SolvingTime-Table Scheduling Problem by Novel Chromosome RepresentationUsing Genetic Algorithm

Paresh M. Chauhan
Department of Information Technology,
Laxmi Institute of Technology,
Sarigam, Gujarat, India
paresh.lit@lvs.co.in

Kashyap B. Parmar
Department of Information Technology,
Laxmi Institute of Technology,
Sarigam, Gujarat, India
kashyap.lit@lvs.co.in

Mahendra B. Mendapara
Surat, Gujarat, India
mahendra.mendapara@gmail.com

***Abstract*—this proposed research i.e. Time–Table Scheduling Using Genetic Algorithm with novel group based chromosome representation for educational institute having number of branches of Engineering. It is desirable to have no clashes of their lectures or laboratory sessions with each other. Using Genetic Algorithm designs highly optimize time table form randomly generated large size of population. keeping all above stated views in mind, our proposed system basically deals with the 2 department (IT/CP) of LIT college, Sarigam.**

Keywords— Genetic Algorithm, Time-Table Scheduling, Group Based Chromosome Representation.

I. INTRODUCTION

In most of the educational institute Time-Table is generated manually. It is time consuming and requires lots of paper work also need more man power. We study the current Time-Table of IT department of LIT College.

We are examine that current Time-Table having problem, faculty have consecutive lecture and lab, the break time and gap between two lecture was not maintain.

We are the study the current hierarchy of faculty post of the college, and according to that examine number of working hour for particular faculty, the workload of subject (number of lecture and lab).

- Lecture: 1- hour, 1-faculty
- Lab: 2-hour, 2-faculty

This research is to design and implement a semester course time table using Genetic Algorithm.

We go for GA because of the below reasons [8]:

1. Alternate solutions are too slow or overly complicated
2. Need an exploratory tool to examine new approaches
3. Problem is similar to one that has already been successfully solved by using a GA
4. Want to hybridize with an existing solution
5. Benefits of the GA technology meet key problem requirements

Difference between GA and Classical Optimization is given in below table 1.

Table 1 Difference of GA and Classical Optimization

GA	Classical Optimization
Works with coding of variable	Work with coding it self
Searches form set of solution	Searches using only one solution
Probabilistic Rules	Deterministic Rules
General method for all category problems	Can be applied to specific category problems

This research is used for assigning time-slots to a given set of courses. The constraints include avoiding clashes of time-slots, assigning appropriate no. of slots and no. of hours to the courses etc. there are two types of constraints. One is soft constraints and other is hard constraint. A soft constraint means the constraints which should be satisfied but not mandatory whereas the hard constraint means the constraints must be satisfied for a solution. In proposed representation we considered soft constraints which are 1) classes scheduled within preferred hours; 2) avoid long runs of consecutive lectures. Hard constraints are 1) All classes must be scheduled; 2) No classes/lecturers are allocated twice at same time.

II. REVIEW OF LITERATURE

Wang, Cheng [1],their research was on University Timetable Problem (UTP).They divide their research in two phase: one is the basic teaching tasks fortitude and another is the optimization of basic teaching time, Simplify the task of basic teaching methods and processes. Using genetic algorithm, they discussed the steps in detail of optimizing the basic teaching time, Stretches the algorithm of the fitness function, genetic crossover, variation and generation of initial population, and verified the value of the algorithm.

Lukas, S. et. al. [2] according to them scheduling problem is quite complex problem. All the constraints must have been taken care to arrange a optimal schedule, such as a great number of classes, availabilities of lecture and courses. To

978-1-4799-7075-9/15/$31.00 ©2015 IEEE

over whelm these difficulties, genetic algorithm with heuristic search is proposed in their paper. Their proposed method was verified many times, and the results show that even with small population, the best schedule still can be achieved.

Ghaemi, S. et. al. [3] they conclude that a highly constrained combinatorial problem, like the timetable, can be solved by evolutionary methods. Among the evolutionary computation (EC) algorithms, a genetic algorithm (GA) for solving university course timetabling problems was applied in their paper. Their main goal was to minimize as many conflicts as possible in the timetable. For this purpose two approaches - cooperative GA and modified GA - were applied. Results they got show that the modified GA (MGA) method was meaningfully enhanced algorithm performance with modified basic genetic operators. operators chosen intelligently improve overall algorithm psilas behavior.

