A NOVEL APPROACH FOR AUTOMATIC TIMETABLE GENERATION

Mayuri Bagul[1], Sunil Chaudhari[2], Sunita Nagare[3], Pushkar Patil[4], Guide: Prof. K.S.Kumavat[5]

[1,2,3, 4](Computer, BVCOERI, Nashik),[5](HOD Computer Dept, BVCOERI, Nashik,(India)

ABSTRACT

The manual system of preparing time table in colleges is very time consuming and tedious task which usually ends up with various classes clashing either at identical room or with same teachers having more than one class at a time. Due to manual approach, proper use of resources is neither effective nor efficient. To overcome all these problems we propose to make an automated system with computer assisted timetable generator. The system will take various inputs like number of subjects, teachers, maximal lectures a teacher can conduct, priority of subject and topics to be covered in a week or a lecture, depending upon these inputs it will generate possible time tables for working days of the week, making optimal use of all resources in a way that will best suit the constraints. An appropriate timetable is then chosen from the optimal solutions generated.Timetable creation is a very arduous and time consuming task. To create timetable it takes lots of patience and man hours. Time table is created for various purposes like to organize lectures in school and colleges, to create timing charts for train and bus schedule and many more. To create timetable it requires lots of time and man power .In our paper we have tried to reduce these difficulties of generating timetable by Heuristic Algorithm.

Keywords: Component, Constraints, Resource Scheduling Algorithm, Heuristic Algorithm Time Table generation.

I. INTRODUCTION

The class timetabling difficulty could be a planning formula with nice interest and implications within the fields of operational analysis and AI. Most institutes subsume this drawback manually, i.e. an effort and error methodology is employed to line a timetable. whereas setting a timetable, importance is given to effective utilization of re that becomes a really exhausting task that has to be self-addressed a minimum of once a year by each educational institute.

1.1 Required

The schedule should meet the necessity of latest course addition and recently listed students to recent batches. This might end in schedule the whole plan yet again for its entire batches and to be regular in shortest doable time before the batches courses begin. Another drawback that happens whereas planning plan for exams, once multiple batches have examination on same day, they have to be schedules effectively taking under consideration all issues associated with facilities that area unit on the market to conduct these exams at the same time. The teaching staffs sometimes pay lots of your time in timetable generation and timetable management. the target of the Timetable Generator project was to develop a tool that permits institutes to dynamically generate timetables for colleges' internet access directly from "raw" schedule. These timetable generation codes conjointly regard the supply of lecturers and different re whereas making timetable. Moreover, changes are often merely created within the timetable as and once necessary counting on the supply of lecturers, replacement, students, and technicians, school rooms and lessons. Timetabling issues area unit a particular kind of planning drawback and area unit in the main involved with the assignment of events to slots subject to constraints with the resultant answer constituting a timetable. Wren (1996) demarcated timetabling as: "Timetabling is that the allocation, subject to constraints, of given to things being placed in house time, in such the way on appease as nearly as doable a collection of covetable objectives." The constraints in timetabling are often divided into 2 categories: exhausting and soft. Exhausting constraints cannot be profaned. Soft constraints don't seem to be imperative however their satisfaction is additional covetable so as to supply a decent quality timetable. A general timetabling drawback made from distribution variety of events like course, examinations, lectures, work meeting etc. into a restricted range of slots and rooms whereas minimising the violations of a collection of constraints. Heuristic optimisation ways area unit expressly aimed toward good possible answers which will not be optimum wherever complication of restricted time on the market doesn't enable actual solution. The empirical analysis of heuristic methodology is predicated on analytic drawback concerned within the tough worst case result. In its simplest kind the planning task made from mapping category, teacher and area combos onto time slots. Time Table creation is exhausting and time consuming process for the faculty in charge. At present this is done manually as there are no efficient time table generators. While framing time table the basic problems are slot clashes. Allotting periods itself is so tedious that allotting the whole time table is not at all efficient when done manually. So, even the software which has already been created does not comply with the constraints. The present system is therefore time consuming, tedious process requiring manual labor and simultaneously, having less flexibility. Therefore we are proposing a system for Automatic timetable generation. The algorithm based application allots periods in such a way that no period clashing or faculty period clash is met. The subjects are paired in such a way that teachers remain associated and no period clash appears. Subjects are allotted as per priority based on the number of lectures per week of that subject. High priority subjects are given preference for number of periods per week and are allotted accordingly. Also the corresponding labs are allotted for the suitable day. This helps satisfy the constraint of number of subjects per day both theory and lab for a faculty. We assign subjects to teacher as per their seniority and as per their preference. This is done starting from the high priority subject first and then the decreasing priority subjects and then the unassigned subjects. This keeps utmost care of the designation of faculties and their p. This procedure creates not only a feasible Time-Table for the department but also an optimal one. Output of the application is not only the class time table but also faculty timetable, both the optimum ones. The proposed system is based on heuristic algorithm that takes values and manages the constraints and source scheduling one by one. The system generates a timetable for the lecture courses as well as the staff timetable. It distributes workload of lectures equally among all the specified time slots. It prioritizes the lectures according to customized antecedence. If lecture cannot be adjusted then it can be moved up in higher priority slot until adjusted accordingly. In our system we have also implemented features other than the Timetable generation, such as a forum for student-teacher interaction, where students can put their queries and teachers can respond. Also teachers can upload eBook's and presentations for the students. This will prove quite helpful for the students as they can have access to these study materials any time, any place and get their queries solved and refer to those anytime again. Also students will be able to provide reviews for all the lectures and teachers. This will be helpful in getting an idea whether the timetable is being followed and how do the students rate the lectures based on their understanding of the subjects and topics taught during those lectures. These reviews will also be helpful for the staff to understand to what extend do the students comprehend and what topics they find difficult to understand during the lectures. Then the teachers will be able to accordingly schedule and reschedule the lectures for the students.

