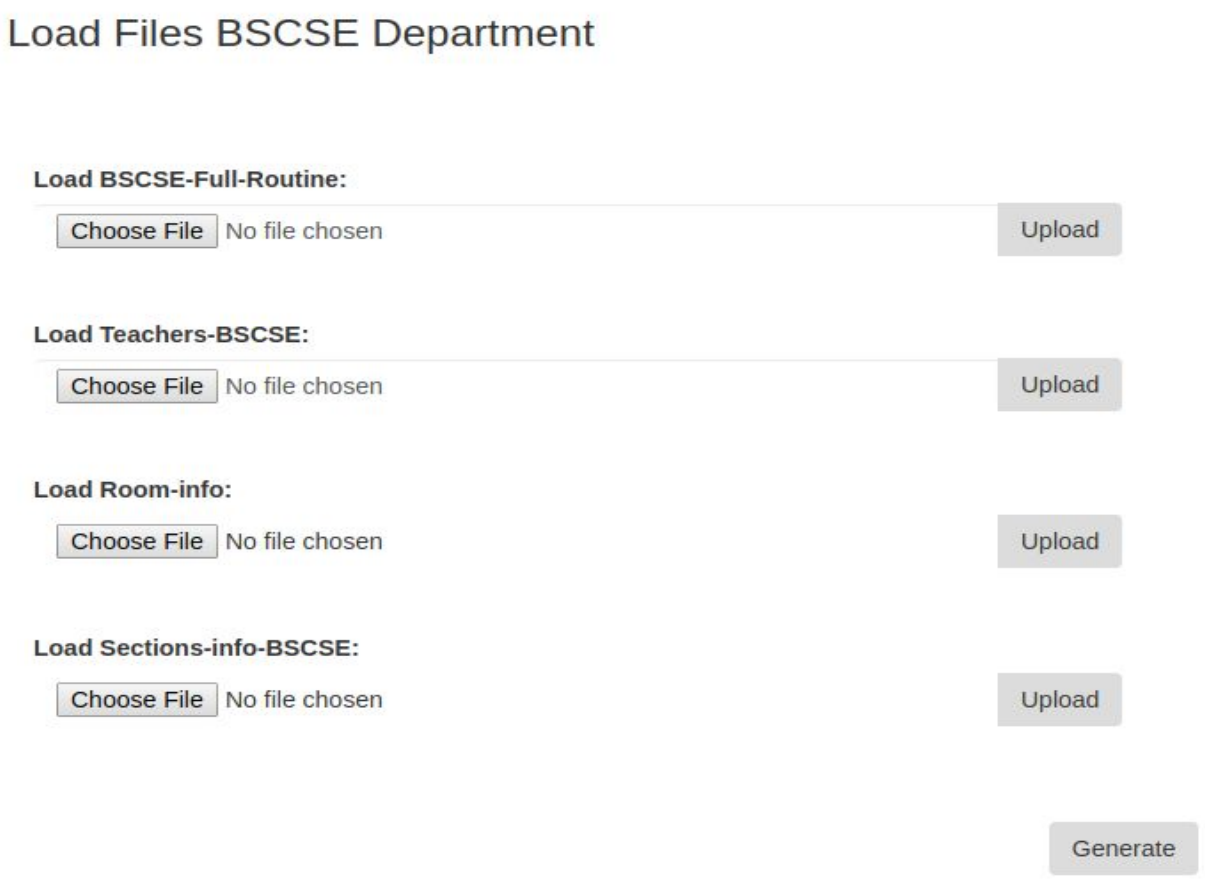
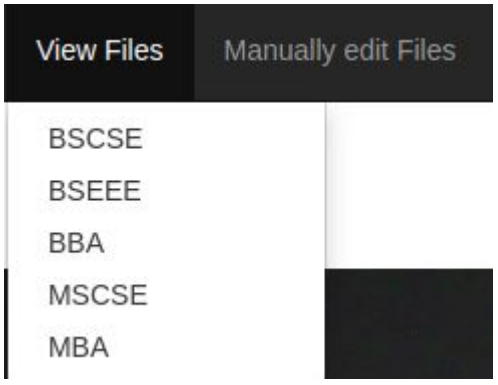


Figure 4.2(a): Load File Page.

4.2.2 View file

User can view excel sheet of routine for BSCSE, BSEEE, MSCSE,BBA, MBA, department. By clicking the view full routine,view faculty list,view room numbers,view course list,view time slot,view laboratory buttons user will see the respective routines. User can download that routine by clicking “download” button.