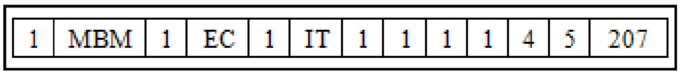
Sigl, B., Golub, M.[4] in their paper they described genetic algorithm for solving a timetable scheduling. The algorithm was tested on small and large instances of the problem. Modification of basic genetic operators improved the performance significantly. Intelligent operators restrain the creation of new conflicts in the individual and improve the overall algorithm's behavior.

Els, R. et. al. [5] their study presented in paper takes a different approach and aims at developing a system, namely, an evolutionary algorithm (EA), hyper-heuristic, that generalizes over a set of problems rather than only producing a feasible timetable for one or more of the problems. The results of a first attempt to implement an EA hyper-heuristic to solve the curriculum based university course timetabling problem were presented in their paper. For time table problem, the optimal heuristic combination evolved is used to construct a solution. The EA hyper-heuristic was tested on the benchmark set of curriculum based course timetabling problems used for the second international timetabling competition. The system evolved feasible solutions for all 14 problems. Their study also revealed areas for further improvement.

Wang Xiao Yun; Wang Feng Kun; [6] they improved the arrange timetable algorithms in common need. Aiming at the problem, classes divide into groups and elitist strategy with difference chromosome methods based on genetic algorithms (GA) was proposed. They compared with standard genetic algorithms by simulation. The experimental results show that it illustrates a good prospect of application and extension.

III. PROPOSED CHROMOSOME REPRESENTATION

Each chromosome will be processed in group of thirteen genes[7].K. B. Parmar, H. B. Prajapati, V. K. Dabhi[7] used Novel group based chromosome representation for larger chromosome.One of those groups of thirteen genes is shown in Fig. 1.



1	MBM	1	EC	1	IT	1	1	1	1	4	5	207

Fig. 1 Group of thirteen genes

Each Gene has its particular predefined meaning which is given in the table 2:

Table 2 Code and Meaning

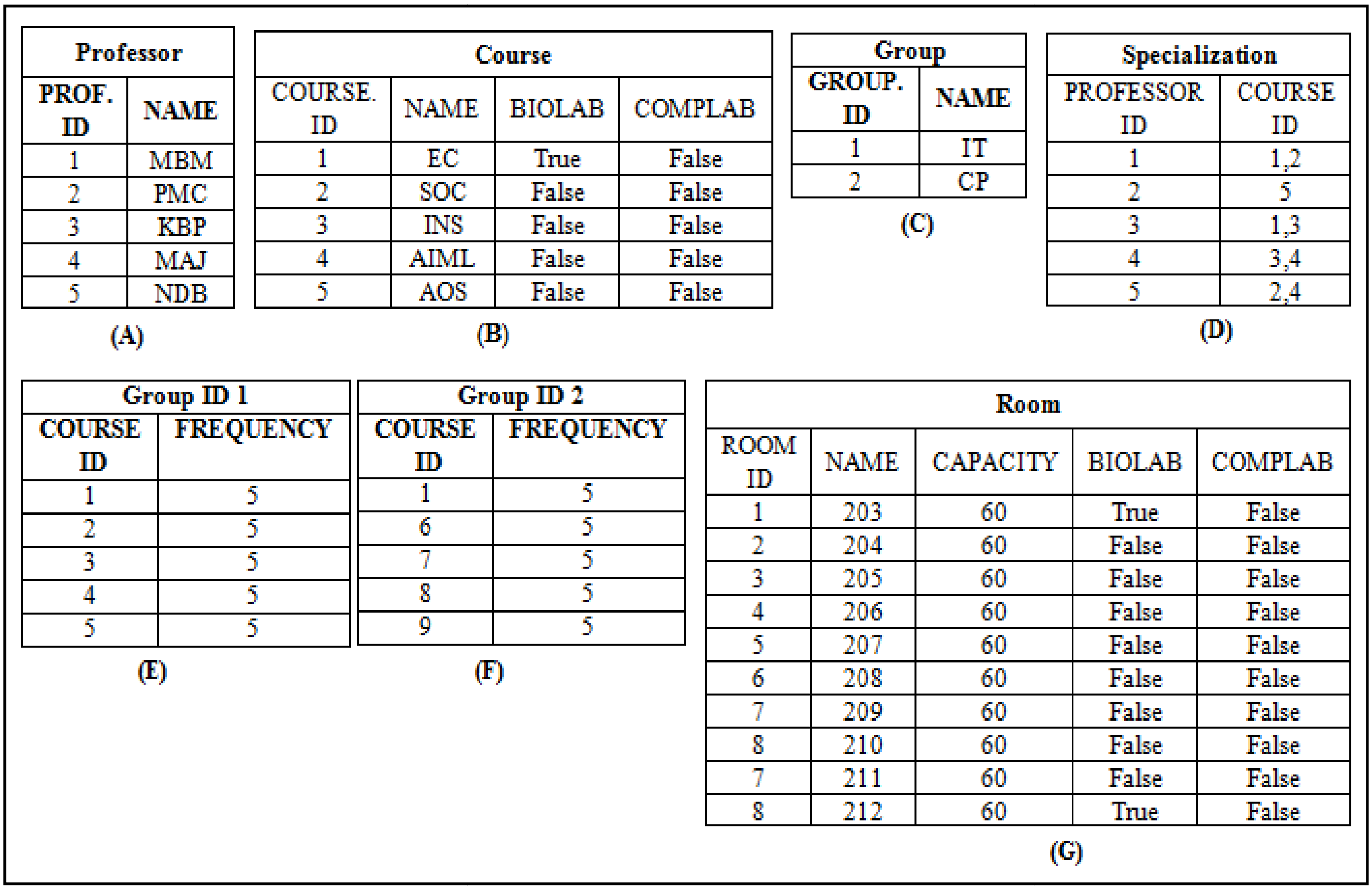
SR. NO.	CODE	MEANING	SR. NO.	CODE	MEANING
1	1	PROF. ID	8	1	COURSE ID
2	MBM	NAME	9	1	GROUP ID
3	1	COURSE ID	10	1	COURSE ID
4	EC	NAME	11	4	FREQ.
5	1	GROUP ID	12	5	ROOM ID
6	IT	NAME	13	207	NAME
7	1	PROF. ID			

In Fig 1first gene represents professor id and second gene represents the name of professor that are shown in Fig. 2 as a table (A). Third gene represents course id and forth gene represents the name of course that are shown in Fig. 2 as a table (B). Fifth gene represents group id and sixth gene represents the name of group (department)that are shown in Fig. 2 as a table (C).Seventh gene represents professor id and eighth gene represents the name of course in which the professor having specialization that are shown in Fig. 2 as a table (D). Ninth gene represents group id, tenth gene represents the course id and eleventh gene represents the frequency of that course in that group that are shown in Fig. 2 as a table (E) and table (F). Twelfth gene represents room id and thirteenth gene represents the name of room that are shown in Fig. 2 as a table (G).

In table (B)shown in Fig. 2, Biology lab and computer lab have two value. One is true which describes that corresponding course requires that lab and another is false which describes that corresponding course doesn't require that lab. In table (G) shown in Fig. 2, Biology lab and computer lab also have two value. One is true which describes that corresponding room have facility to conduct that lab and another is false which describes that corresponding room haven't facility to conduct that lab. XML representation of group of thirteen genes representation is given below:

```
<prof id="1" name=“MBM."></prof>
<course id="1" name="EC"></course>
<group id="1" name=“IT"></group>
<specialization prof="1" course="1"></specialization>
<lesson group="1" course="1" frequency=“4"></lesson>
 <room id="5" name="207"></room>
```

All tables which describe the meaning of every gene are shown in below Fig. 2.