II. PROBLEM STATEMENT

The time tabling problem can be modeled as a constraint satisfaction problem with many parameters and loose constraints. These constraints have to be modeled in a format that can be handled efficiently by the scheduling algorithm. The scheduling involves allowing for a number of pair wise restrictions on which tasks can be done simultaneously. For instance, in attempting to schedule classes at an institute, two courses taught by the same faculty member cannot be scheduled for the same time slot. Similarly, two courses that are required by the same group of students also should not conflict.

III. LITERATURE SURVEY

In some years two main approaches have been successful for implementing the timetabling problem. The first approach is based on local search procedures method such as simulated annealing, tabu search and genetic algorithms. These methods express constraints as some value of functions, which are minimized by a heuristic search of better solutions in reference of some initial feasible solution. The second approach is based on constraint programming (CP). Its main advantage is declaratively a direct statement of the constraints serves as part of the program. This makes the program easy to modify, which is critical in timetabling problems. The constraints are handled through a system of constraint propagation, which decrease domains of variables, coupled with backtracking search. The main disadvantages of these approaches are

1. Difficulties with expressing hard and soft constraints,

2. The need to determine their parameters through experimentation and

3. Possible problems with improving the initial feasible solution,

Which – as a rule – may be determined without problem? An attempt to overcome the drawbacks along soft constraints was discussed, successfully combined local search with constraint satisfaction to decrease their drawbacks. A custom-tailored distribution strategy is able to introduce soft constraints during a search, leading immediately to a “good” timetable; giving the ability to effectively optimize the timetable.

A. Bhaduri A [1], evolutionary techniques have been used to solve the time table scheduling problem. Methodologies like Genetic Algorithms, Evolutionary Algorithms etc. have been used with mixed success. In this paper, we have reviewed the problem of educational time table scheduling with genetic algorithm. We have further solved the problem with a mimetic hybrid algorithm, genetic artificial immune network and compare the result with that obtained from genetic algorithm. Results show that GAIN is able to reach the ideal feasible solution faster than that of GA.

B. DiptiShrinivasan [2], Find a feasible tutorial timetable in a large university department is a challenging problem faced repeatedly in educational establishments. This paper represents an evolutionary algorithm (EA) based approach to solving a strong constrained university timetabling problem. The move toward a problem-specific chromosome representation. Heuristics and also context-based reasoning have been used for received feasible timetables in a reasonable computing time. An intelligent flexible mutation scheme has been employed for fast-moving up the convergence. The comprehensive course timetabling system presented in this paper has been approved, tested and discussed using real world data from a large university.

