International Journal of Advance Research in Science and Engineering
Volume No.07, Special Issue No.04, March 2018
www.ijarse.com
IJARSE
ISSN: 2319-8354

Datesheet Generation Algorithm for Practical Examinations of JKSBOTE

Shakeel Ahmad Dar

National Institute of Technology, Srinagar

ABSTRACT

J&K State Board of Technical Education is entrusted with the responsibility for conduction of examination and result processing of all the polytechnics of Jammu and Kashmir besides certain other tasks. The examination conduction process is done in two stages, Theory Examination and Practical Examination. For Smooth Conduction of Practical Examination, an automated Date Sheet Generation Process has been done in this study to ensure minimum time required for conduction of examination besides ensuring the date sheet clash free i.e. no student is having two exams at a single time slot (Meeting).

Keywords: Optimization Algorithm, Minimum Time Slot Algorithm, Datasheet Generation, J&K State Board of Technical Education (JKSBOTE).

I INTRODUCTION

Date Sheet Development is an important and time consuming process in the Jammu and Kashmir State Board of Technical Education. The Examination Process in the Polytechnics of J&K are conducted two times per year (i.e the Semester System is in Place). In each session of the year, around thousands of students appear in the examination and there are a number of constraints that are to be considered while framing the practical datasheet. The important constraints that define how the date sheet is framed include:

1. Number of Examination Centers
2. Capacity/Infrastructure of each examination center
3. Number of Students in each Branch
4. No of Subjects in each Branch
5. No of faculty members available in each Branch.
6. Feasibility of Students/Faculty in appearing at a particular examination center.

The primary aim of this project is to minimize the number of days for the conduction of examination and no student having more than one exam at a single time besides no faculty member engaged for duty at more than one place in a single time slot. The Number of days are minimised in order to give more students more time for academics and time saved is time earned. Manually it used to take weeks for the generation of optimal datesheet and would often had clashes in students having multiple exams at same time. In this part of the project , the date

sheet so generated is clash free among students as well as faculty members besides generates the datesheet in the minimum possible number of days.

II.ALGORITHM AND ITS EXPLANATION

The Practical Date Sheet Algorithm is based on few assumptions. The assumptions are as:

Assumption 1: Each Day is Divided into two Time Slots only,viz 1st Meeting(1M) and 2nd Meeting (2M). For More Slots , the Algorithm needs minor modifications for the effective implementation.

Assumption 2: Time Period for conduction of one Practical Exam is one Time Slot i.e for a particular student for any subject, the exam can be conducted just in one time slot(1M or 2M).

Assumption 3: Time Period for Conduction of Practical Exam for all students in a given subgroup is one time slot.(The Max Size that a subgroup can have is limited to 30)

The Date Sheet Generation Process starts first with the selection of Master Group and subsquent Creation of Sub Groups. A Master Group is selected amongst the various institutes for a particular branch (Programme) and then based upon the number of students, the number of subgroups are created.

The Sub Groups are Put in Sub Group Pairs to generate the list in the two slots in each day. If the Number of Time Slots are increased then accordingly that number of groups shall be clubbed together. The Subjects for a particular branch in which the students have to appear are also clubbed together in Subject Pairs.

```
        $PracSubQry="SELECT S.SubjectID from Subject**********//SubjectQuery"
$c=0;
        //echo $PracSubQry;
            $ResultPracSubQry=sqlsrv_query($conn, $PracSubQry);
            $Pair=0;
            $i=0;
            $SubPairCount=0;
while ($PracSubQryrow=sqlsrv_fetch_array($ResultPracSubQry))
{


        $SubjectPair[$i][$Pair]=$PracSubQryrow['SubjectID'];

    // echo $PracSubQryrow['SubjectID']; echo"</br>";
     echo "<input  type='hidden' name='Subject[]' value='".$PracSubQryrow['SubjectID']."'  >";


     if($Pair==1)
          {
               $i=$i+1;
               $Pair=0;
          }
     else
          {
               $Pair=1;
               $SubPairCount++;
          }

     echo "<th colspan='2'>";
     echo $PracSubQryrow['SubjectName'];


     echo "</th>";
     $c++;


}
```

Fig-1: Creation of Subject Pairs.

```
  $SubGrpQry="SELECT SubGroupID from SubGroup where MGroupID='$GrpID' order by SubGroupID ASC ";
$ResultSubGrpQry=sqlsrv_query($conn, $SubGrpQry);
//echo $SubGrpQry;
            $Pair=0;
            $i=0;
            $GrpPairCount=0;
    while ($SubGrpQryRow=sqlsrv_fetch_array($ResultSubGrpQry))
      {


            $GroupPair[$i][$Pair]=$SubGrpQryRow['SubGroupID'];

            if($Pair==1)
                  {
                        $i=$i+1;
                        $Pair=0;
                  }
            else
             {
                   $Pair=1;
                   $GrpPairCount++;
             }

     }
     //////print_r($GroupPair);
            ?>
```

Fig.2: Creation of Sub Group Pairs.