Professor	
PROF. ID	NAME
1	MBM
2	PMC
3	KBP
4	MAJ
5	NDB

(A)

Course			
COURSE. ID	NAME	BIOLAB	COMPLAB
1	EC	True	False
2	SOC	False	False
3	INS	False	False
4	AIML	False	False
5	AOS	False	False

(B)

Group	
GROUP. ID	NAME
1	IT
2	CP

(C)

Specialization	
PROFESSOR ID	COURSE ID
1	1,2
2	5
3	1,3
4	3,4
5	2,4

(D)

Group ID 1	
COURSE ID	FREQUENCY
1	5
2	5
3	5
4	5
5	5

(E)

Group ID 2	
COURSE ID	FREQUENCY
1	5
6	5
7	5
8	5
9	5

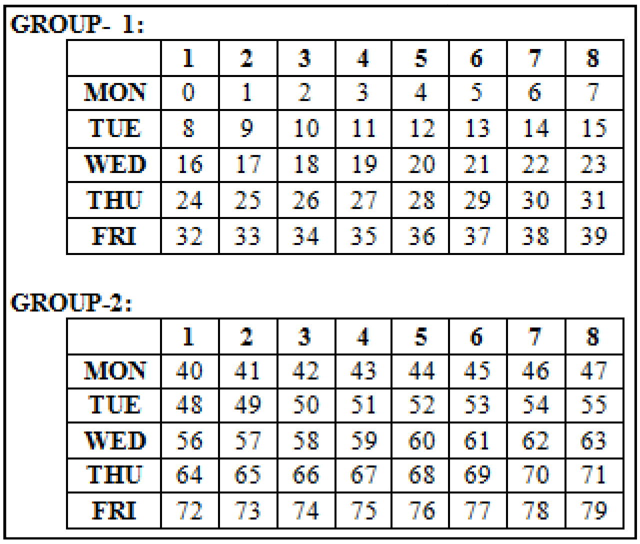
(F)

Room				
ROOM ID	NAME	CAPACITY	BIOLAB	COMPLAB
1	203	60	True	False
2	204	60	False	False
3	205	60	False	False
4	206	60	False	False
5	207	60	False	False
6	208	60	False	False
7	209	60	False	False
8	210	60	False	False
7	211	60	False	False
8	212	60	True	False

(G)

Fig.2Group of tables where each table having different gene value

Suppose we have to do chromosome representation of two (Group 1 and Group 2) time tables and each time table contains (5 days X 8 time slots) 40 lectures. So, two time table contains 40 X 2 that is 80 lectures. Single lecture is represented by group of thirteen genes (e.g. shown in Fig. 1). So, we require (80) X (group of 13 genes) to generate one solution which represents two times tables (shown in Fig. 3).



GROUP- 1:

	1	2	3	4	5	6	7	8
MON	0	1	2	3	4	5	6	7
TUE	8	9	10	11	12	13	14	15
WED	16	17	18	19	20	21	22	23
THU	24	25	26	27	28	29	30	31
FRI	32	33	34	35	36	37	38	39

GROUP-2:

	1	2	3	4	5	6	7	8
MON	40	41	42	43	44	45	46	47
TUE	48	49	50	51	52	53	54	55
WED	56	57	58	59	60	61	62	63
THU	64	65	66	67	68	69	70	71
FRI	72	73	74	75	76	77	78	79

Fig.3 Time Table of Two Groups

From 0 to 39 means 40 X (group of 13 genes) represents time table of Group 1 and from 40 to 79 means 40 X (group of 13 genes) represents time table of Group 2 which is shown in Fig. 3.

Initial Population:

In initial population [9], as we discussed before we require (80) X (group of 13 genes) to generate one individual. Initially we generate 88 individuals that is initial population shown in below Fig. 4.