C. AnujaChowdhary [3] introduces a practical timetabling algorithm capable of taking care of both strong and weak constraints effectively, used in an automatic timetabling system. So that each teacher and student can view their timetable once they are final for a given semester but they do not edit them. Timetable Generation System generates timetable for every class and teacher, in keeping with the convenience calendar of teachers, availability and capacity of physical sources and some rules applicable at distinct classes, semesters, teachers and subject's level.

D. Anirudha Nanda [4], proposes a general solution for the timetabling problem. Most heuristic proposed previous approaches the difficulties from the students' point of view. This solution, however, works from the lecture's point of view i.e. lecturer availability for a given time slot. While all the strong constraints (e.g. the availability of teachers, etc.) are resolved rigorously, the scheduling solution presented in this paper is a flexible one, with a primary aim to solve the issues of clashes of lectures and subjects, pertaining to teachers.

E. A.Elkhyari [5], the proposed algorithm aids solving the timetabling problem while giving import to teacher availability. This algorithm uses a heuristic approach to give a overall solution to school timetabling difficulties. It initially uses randomly generated subject sequence to make a temporary time table. If the teacher has been apportion more than the allowed maximum lectures the subjects is moved toward into a Clash data structure. To avoid cycling to improve the search, this variable selection criterion can be randomized.

F. [6], Harmony search algorithm (HSA) is a meta-heuristic Population-based method that is developed by Geem et al.

G. [7], the meta-heuristics were classified based on the number of solutions that are used at the same time into local searchbased (or trajectory) methods (i.e., hill climbing, tabu search, simulated annealing) and population-based methods (i.e., genetic algorithm, harmony search algorithm, ant colony optimization).

H. [8] The MA has come to light as population-based metaheuristic algorithm by means of hybridizing the natural(population-based) and cultural (local search-based) selection Principles drawing from the principles of natural selection and the notion of an meme.

I. [9] In a recent comprehensive survey to examine timetabling, Qu et al. suggested "There are many research directions generated by considering the hybridization of meta-heuristic methods particularly between population-based methods and other approaches."

IV. EXISTING SYSTEM

Time Table creation is exhausting and time consuming process for the faculty in charge. At present this is done manually as there are no efficient time table generators. While framing time table the basic problems are slot clashes. Allotting periods itself is so tedious that allotting the whole time table is not at all efficient when done manually. So, even the software which has already been created does not comply with the constraints. The present system is therefore time consuming, tedious process requiring manual labor and simultaneously, having less flexibility. Therefore we are proposing a system for Automatic timetable generation. The algorithm based application allots periods (theory as well as labs) in such a way that no period clashing or faculty period clash is met. The subjects are paired in such a way that teachers remain associated and no period clash appears. Subjects are allotted as per priority based on the number of lectures per week of that subject. High priority subjects are given preference for number of periods per week and are allotted accordingly. Also the corresponding labs are allotted for the suitable day. This helps satisfy the constraint of number of subjects per day both theory and lab for a faculty. We assign subjects to teacher as per their seniority and as per their preference. This is done starting from the high priority subject first and then the decreasing priority subjects and then the unassigned subjects. This keeps utmost care of the designation of faculties and their p. This procedure creates not only a feasible Time-Table for the department but also an optimal one. Output of the application is not only the class time table but also faculty timetable, both the optimum ones. The proposed system is based on heuristic algorithm that takes values and manages the constraints and resource scheduling one by one. The system generates a timetable for the lecture courses as well as the staff timetable. It distributes workload of lectures equally among all the specified time slots. It prioritizes the lectures according to customized antecedence. If lecture cannot be adjusted then it can be moved up in higher priority slot until adjusted accordingly. In our system we have also implemented features other than the Timetable generation, such as a forum for student-teacher interaction, where students can put their queries and teachers can respond. Also teachers can upload e-books and presentations for the students. This will prove quite helpful for the students as they can have access to these study materials any time, any place and get their queries solved and refer to those anytime again. Also students will be able to provide reviews for all the lectures and teachers. This will be helpful in getting an idea whether the timetable is being followed and how do the students rate the lectures based on their understanding of the subjects and topics taught during those lectures. These reviews will also be helpful for the staff to understand and analyze to what extend do the students comprehend and what topics they find difficult to understand during the lectures. Then the teachers will be able to accordingly schedule and reschedule the lectures for the students.