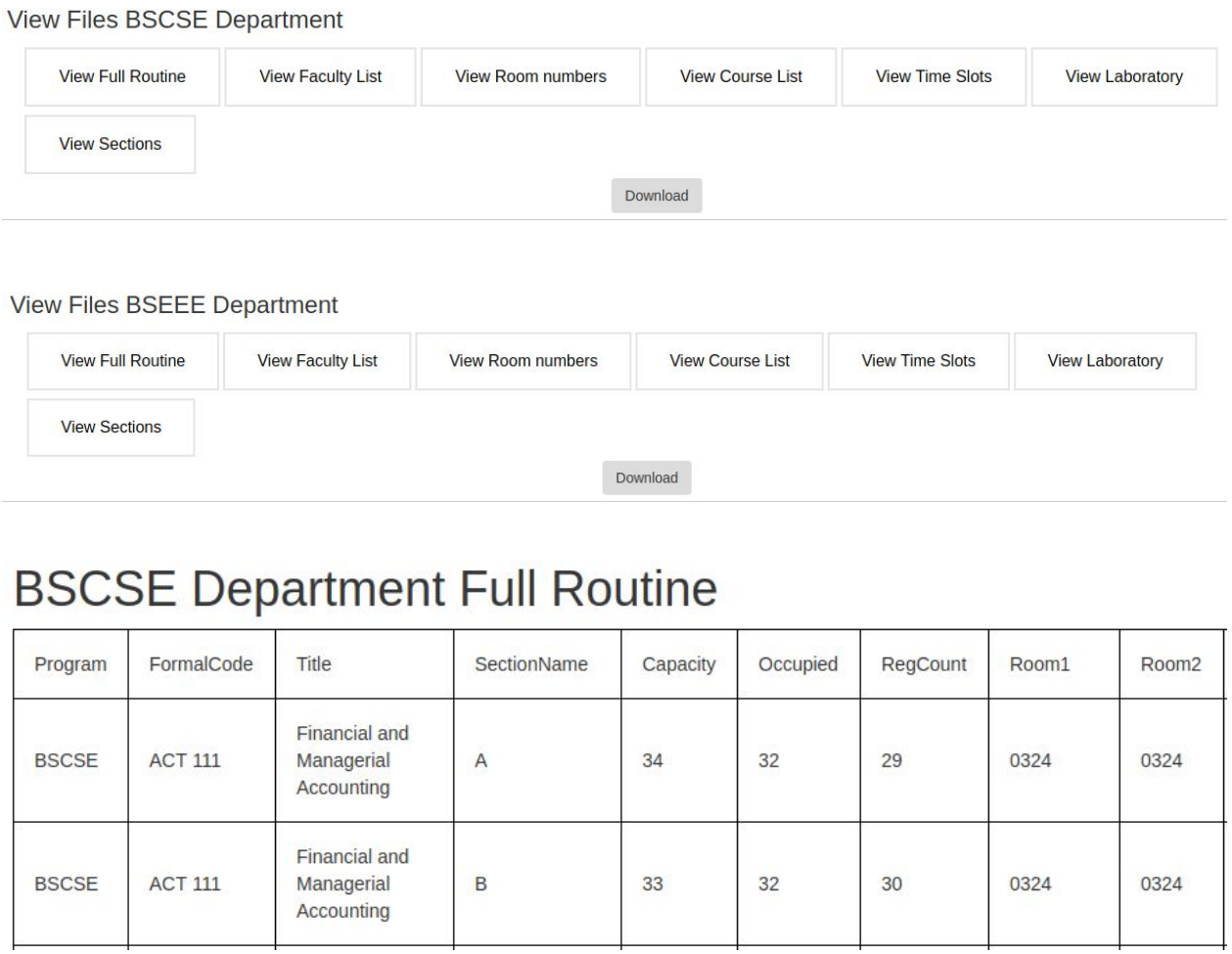
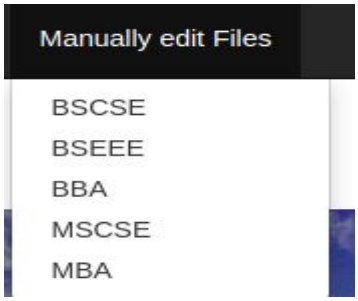


Figure 4.2(b) : View File Page.

4.2.3 Edit file

User can manually edit excel sheet of routine for BSCSE, BSEEE, MSCSE, BBA,MBA, department. By clicking the edit full routine,edit faculty list, edit room numbers,edit course list,edit time slot,edit laboratory buttons user will edit the respective routines. User can download that manually edited routine by clicking “Edit routine Download” button.