Among the two values *The Number of Group Pairs* or *The Number of Subject Pairs* , the maximum value is taken as the number of Minimum days that the practical exam conduction can take. As for each subject Pair , the minimum number of days it would take for the conduction of examination will.

In case the number of subject Pairs or Group Pairs is odd, the last element of the last pair among the subject Pairs or Group Pairs as the case may be is kept as blank/Null Value.

For Every Day , the loop is run. In each Day Number , another loop is run for each subject Pair.

For each day and for each Subject Pair, a value Termed here as Group Pair Index is calculated. This value is very important keeping in view its utility for the generation. The value of this variable gives the index of the Group Pair that has to be fitted in the slot for a particular exam Day in a particular subject pair. In the first day, the value of this variable is in sync with the subject pair index. For the next day , the index is taken one ahead as the exam of 1st group for that subject has been already scheduled for day one and accordingly the values are taken for consequent days and subjects.

For each day and for each subject Pair , four variables are stored for exam Day.

i) The First Element stores the First Group of the Group Pair to be used in this part in the 1st time slot of 1st Subject

ii) The Second Element Stores the Second Group in the 2nd Time Slot of the 1st Subject.

iii) The Third Element Stores the Second Group in the 1st Time Slot of 2nd Subject.

iv) The Fourth Element Stores the First Group in the 2nd Time Slot of the 2nd Subject.

The Algorithm for date sheet generation is as under:

1. *Initilization of SubjectPairs and SubjectPairCount*
2. *Initilization of GroupPairs and GroupPairCount*
3. *Get MinimumDaysCount =Maximum(SubjectPairCount , GroupPairCount).*
4. *for (int i=0;i<MinimumDaysCount;i++)*

```
{
    DayNumber=i+1;
      for(int g=0;g<SubjectPairCount;g++)
        {
          //GroupPair Index is used to put the particular subject in the time slot and for every new day , the GroupPairs are selected with one shift to ensure clashfree generation of algo.
    GroupPairIndex=(d+g)%MinimumDaysCount; /// this line shifts the group pairs

          SubjectElement1=SubjectPair[g][0]
          SubjectELement2=SubjectPair[g][1]
          GroupElement1=GroupPair[GroupPairIndex][0]
          GroupElement2=GroupPair[GroupPairIndex][1]

    // Ist Time Slot of Ist Subject in Pair
    ExamDay[DayNumber][SubjectElement1][0]= GroupElement1

    // 2nd time Slot of 1st Subject in Pair
    ExamDay[DayNumber][SubjectElement1][1]= GroupElement2

    // 1st Time Slot of 2nd  Subject in Pair
    ExamDay[DayNumber][SubjectElement2][0]= GroupElement2

    // 2nd Time Slot of 2nd  Subject in Pair
    ExamDay[DayNumber][SubjectElement2][1]= GroupElement1

    }
}
```

//At the end of the outer for loop , the ExamDay will contain the optimal solution for the DateSheet.

5. *Display ExamDay Array.*

III CONCLUSION AND RESULTS:

The Algorithm has been implemented in php and has been deployed for the Examination Session of Nov-December 2017 in the Jammu and Kashmir State Board of Technical Education and it has resulted in an efficient development of datesheet with 100% accuracy in terms of clashes for the examination of a student besides the time required in the prepartaion of the datesheet has reduced from weeks to few mins.

A Snapshot of the datesheet so developed with some formatting is shown as Fig.3.