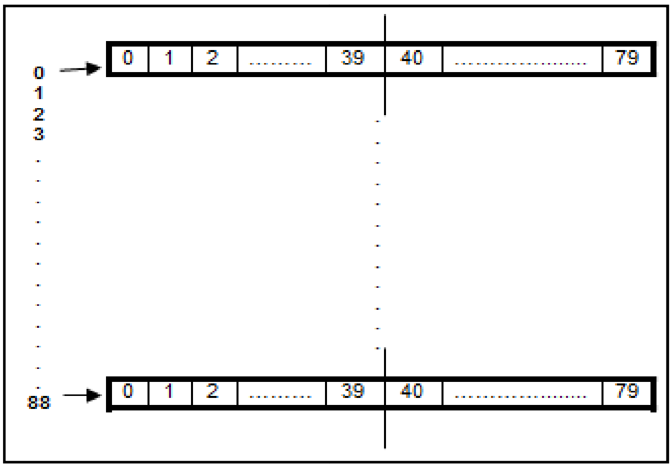


Fig.4 Population of 88 individuals

Fitness Function:

Fitness [11]of the time table is calculated by below eq. 1:

$$F(X) = \frac{\sum_{i=1}^{n} Xi*W}{t} \tag{1}$$

Where,

x = Time-Table under evaluation.
w = number of Constraints.
t = total fitness value.

Crossover:

Crossover is performed between two parents [10].This research work used well known 1-point crossover.Group-1 of Parent-A and Parent-Bremain as it is whereasGroup-2of Parent-A and Parent-B is swapped. Finally, two new children are generated that are Child-AB and Child-BA which are shown in Fig. 5.

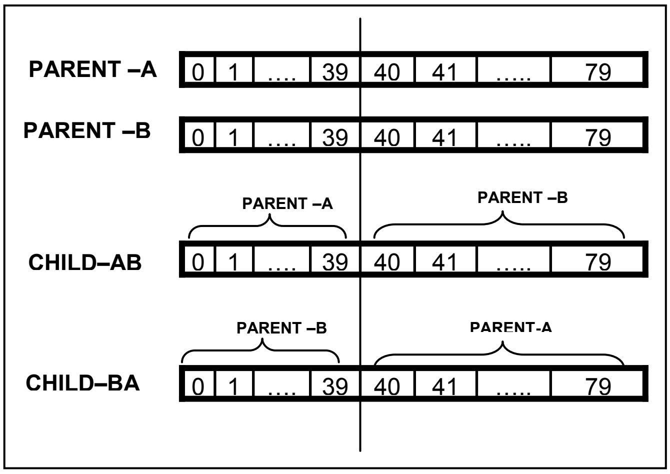


Fig.5Crossover

Mutation:

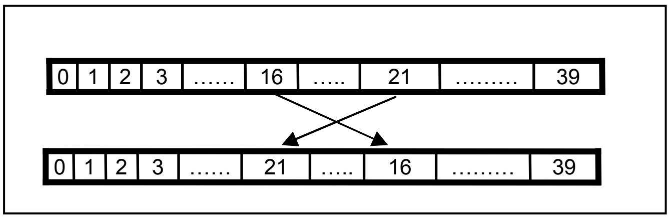
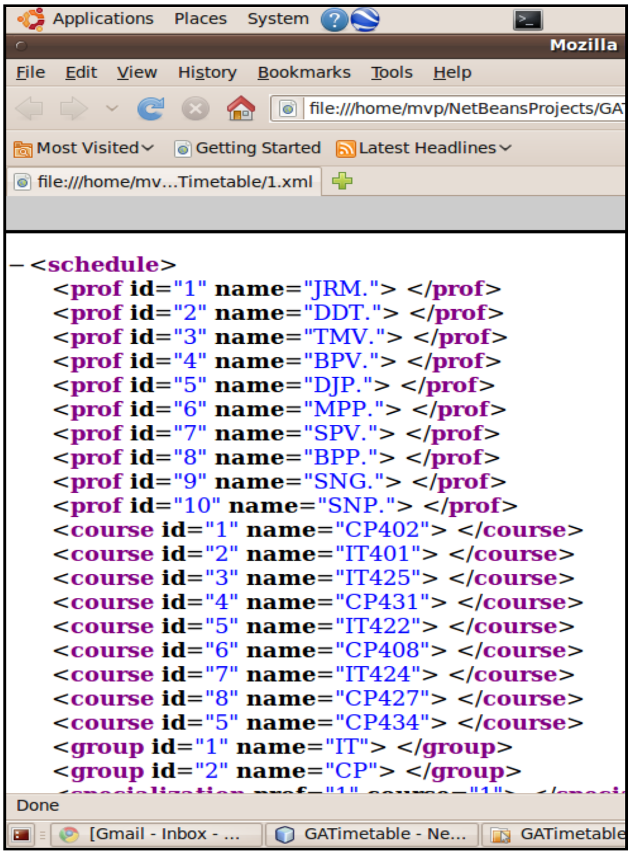