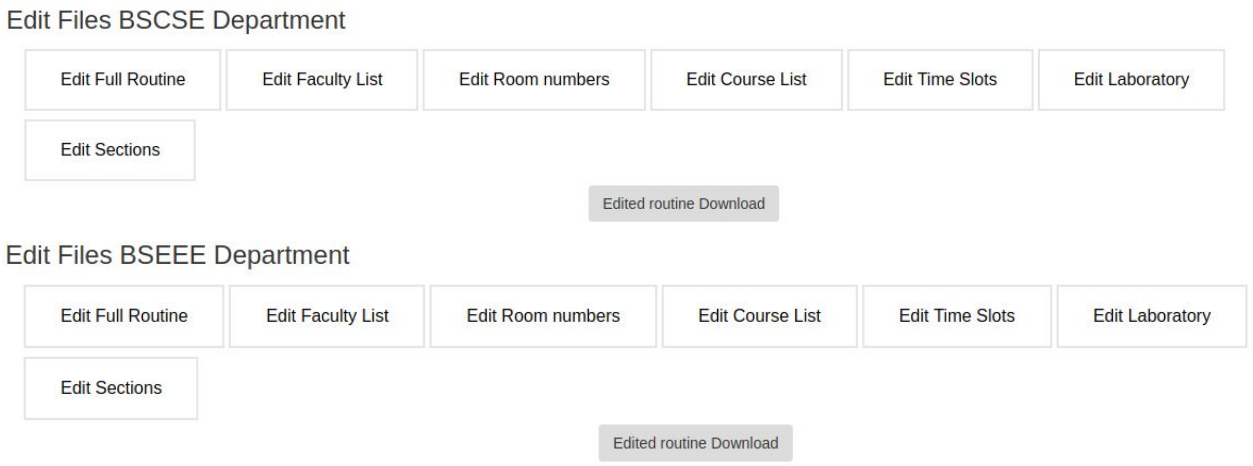


Figure 4.2(c): Edit File Page.

4.3 Sign in and Sign up

User needs to open an account to use the services.

4.3.1 Sign up

At first user will open an account by Sign up. Then use the services.

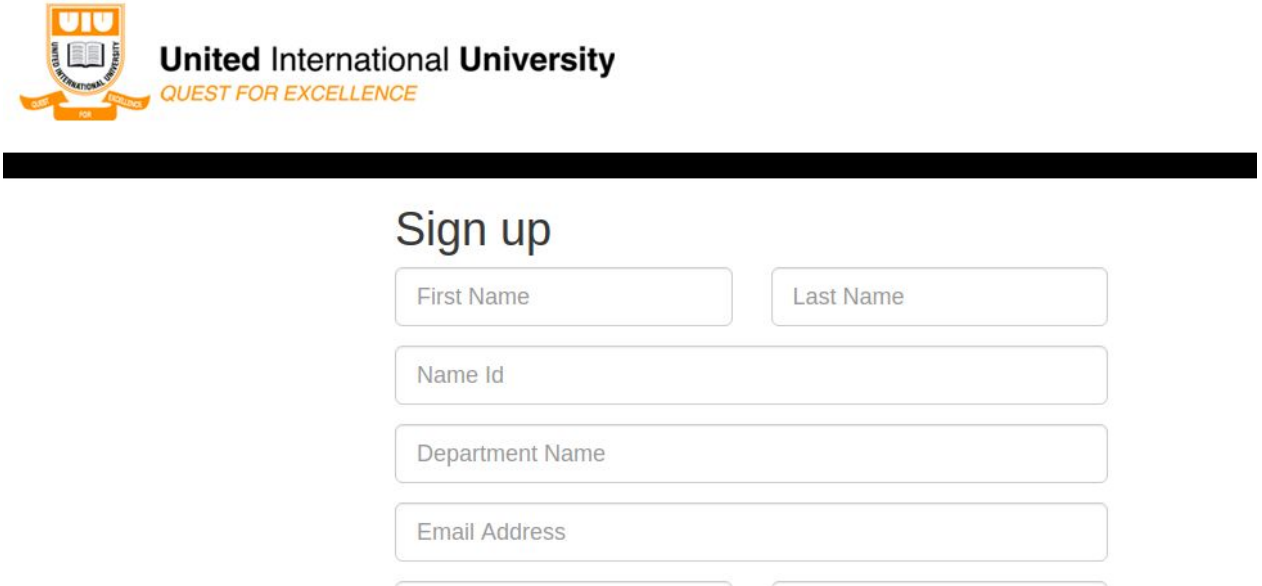


Figure 4.3(a): Sign Up Page.

4.3.2 Sign in

After creating an account. User only needs to sign in for further use.