V. ALGORITHM

Algorithmic strategy contains heuristic algorithm, variable used for that and different assumption for satisfying goals:

HEURISTIC ALGORITHM:-

The term **heuristic** is used for algorithms which find solutions among all possible ones, but they do not guarantee that the best will be found, therefore they may be considered as approximately and not accurate algorithms. Thesealgorithms, usually find a solution close to the best one and they find it fast and easily. Sometimes these algorithms can be accurate, that is they actually find the best solution, but the algorithm is still called heuristic until this best solution is proven to be the best. The method used from a heuristic algorithm is one of the known methods, such as greediness, but in order to be easy and fast the algorithm ignores or even suppresses some of the problem's demands.

Consider the example of automatic timetable generator.

VARIABLE USED:-

--Time slots of the time tables:- ts1, ts2, ts3....,tsn

--List of Subjects:- s1,s2,s3, ..., sn

--Teachers:- t1,t2,t3, ..., tn

--Batches of students:- c1,c2,c3, ..., cn

--Flags indicating finalized timeslots :- tsf1, tsf2, tsf3, ..., tsfn

--Data structure to hold Final Timetable:-final_tt

--Count for day of week: Daycount

--Number of days of the week:- n

--Data structure to hold Subject-clash within the day:- clash

--Each element of Clash data structure:-clash_element

--Data structure for Subject-clash across days:-Dayclash

--Each element of Dayclash data structure:-day_clash_element

--Subject contained in dayclash:-sdc

--Teacher associated with subject in dayclash:-tdc

--Max number of lectures of subject si in the week:-k

--Counter for the number of subjects:-counter_sub

--Random number indicating random slot allotment for subject:-rand_sub_allot

--Data structure to hold randomly allotted subject:-rand_sub_seq

--Data structure to hold all subjects:-init_sub

ASSUMPTION:-

This algorithm is designed to solve and generate school time tables. The following is a list of assumptions made while developing this algorithm:

- The algorithm produces optimum outputs in a five-day week.
- The number of subjects (s1, s2, ...,sn) need to be finalized before the algorithm begins execution.
- Number of teachers (t1, t2, ...,tn) entered before execution of the algorithm are assumed to be constant and cannot be changed during or after the algorithm has been executed.
- Any change in the above two assumptions will require a new generation of Timetable for the changed data.
- In each time table, all time-slot is filled with, a unique combination of subjects without any repetition of subjects.
- Any teacher is allowed at most ‘k’ number of lectures in a week. The value of k is accepted before execution of the algorithm.
- It is assumed that a teacher cannot take more than one lecture for the same class in a day.
- Timeslots ts1, ts2, ... ,tsn once entered at the beginning cannot be changed throughout the execution.
- Every day in the week is assumed to have equal number of time slots.
- Classrooms for any batch id fixed throughout the day.

USER-INPUT:-

--Time slots of the time tables:- ts1, ts2, ts3.... Tsn

--List of Subjects:- s1,s2,s3, ..., sn

--Teachers:- t1,t2,t3, ..., tn

--Max number of lectures of subject sI in the week:- k

--Batches of students:- c1,c2,c3, ..., cn

--Count for day of week: Daycount

SOME IMPORTANT EXPECTED RESULTS:

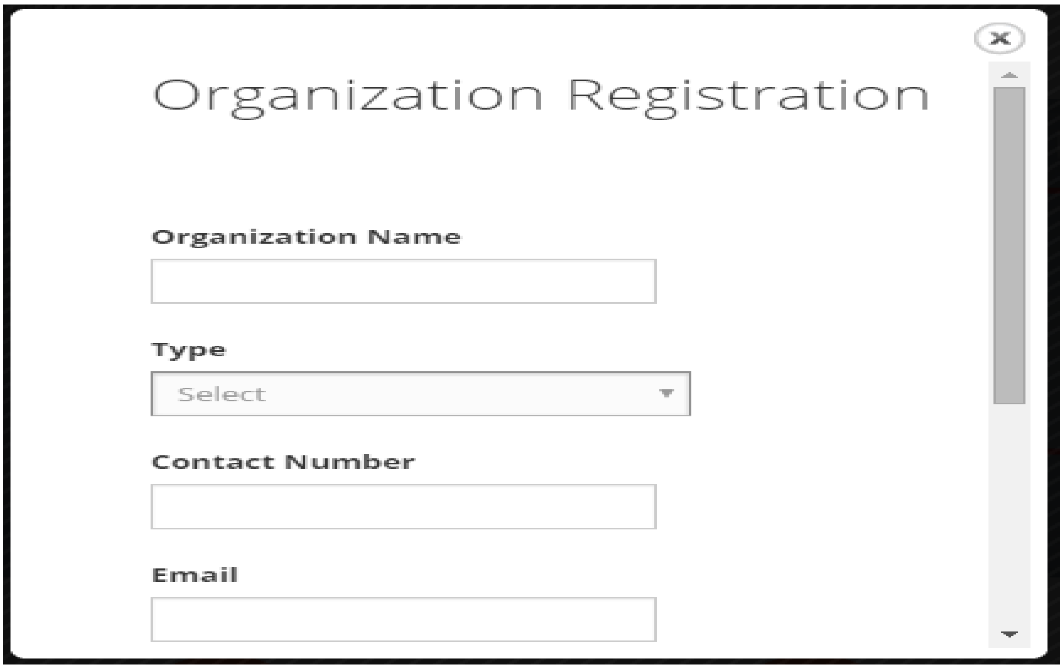
Expected result for the system is shown in figure 3 as follows-

	Mon	Tue	Wed	Thurs	Fri
TS1	T1,S1	T1,S2	T2,S3	T3,S4	T4,S5
TS2	T4,S5	T1,S1	T1,S2	T2,S3	T3,S4
TS3	T3,S4	T4,S5	T1,S1	T1,S2	T2,S3
TS4	T2,S3	T3,S4	T4,S5	T1,S1	T1,S2
TS5	T1,S2	T2,S3	T3,S4	T4,S5	T1,S1

VI. IMPLEMENTED MODULE

6.1 User Module of Organization Registartion

Following picture shows the Organization Registration page.



Organization Registration

Organization Name

Type

Select

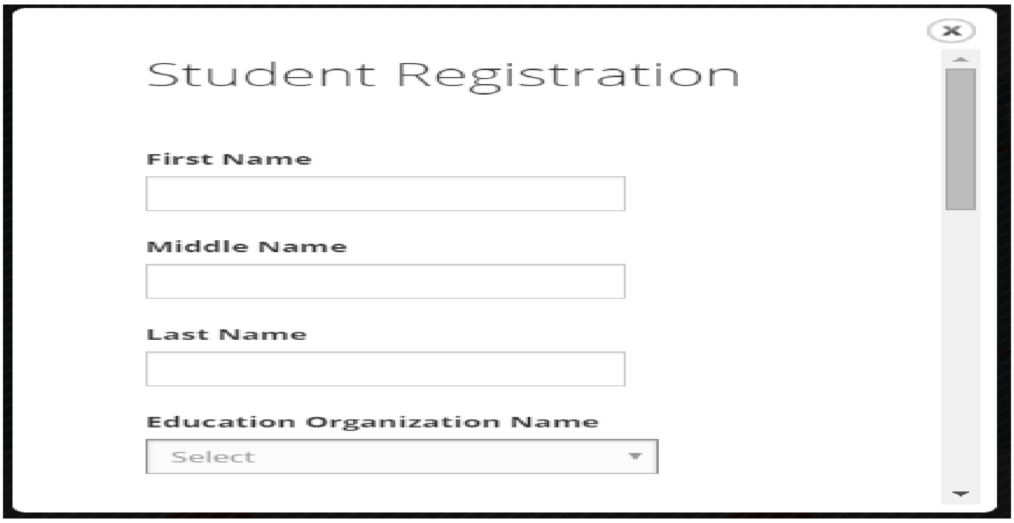
Contact Number

Email

Fig. : Organization Registration

In Organization Registration ,we can fill up the information of Organization such as Name of Organization,type,contact number, email address etc.

