

Time Table Generation Using Random Numbers

Prakhar Gupta, Rahul Chaudhary, Arun Kudiyal, Parveen Kumar, Devendra Prasad, Govind Prasad Arya

Abstract-The creation of time table is very time consuming and tedious work. While creating a time table the important point which we have to keep in mind is to manage clashing between the teachers. So present research has been conducted in which we have constructed a simple time table, which ensures no clashing of teachers takes place.

Index Terms- Java, Java Program for time table generation, time table generator

1 INTRODUCTION

The class timetabling problem is a typical scheduling problem that appears to be a tedious job in every academic institute once or twice a year. The time table creation is very time consuming and we also have to keep in mind all the constraints related to the time table. In earlier days, time table scheduling was done manually with a single person or some group involved in task of scheduling it manually, which takes a lot of effort and time. Timetabling is the task of creating a timetable while satisfying some constraints. There are basically two types of constraints, soft constraints and hard constraints. Soft constraints are those if we violate them in scheduling, the output is still valid, but hard constraints are those which if we violate them, and then the time table is not valid. But this time table takes the very important constraint that is the clashing between the teachers don't take place. And keeping in mind this major constraint, the time table is generated. The time table generated using the algorithm given in this research paper is free from clashing. The algorithm is based on the use of random numbers.

2 EXISTING METHODOLOGY

- Prakhar Gupta is currently pursuing bachelor degree program in Computer Science & Engineering from Shivalik College of Engineering, Dehradun (Affiliated to Uttarakhand Technical University), Uttarakhand, India.
- Rahul Chaudhary is currently pursuing bachelor degree program in Computer Science & Engineering from Shivalik College of Engineering, Dehradun (Affiliated to Uttarakhand Technical University), Uttarakhand, India.
- Arun Kudiyal is currently pursuing bachelor degree program in Computer Science & Engineering from Shivalik College of Engineering, Dehradun (Affiliated to Uttarakhand Technical University), Uttarakhand, India.
- Parveen Kumar is currently working as an Assistant Professor in the Department of Computer Science & Engineering of Shivalik College of Engineering, Dehradun, Uttarakhand.
- Devendra Prasad is currently working as an Assistant Professor in the Department of Computer Science & Engineering of Shivalik College of Engineering, Dehradun, Uttarakhand.
- Govind Prasad Arya is currently working as an Assistant Professor in the Department of Computer Science & Engineering of Shivalik College of Engineering, Dehradun, Uttarakhand. E-mail: govind.arya10@gmail.com

The existing approach to the generation of time table is the use of Genetic Algorithms. Genetic Algorithms are general search and optimization algorithms inspired by processes and normally associated with natural world. Genetic algorithm mimics the process of natural selection and can be used as a technique for solving complex optimization problems which have large spaces. Genetic algorithms maintain many individual solutions in the form of population. Individuals (parents) are chosen from the population and are then mated to form a new individual (child). The child is further mutated to introduce diversity into the population.

3 PROPOSED METHODOLOGY

The methodology used in generating the time table is the use of random numbers. The random numbers are uniquely assigned to each faculty name and then we make a table such that rows represent the Days (Monday, Tuesday and so on) and the column represent the no. of periods, and fill each row with the distinct random numbers (ranging from 0 to no. of faculty-1). In this way in every row (i.e. every day) there is distinct random numbers and means this means there are different teachers in each period which ensures no clashing.

The algorithm is given below-

- Step 1: START
- Step 2: Declare 2-D Array A[6][7] of integer type and fac_name[7] of string type.
- Step 3: Declare variable nos, i, j of integer type
- Step 4: PRINT "Enter number of subjects".
- Step 5: SCAN in nos.
- Step 6: PRINT "Enter name of faculty with assigned subject in the format name_of_faculty(Subject name) "
- Step 7: FOR i=0 to i=nos-1.
- Step 8: SCAN in fac_name[i].
- Step 9: FOR i=0 to i=5
- Step 10: FOR j=0 to j=nos-1
- Step 11: Assign random numbers between 0 to nos-1 in

a[i][j] without repetition.

Step 12: Display the time table as follows:

Step 13: FOR i=0 to i=5.

Step 14: PRINT different days each time (i.e. Monday, Tuesday and so on).

Step 15: FOR j=0 to j=nos-1

Step 16: PRINT fac_name[a[i][j]].

Step 17: PRINT new line

Step 18: STOP

Array A[6][7]

5	2	0	1	4	3	
3	0	4	5	2	1	
1	5	2	4	3	0	
1	3	5	0	4	2	
4	5	2	3	0	1	
2	0	3	5	1	4	

The time table generated using this algorithm is only for one class. If we want to generate time table for another class such that some faculties are teaching both the classes then we assign same random no. to the same faculty member in each time table and then check that number for not coming simultaneously in each cell of both the tables. In this way we can generate no. of time tables for different classes without clashing between the teachers.

4 FUTURE SCOPE

The time table generation is a major issue in colleges & Institutes where large no. of courses are running in present as well as in future time. We can add more constraints to this algorithm of time table to make it a full-fledged time table generator without any errors.

5 CONCLUSION

The time table generated using this algorithm is not taking much of the constraints. But in future we can add more constraints to the algorithm to make a reliable and error free time table.

6 REFERENCES

- [1] DilipDatta, Kalyanmoy Deb, Carlos M. Fonseca, "Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm".*KanGAL* 2005.
- [2] M. Doulaty, M. R. FeiziDerakhshi, and M. Abdi, "Timetabling: A State-of-the-Art Evolutionary Approach", *International Journal of Machine Learning and Computing*, Vol. 3, No. 3, June 2013.
- [3] D. Abramson, J. Abela, "A Parallel Genetic Algorithm for Solving the School Timetabling Problem", 15 Australian Computer Science Conference, Hobart, Feb 1992.
- [4] <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.468.6391>
- [5] http://www.academia.edu/3370642/Solving_class_timetabling_problem_of_IIT_Kanpur_using_multi-objectiv...
- [6] <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.103.8703>
- [7] <http://nairaproject.com/projects/2293.ht...>
- [8] <http://nairaproject.com/projects/2293.ht...>
- [9] <https://tensix.com/2016/06/why-you-shoul...>
- [10] <https://el-tramo.be/lcg-random/>