Fig. 6 Mutation

IV. IMPLIMANTATION AND RESULT

Input:

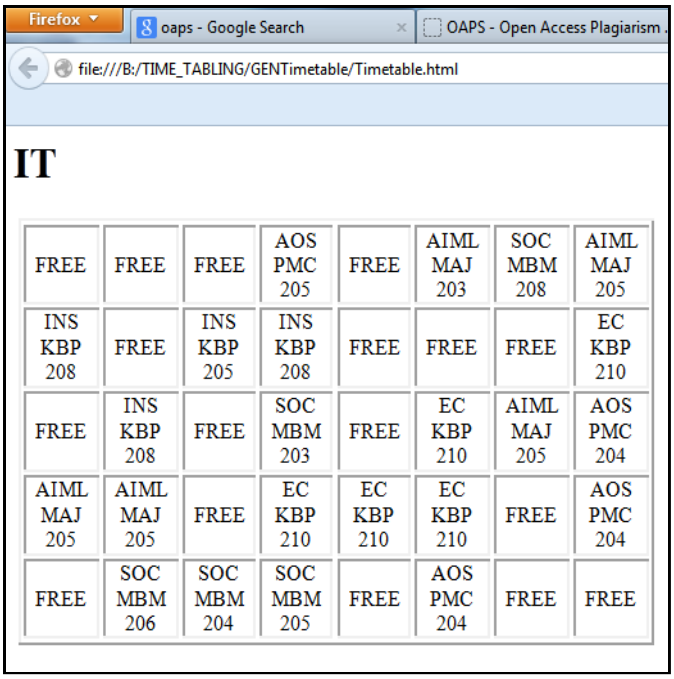
We are taking input from XML file. In XML file data sets are represent as shown in below figure. This XML file is given to XML Handler for generating array list which is responsible to generate population in genetic algorithm [13].Now Genetic Algorithm can perform its operation on population and generate optimized population. This is passing to the HTML generator [12] to generate output in the HTML file.


Applications Places System
Mozilla
File Edit View History Bookmarks Tools Help
file:///home/mvp/NetBeansProjects/GA
Most Visited
Getting Started
Latest Headlines
file:///home/mv...Timetable/1.xml
Done
[Gmail - Inbox - ...
GATimetable - Ne...
GATimetable

```
– <schedule>
    <prof id="1" name="JRM."> </prof>
    <prof id="2" name="DDT."> </prof>
    <prof id="3" name="TMV."> </prof>
    <prof id="4" name="BPV."> </prof>
    <prof id="5" name="DJP."> </prof>
    <prof id="6" name="MPP."> </prof>
    <prof id="7" name="SPV."> </prof>
    <prof id="8" name="BPP."> </prof>
    <prof id="9" name="SNG."> </prof>
    <prof id="10" name="SNP."> </prof>
    <course id="1" name="CP402"> </course>
    <course id="2" name="IT401"> </course>
    <course id="3" name="IT425"> </course>
    <course id="4" name="CP431"> </course>
    <course id="5" name="IT422"> </course>
    <course id="6" name="CP408"> </course>
    <course id="7" name="IT424"> </course>
    <course id="8" name="CP427"> </course>
    <course id="5" name="CP434"> </course>
    <group id="1" name="IT"> </group>
    <group id="2" name="CP"> </group>
```