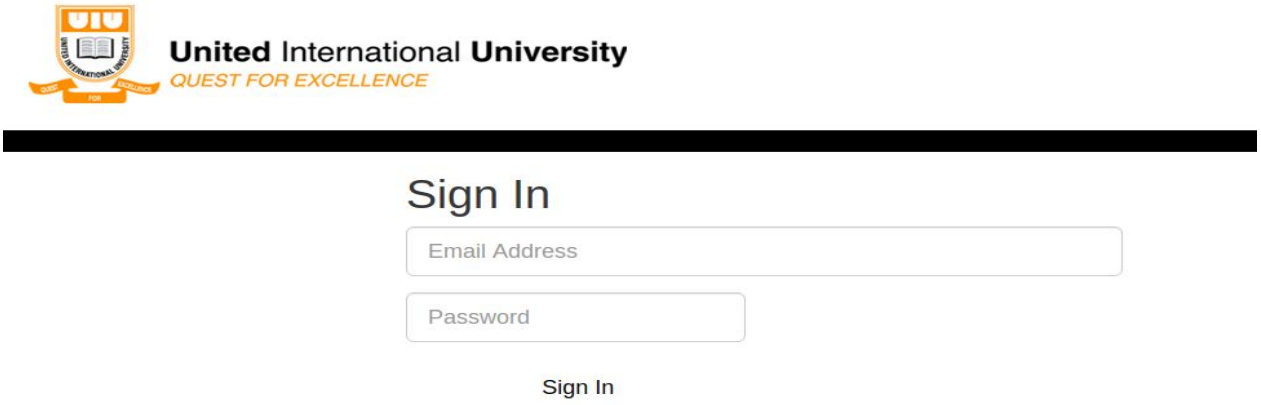


Figure 4.3(b): Sign In Page.

Chapter 5

Conclusion

5.1 Summary

We tried to make an automated system. So that the university resource planner, department resource planner and department course coordinator can make routine automatically and the routine will be optimized.

We used simulated annealing for local search for optimization. We used cost function to satisfy hard and soft constraints.

In the process of doing so we learn about different algorithms and tool implemented by different universities and researchers to solve this problem. We hope to find the best result and automate another side of our university administrative system.

5.2 Limitations

Everything has limitations of some sort or another. The front end design of our website is not very modern looking we want to fix this. The data display options are not very dynamic. We used PHP which is slower then python. These are the limitations we want to fix.

5.3 Future Work

In future we will try to include more option for users about making routine. That routine will have connection with students and academic calendar. We will try to apply Django for back end development. We will try to replace php7 with Django. We will try to make new neighborhood. And analysis the result. If it is good. Then we will replace our cost function and present neighborhood with it. We will try to upgrade our code implementation. Now it is in Beta version. Then it will be full version. We will try to reduce our computation time and boost computation speed.We will try to give data visualization service and routine model prediction option.