6.2 User Module of Student Registartion



Student Registration

First Name

Middle Name

Last Name

Education Organization Name

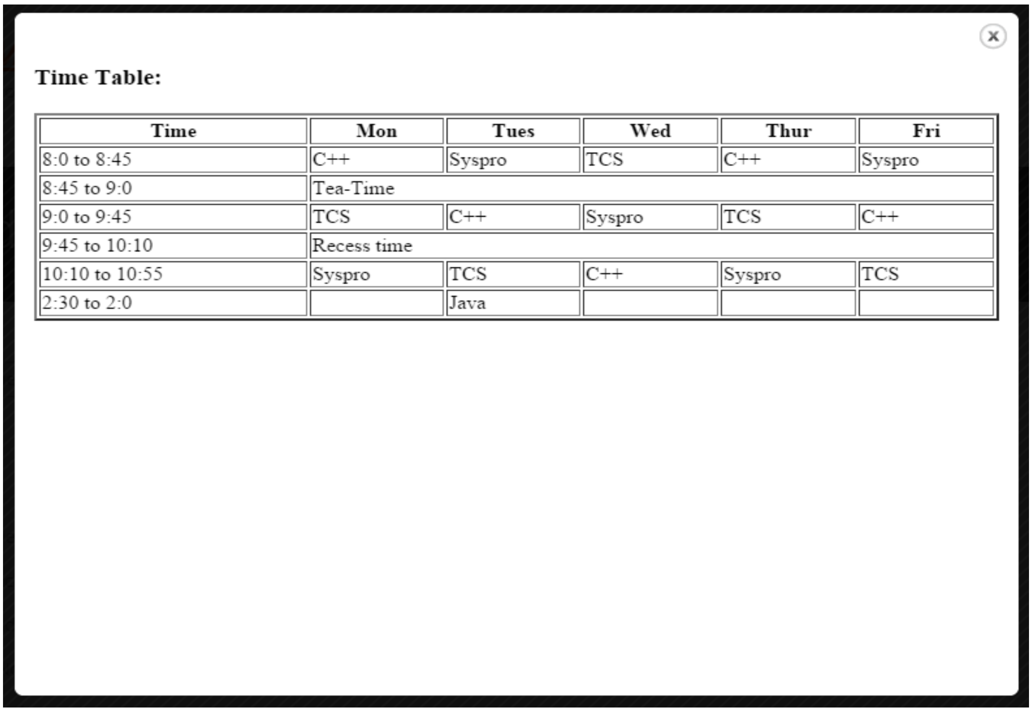
Select

Fig. : Student Registration

Above picture shows the Student Registration page. In student registration, we can fill up the information of student such as full name of student, organization name, contact number, address etc.

6.3 Graph & Generated time table

Following picture shows the Automatic Timetable Generated page.



Time Table:

Time	Mon	Tues	Wed	Thur	Fri
8:0 to 8:45	C++	Syspro	TCS	C++	Syspro
8:45 to 9:0	Tea-Time				
9:0 to 9:45	TCS	C++	Syspro	TCS	C++
9:45 to 10:10	Recess time				
10:10 to 10:55	Syspro	TCS	C++	Syspro	TCS
2:30 to 2:0		Java			

Fig. : Timetable

In organization level, admin fill up the information staff i.e subject specialty, qualification, contact number etc. After that staff canfill up the information of shows studentsdetails, shows attendance details, uploadnotes, fill up all the information of classroom,lab,library details,timeshift information. Using all this information generate the time table.Result will be generated in graphical chart; here this chart is generated from using information.