Fig. 7Input XML file

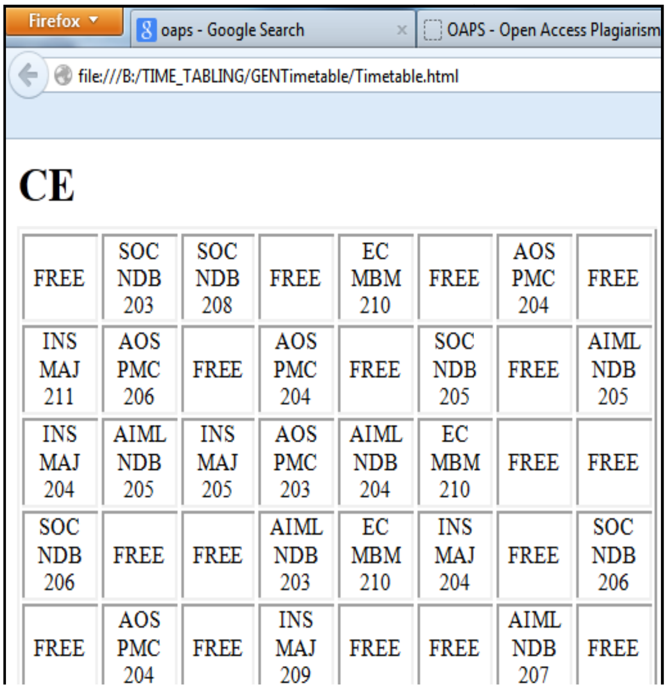
Output:


Firefox
oaps - Google Search
OAPS - Open Access Plagiarism
file:///B:/TIME_TABLING/GENTimetable/Timetable.html

IT

FREE	FREE	FREE	AOS PMC 205	FREE	AIML MAJ 203	SOC MBM 208	AIML MAJ 205
INS KBP 208	FREE	INS KBP 205	INS KBP 208	FREE	FREE	FREE	EC KBP 210
FREE	INS KBP 208	FREE	SOC MBM 203	FREE	EC KBP 210	AIML MAJ 205	AOS PMC 204
AIML MAJ 205	AIML MAJ 205	FREE	EC KBP 210	EC KBP 210	EC KBP 210	FREE	AOS PMC 204
FREE	SOC MBM 206	SOC MBM 204	SOC MBM 205	FREE	AOS PMC 204	FREE	FREE

Fig. 8Generated time table for IT Department


Firefox
oaps - Google Search
OAPS - Open Access Plagiarism
file:///B:/TIME_TABLING/GENTimetable/Timetable.html

CE

FREE	SOC NDB 203	SOC NDB 208	FREE	EC MBM 210	FREE	AOS PMC 204	FREE
INS MAJ 211	AOS PMC 206	FREE	AOS PMC 204	FREE	SOC NDB 205	FREE	AIML NDB 205
INS MAJ 204	AIML NDB 205	INS MAJ 205	AOS PMC 203	AIML NDB 204	EC MBM 210	FREE	FREE
SOC NDB 206	FREE	FREE	AIML NDB 203	EC MBM 210	INS MAJ 204	FREE	SOC NDB 206
FREE	AOS PMC 204	FREE	INS MAJ 209	FREE	FREE	AIML NDB 207	FREE

Fig. 9 Generated time table for CE Department

V. CONCLUSION & FUTURE ENHANCEMENT

Conclusion:

We conclude that using Genetic Algorithm in Scheduling Time-Table, we get fittest and highly optimized time table for large number of population. In Genetic Algorithm we use large number of population and apply genetic operator on the population for checking the constraints.

We use XML in this system because we require easily use and easily editable database. XML is easily editable with changing requirements, and no need to create any special input design form. XML is platform independent, so it will work in all o.s. So using XML we reduce time of implementation and complexity as compare to Sql database.

We use JAVA in this system because it is platform independent and robust language. It is run in any o.s. The connection between JAVA and XML is also easy as compare to JAVA and SQL.

We use HTML in this system because it is also run any o.s. We use HTML as output because it is easy to design as compare to design in java applet.