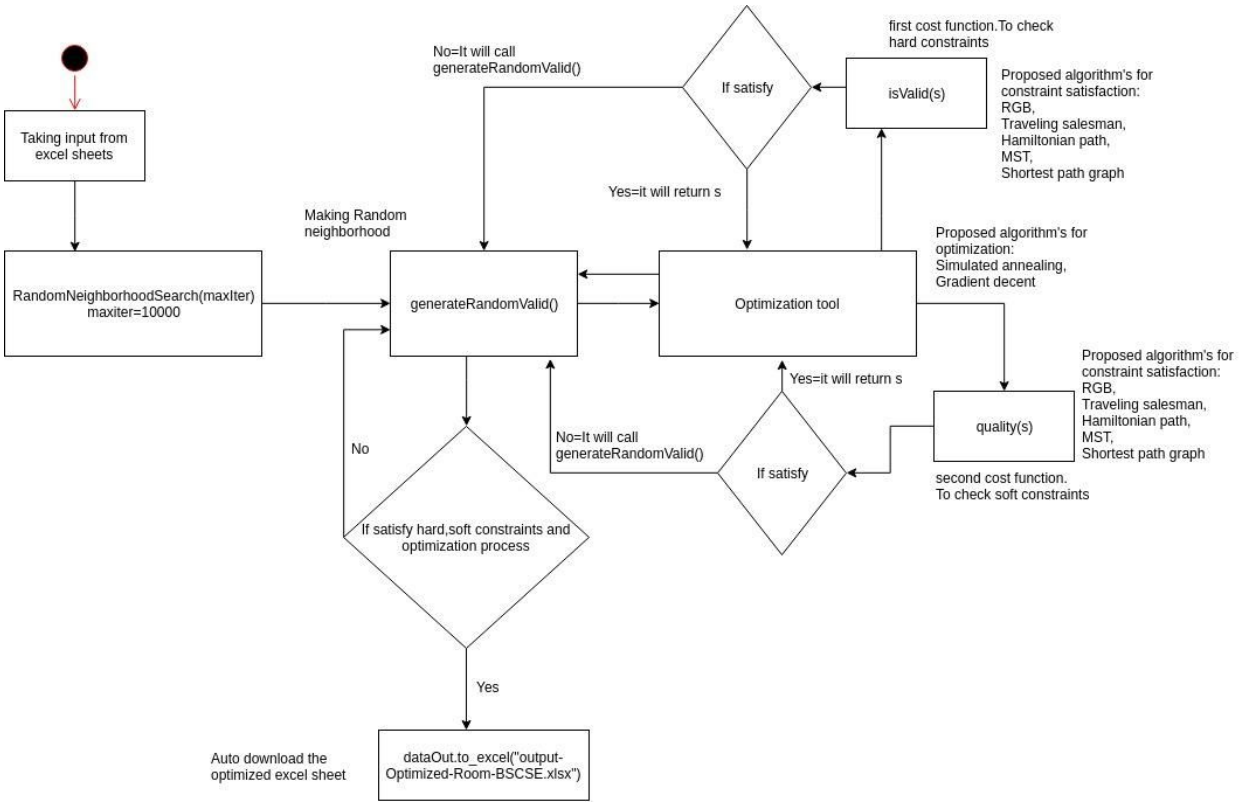


Figure 5(a) : Optimized Constraint Satisfaction Diagram.