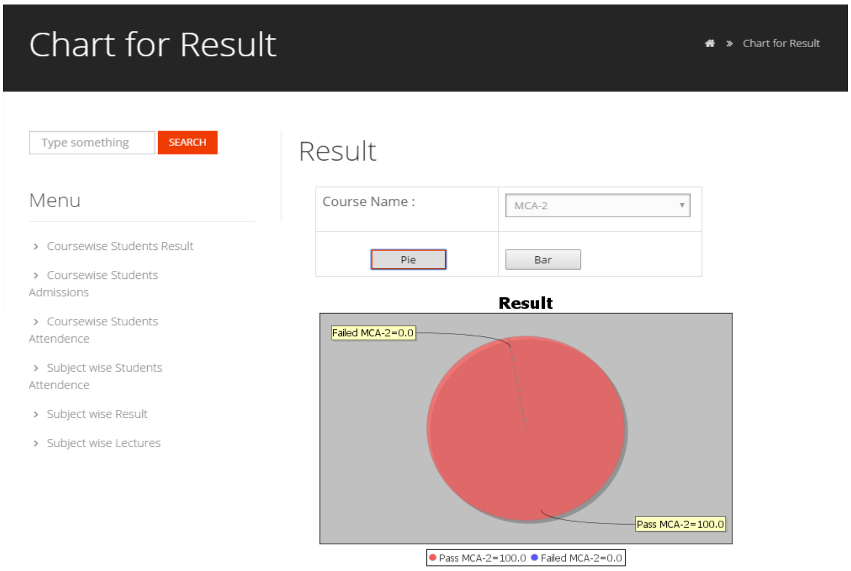


Chart for Result
» Chart for Result
Type something
SEARCH
Menu
› Coursewise Students Result
› Coursewise Students Admissions
› Coursewise Students Attendence
› Subject wise Students Attendence
› Subject wise Result
› Subject wise Lectures
Result
Pie
Bar
Result
Failed MCA-2=0.0
Pass MCA-2=100.0
● Pass MCA-2=100.0 ● Failed MCA-2=0.0

Course Name : MCA-2

VII. CONCLUSION

This application will simplify the process of time table generation smoothly which may otherwise needed to done using spread sheet manually possibly leading to constraints problem that are difficult to determine when time table is generated manually. The intention of the algorithm to generate a time-table schedule automatically is satisfied. The algorithm incorporates a number of techniques, aimed to improve the efficiency of the search operation. It also, addresses the important hard constraint of clashes between the availability of teachers. The non-rigid soft constraints i.e. optimization objectives for the search operation are also effectively handled. Given the generality of the algorithm operation, it can further be adapted to more specific scenarios, e.g. University, examination scheduling and further be enhanced to create railway time tables. Thus, through the process of automation of the time-table problem, many an-hours of creating an effective timetable have been reduced eventually. The most interesting future direction in the development of the algorithm lies in its extension to constraint propagation. When there is a value assigned to a variable, such assignment can be propagated to unassigned variables to prohibit all values which come into conflict with the current assignments. The information about such prohibited values can be propagated as well.

VIII. ACKNOWLEDGMENT

It gives us great pleasure in presenting the preliminary project report on ‘A Novel Approach for Automatic Timetable Generation’.

I would like to take this opportunity to thank my internal guide Prof. K.S.Kumavatfor giving me all the help and guidance I needed. I am really grateful to them fortheir kind support. Their valuable suggestions were very helpful.

I am also grateful to Prof. H.D.Sonawane, Head of Computer Engineering Department,CollegeName for his indispensable support, suggestions.

In the end our special thanks to all staff members for providing various resourcessuch as laboratory with all needed software platforms, continuous Internet connection,for Our Project.

Miss. Mayuri R. Bagul
Mr. Sunil C. Chaudhari
Miss. Sunita N. Nagare
Mr. Pushkar R. Patil

REFERENCES

[1] Mayuri Bagul[1], Sunil Chaudhari[2], Sunita Nagare[3], Pushkar Patil[4], K.S.Kumavt[5],"A Novel Approach For Automatic Timetable Generation", at IJCA (International Journal of Computer Applications) (0975 – 8887) Volume 127 – No.10, October 2015.

[2] Bhaduri a "university timetable scheduling using genetic algorithm". Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference

[3] DiptiShrinivasan "automated time table generation using multiple context reasoning for university modules" Published in: evolutionary computation, 2002. cec '02. proceedings of the 2002 congress on (volume:2)

[4] AnujaChowdhary "TIME TABLE GENERATION SYSTEM" .Vol.3 Issue.2, February- 2014, pg. 410-414

[5] Anirudha Nanda "An Algorithm to Automatically Generate Schedule for School Lectures Using a Heuristic Approach".International Journal of Machine Learning and Computing, Vol. 2, No. 4, August 2012.

[6] A. Elkhyari, C. Gu´eret, and N. Jussien, "Solving dynamic timetabling problems as dynamic resource constrained project scheduling problems using new constraint programming tools. In Edmund Burke and Patrick De Causmaecker, editors, Practice And Theory of Automated Timetabling, Selected Revised Papers," pp. 39–59. Springer- Verlag LNCS 2740, 2003.