Finally we conclude that using XML + JAVA + HTML and Genetic Algorithm in this system, we get highly optimize time table in very short time.

5.2- Future Enhancement:

Currently system is limited we can expand this system in following area

- This system will expand for all the department of the institute.
- We can give priority to each faculty.
- According to the priority and faculty requirement we can change faculty availability time duration.
- We can also derive the time table for each individual student according to its subject from this main time table.
- It will save time and reduce human effort.

REFERENCES

[1] Wang, Cheng ; Wang, Shi-bo ; "Research of University Timetable Problem based on genetic algorithm" in Information Science and Engineering (ICISE), 2010

[2] Lukas, S., Aribowo, A. "Genetic algorithm and heuristic search for solving timetable problem case study: Universitas Pelita Harapan timetable" in Applications of Digital Information and Web Technologies, ICADIWT '09, 2009

[3] Ghaemi, S. et. al. "Using a genetic algorithm optimizer tool to solve University timetable scheduling problem" in Signal Processing and Its Applications, 2007. ISSPA 2007.

[4] Sigl, B., Golub, M., "Solving timetable scheduling problem using genetic algorithms" in Information Technology Interfaces, 2003. ITI 2003

[5] Els, R. et. al. "An evolutionary algorithm hyper-heuristic for producing feasible timetables for the curriculum based university course timetabling problem" in Nature and Biologically Inspired Computing (NaBIC), 2010

[6] Wang Xiao Yun; Wang Feng Kun; "Optimize Timetabling Problem Using Improved Genetic Algorithm" in Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008.

[7] K. B. Parmar, H. B. Prajapati, V. K. Dabhi, "Cutting Stock Problem: A Survey of Evolutionary Computing Based Solution,"*Green Computing Communication and Electrical Engineering (ICGCCEE)*, 10.1109/ICGCCEE.2014.6921411, 2014

[8] Yu, E. and Sung, K. (2002), "A genetic algorithm for a university weekly courses timetabling problem" in International Transactions in Operational Research, 2003

[9] Al-Amoush, Farhan R., "Improving Primary School Timetabling Using Genetic Algorithm" (June 13, 2011).

[10] R. Piola, "Evolutionary solutions to a highly constrained combinatorial problem", Dipartimento di informatica, Università degli di Torino, Italy, 1992.

[11] M. Grobner, et al., "A standard framework for timetabling problems," Proc. International Conference on the Practice and Theory of Automated Timetabling 2002, LNCS 2740, pp.24-38, 2003.

[12] H.Rudova, et al., "University course timetabling with soft constraints," Proc. International Conference on the Practice and Theory of Automated Timetabling 2002, LNCS 2740, pp. 310-328, 2003.

[13] Sait, S., Ali, S., and Benten, M.: "Scheduling and allocation in highlevel synthesis using stochastic techniques," Microelectron. J., 1996, 27, pp. 693-712

[14] Mandal, C., Chakrabarti, P., and Ghose, S.: "GABIND: a GA approach to allocation and binding for the high-level synthesis of data paths," IEEE Trans. Very Large Scale (VLSI) Syst., 2000

[15] International Timetabling Competiton accessed March 2005. http://www.idsia.ch/Files/ttcomp2002/

[16] Kostuch P. (2004) "The University Course Timetabling Problem with a 3-stage approach", In E. Burke, M. Trick (Eds.) Practice and Theory of Automated Timetabling - PAT AT V, pp 251-266.

[17] R. Weare, E. Burke, and D. Elliman, "A Hybrid Genetic Algorithm for highly Constrained Timetabling Problems", Department of Computer Science, January 1995.

[18] A. Hertz, "Tabu Search For Large Scale Timetabling Problem", European Journal of Operational Research, Vol. 31, No 1, 1991, pp. 39-47.

[19] O.Mahdi, R.N.Ainon and R. Zainuddin, "Using a Genetic Algorithm Optimizer Tool to Generate Good Quality Timetables," Proc. of the 2003 10th IEEE International Conference on Volume 3, 14-17 Dec. 2003 Page(s):1300 - 1303 Vol.3