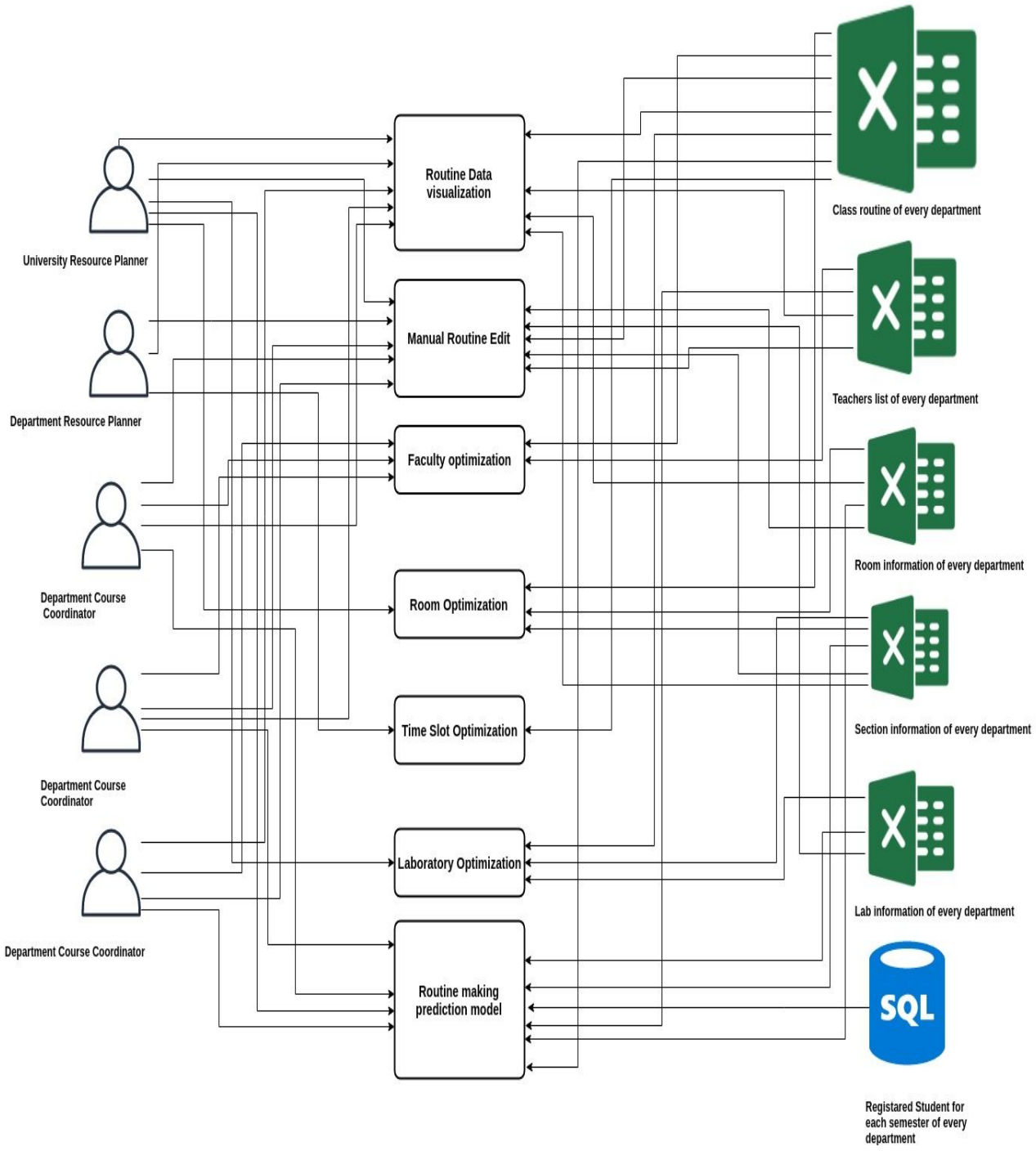


Figure 5(b) : Optimized Constraint Satisfaction Diagram.

References

[1] Abdullah, S., Burke, E.K. and McCollum, B., 2007. Using a randomised iterative improvement algorithm with composite neighbourhood structures for the university course timetabling problem. In Metaheuristics (pp. 153-169). Springer, Boston, MA..

[2] Socha, K., Knowles, J. and Samples, M., 2002, September. A max-min ant system for the university course timetabling problem. In International Workshop on Ant Algorithms (pp. 1-13). Springer, Berlin, Heidelberg.

[3] Abdullah, S., Burke, E.K. and McCollum, B., 2007, September. A hybrid evolutionary approach to the university course timetabling problem. In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on (pp. 1764-1768). IEEE.

[4] Ugat, E.B., Montemayor, J.J.M., Manlimos, M.A.N. and Dinawanao, D.D., 2012. Parallel Ant Colony Optimization on the University Course-Faculty Timetabling Problem in MSU-IIT. GSTF Journal on Computing, 1(4).

[5] Chang, L.S. and Kahar, M.N.M., 2018. MODELLING THE UNIVERSITI MALAYSIA PAHANG EXAMINATION TIMETABLING PROBLEM. International Journal of Software Engineering and Computer Systems, 4(1), pp.106-119.

[6] Goh, S.L., Kendall, G. and Sabar, N.R., 2018. Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem. Journal of the Operational Research Society, pp.1-16.

[7] Zervoudakis, K. and Stamatopoulos, P., 2000, August. A generic object-oriented constraint-based model for university course timetabling. In International Conference on the Practice and Theory of Automated Timetabling (pp. 28-47). Springer, Berlin, Heidelberg.

[8] Rossi-Doria, O. and Paechter, B., 2004. A memetic algorithm for university course timetabling. Combinatorial optimisation, p.56.

[9] Adaptive Selection of Heuristics for Improving Exam Timetables.

[10] Socha, K., Samples, M. and Manfrin, M., 2003, April. Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In Workshops on Applications of Evolutionary Computation (pp. 334-345). Springer, Berlin, Heidelberg.

[11] Kostuch, P. and Socha, K., 2004, April. Hardness prediction for the university course timetabling problem. In European Conference on Evolutionary Computation in Combinatorial Optimization (pp. 135-144). Springer, Berlin, Heidelberg.

[12] Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E. and Qu, R., 2013. Hyper-heuristics: A survey of the state of the art. Journal of the Operational Research Society, 64(12), pp.1695-1724.

[13] Ahandani, M.A. and Baghmisheh, M.T.V., 2011, October. Memetic algorithms for solving university course timetabling problem. In 2011 1st International eConference on Computer and Knowledge Engineering (ICCKE) (pp. 40-44). IEEE.

[14] Rudová, H. and Vlk, M., Multi-criteria soft constraints in timetabling, 2006. Submitted to Annals of Operations Research. Special volume on Multi-disciplinary Scheduling: Theory and Applications.

[15] Abdennadher, S. and Marte, M., 2000. University course timetabling using constraint handling rules. Applied Artificial Intelligence, 14(4), pp.311-325.

[16] Autry, B.M., 2008. University course timetabling with probability collectives (Doctoral dissertation, Monterey, California. Naval Postgraduate School).

[17] Rudová, H. and Murray, K., 2002, August. University course timetabling with soft constraints. In International Conference on the Practice and Theory of Automated Timetabling (pp. 310-328). Springer, Berlin, Heidelberg.

[18] Basir, N., Ismail, W. and Norwawi, N.M., 2013. A simulated annealing for Tahmidi course timetabling. Procedia Technology, 11, pp.437-445.

[19] Lewis, R., Paechter, B. and Rossi-Doria, O., 2007. Metaheuristics for university course timetabling. In Evolutionary scheduling (pp. 237-272). Springer, Berlin, Heidelberg.

Appendix A

Data visualization code by python.By using python matplotlib.

```
import matplotlib.pyplot as plt
import pandas as pd
plt.style.use('ggplot')
teachers=pd.read_excel('Teachers-BSCSE.xlsx')
Faculty_name = teachers["Faculty Member"]
TheoryBSCSE = teachers["Theory of BSCSE"]
LabBSCSE = teachers["Lab of BSCSE"]
TheoryBSEEE = teachers["Theory of BSEEE"]
LabBSEEE = teachers["Lab of BSEEE"]
TheoryMSCSE = teachers["Theory of MSCSE"]
LabMSCSE = teachers["Lab of MSCSE"]
ind = [x for x, _ in enumerate(Faculty_name)]
plt.figure(figsize=(15,10))
plt.bar(ind, TheoryBSCSE, width=0.8, label='Theory BSCSE', color='gold',
bottom=LabMSCSE+TheoryMSCSE+LabBSEEE+TheoryBSEEE+LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE+TheoryMSCSE+LabBSEEE+TheoryBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE+LabBSEEE+TheoryBSEEE+LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=TheoryMSCSE+LabBSEEE+TheoryBSEEE+LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE+TheoryMSCSE+LabBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE+LabBSEEE+TheoryBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE+TheoryBSEEE+LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=TheoryMSCSE+LabBSEEE+TheoryBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=TheoryMSCSE+LabBSEEE+LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabBSEEE+TheoryBSEEE+LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE+TheoryMSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE+LabBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE+TheoryBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE+LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=TheoryMSCSE+LabBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=TheoryMSCSE+TheoryBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=TheoryMSCSE+LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabBSEEE+TheoryBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabBSEEE+LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabBSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=TheoryBSEEE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=TheoryMSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold', bottom=LabMSCSE)
plt.bar(ind, TheoryBSCSE, width=0.8, color='gold')

plt.bar(ind, LabBSCSE, width=0.8, label='Lab BSCSE', color='silver', bottom=LabMSCSE+TheoryMSCSE+LabBSEEE+TheoryBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=LabMSCSE+TheoryMSCSE+LabBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=LabMSCSE+TheoryMSCSE+TheoryBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=LabMSCSE+LabBSEEE+TheoryBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=TheoryMSCSE+LabBSEEE+TheoryBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=LabMSCSE+TheoryMSCSE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=LabMSCSE+LabBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=LabMSCSE+TheoryBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=TheoryMSCSE+LabBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=TheoryMSCSE+TheoryBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=LabBSEEE+TheoryBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=TheoryBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=LabBSEEE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=TheoryMSCSE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver', bottom=LabMSCSE)
plt.bar(ind, LabBSCSE, width=0.8, color='silver')

plt.bar(ind, TheoryBSEEE, width=0.8, label='Theory BSEEE', color='#CD853F',bottom=LabMSCSE+TheoryMSCSE+LabBSEEE)
plt.bar(ind, TheoryBSEEE, width=0.8, color='#CD853F',bottom=LabMSCSE+TheoryMSCSE)
plt.bar(ind, TheoryBSEEE, width=0.8, color='#CD853F',bottom=LabMSCSE+LabBSEEE)
plt.bar(ind, TheoryBSEEE, width=0.8, color='#CD853F',bottom=TheoryMSCSE+LabBSEEE)
plt.bar(ind, TheoryBSEEE, width=0.8, color='#CD853F',bottom=LabBSEEE)
```

```
plt.bar(ind, TheoryBSEEE, width=0.8, color='#CD853F',bottom=TheoryMSCSE)
plt.bar(ind, TheoryBSEEE, width=0.8, color='#CD853F',bottom=LabMSCSE)
plt.bar(ind, TheoryBSEEE, width=0.8, color='#CD853F')

plt.bar(ind, LabBSEEE, width=0.8, label='Lab BSEEE', color='orange', bottom=LabMSCSE+TheoryMSCSE)
plt.bar(ind, LabBSEEE, width=0.8, color='orange', bottom=TheoryMSCSE)
plt.bar(ind, LabBSEEE, width=0.8, color='orange', bottom=LabMSCSE)
plt.bar(ind, LabBSEEE, width=0.8, color='orange')

plt.bar(ind, TheoryMSCSE, width=0.8,label='Theory MSCSE',  color='red',bottom=LabMSCSE)
plt.bar(ind, TheoryMSCSE, width=0.8, color='red')

plt.bar(ind, LabMSCSE, width=0.8, label='Lab MSCSE', color='yellow')

plt.hlines(6,0,teachers.shape[0],color='r')
plt.hlines(3,0,teachers.shape[0],color='g')
plt.xticks(ind, Faculty_name,rotation=90)
plt.ylabel("Total Sections of Courses")
plt.xlabel("Faculty Members Name")
plt.legend(loc="upper right")
plt.title("BSCSE Faculty section of courses information")
plt.show()
```

Appendix B

Constraint satisfaction code by python.By using python random and pandas.

```
import pandas as pd
import random
template = pd.read_excel('Databases/BSCSE-Full-Routine.xlsx')
teachers = pd.read_excel('Databases/Teachers-BSCSE.xlsx')
rooms=pd.read_excel('Databases/Room-info.xlsx')
sections=pd.read_excel('Databases/Sections-info-BSCSE.xlsx')
# lets decide the variables
no_of_faculty =  teachers.shape[0] # number of faculty members
teachers=teachers.fillna(0) # fill all NaN by 0
faculty_names = teachers["Faculty Member"].values.tolist()  # Faculty Names
theory_limit=teachers["Theory"].values.tolist() # limit of theory 0 if none
lab_limit=teachers["Lab"].values.tolist() # limit of labs 0 if none
# set global limits
max_course_limit=6
max_credit_limit=14
min_credit_limit=10
no_of_courses = template.shape[0]#gets the number of course
taken = [0] * no_of_courses
course_codes =  template["FormalCode"].values.tolist()
sections = template["SectionName"].values.tolist()
room = template["Room1"].values.tolist()
day1 = template["Day1"].values.tolist()
day2 = template["Day2"].values.tolist()
time1 = template["Time1"].values.tolist()
credits=[]
final_faculty_list=[]
## find out credits from course codes
for code in course_codes:
  c=(code.split(" ")[1].split("."))[0]
  if int(c)%2==0:
    credits.append(1)
  else:
    credits.append(3)
startTime=[]
endTime=[]
for duration in time1:
  tokens=duration.split("-")
  start=tokens[0]
  end = tokens[1]
  start=start.rstrip()
  start=start.lstrip()
  end=end.rstrip()
  end=end.lstrip()
  m=start.split(":")
  if m[2]=="AM":
    startTime.append(int(m[0])*100+int(m[1]))
  else:
    startTime.append(int(m[0])*100+1200+int(m[1]))
  m=end.split(":")
  if m[2]=="AM":
    endTime.append(int(m[0])*100+int(m[1]))
  else:
    endTime.append(int(m[0])*100+1200+int(m[1]))
assigned_teacher=template["Teacher1"].values.tolist()
# clear all assignments
for i in range(no_of_courses):
    assigned_teacher[i]=0
def isValid(s):
  # in this function we need to check whether the solution s violates any hard constraints
  # we have got random faculty members assigned in each course
  # lets populate courses for each faculty
  faculty_course_list={}
```

```
  for i in range(no_of_courses):
    if s[i] in faculty_course_list:
      if taken[i] == 0:
        list=faculty_course_list[s[i]]
        list.append(i)
        faculty_course_list[s[i]]=list
        taken[i] = 1
    else:
      if taken[i] == 0:
        list=[i]
        faculty_course_list[s[i]]=list
        taken[i] = 1

  return True
def quality(s):
  # in this function we need to check whether the solution s violates any soft constraints
  # we have got random faculty members assigned in each course
  # lets populate courses for each faculty
  faculty_course_list={}
  for i in range(no_of_courses):
    if s[i] in faculty_course_list:
      list=faculty_course_list[s[i]]
      list.append(i)
      faculty_course_list[s[i]]=list
    else:
      list=[i]
      faculty_course_list[s[i]]=list

  return True
  # now we have got course list for each of the faculty members, now lets check conflicts
  # lets first check if the total courses exceeds global limit
def generateRandomValid():
  s=[]
  faculty_course_list={}
  for i in range(no_of_courses):
    while(True):
      f=random.randrange(no_of_faculty)
      if f in faculty_course_list:
        list=faculty_course_list[f]
        if len(list)<max_course_limit:
          list.append(i)
          faculty_course_list[f]=list
          s.append(f)
          break
      else:
        list=[i]
        faculty_course_list[f]=list
        s.append(f)
        break
  return s
def RandomNeighborhoodSearch(maxIter):
  # lets create an initial solution
  s = generateRandomValid()

  if isValid(s) and quality(s):
    return s
final_faculty_list = RandomNeighborhoodSearch(1000)
assigned_faculty_names=[]
for i in range(no_of_courses):
  assigned_faculty_names.append(faculty_names[final_faculty_list[i]])
dataOut=pd.DataFrame(
    {'Course Code': course_codes,
     'Section': sections,
     'room': room,
     'Time': time1,
```

```
    'Day1': day1,
    'Day2': day2,
    'Faculty': assigned_faculty_names
   })
dataOut.to_excel("output-BSCSE-Faculty.xlsx")
```