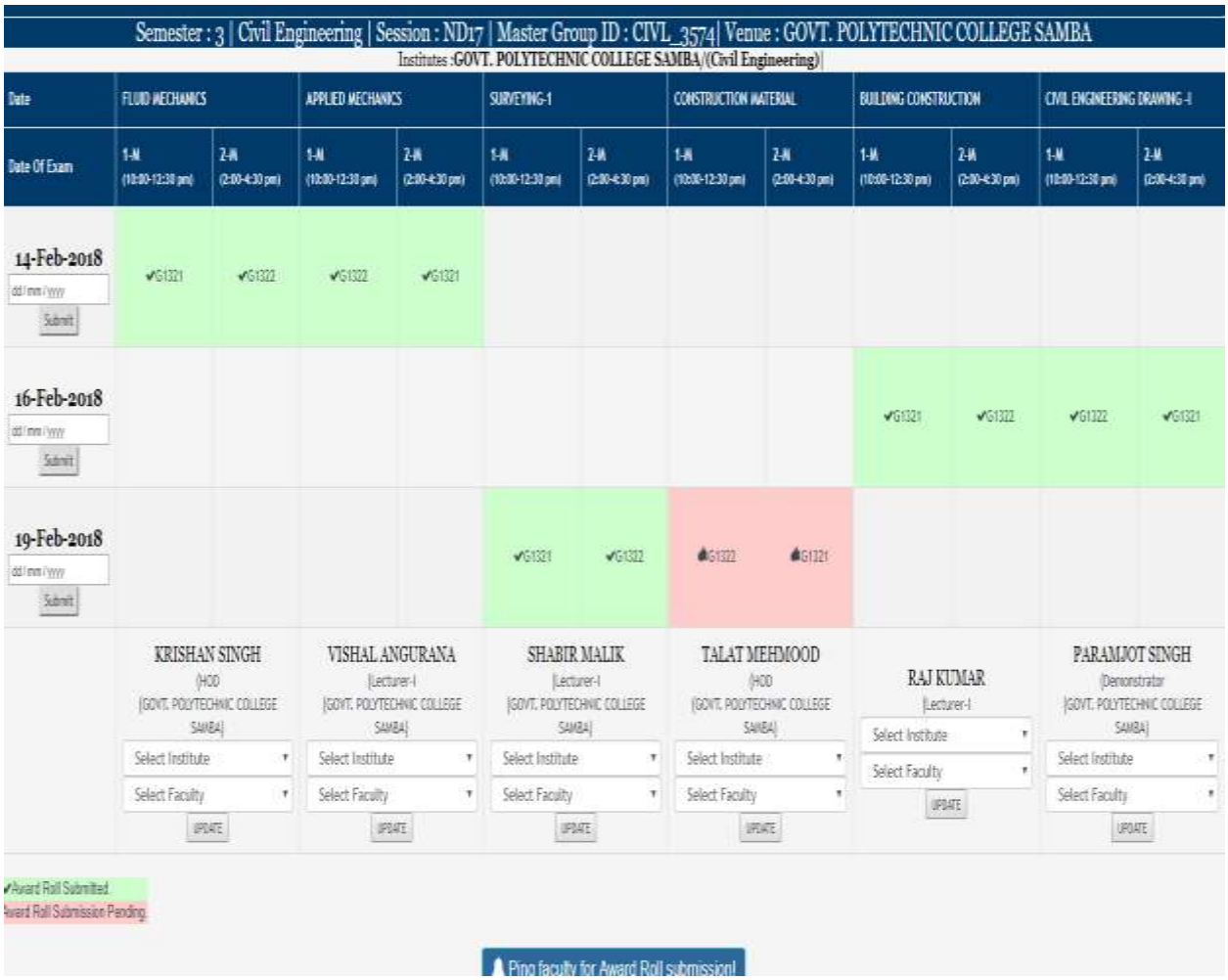

Fig.3: Generated Date Sheet.

REFERENCES

[1]A. V. AHO, J. E. HOPCROFT AND J. D. ULLMAN, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] P. A. BLONIARZ, M. J. FISHER AND A. R. MEYER, A note on the average time to compute transitive closures, Proc. of the 3rd Int. Colloquium on Automata, Languages and Programming (July 1976), S. Michelson and R. Milner, eds.

[3] P. ERDOS AND J. SPENCER, Probabilistic Methods in Combinatorics, Academic Press, New York, 1974.

[4] A. ITA! AND M. RODEH, Some matching problems, Proc. of the 4th Int. Colloquium on Automata, Languages and Programming (July 1977), A. Salomaa, ed.

[5] P. ERDt3S, Graph theory and probability II, Canad. J. Math., 13 (1961), pp. 346-352.

[6] F. HARARY, Graph Theory, Addison-Wesley, Reading, MA, 1969.

[7] L. LovAsz, On chromatic number offinite set systems, Acta Math. Acad. Sci. Hungar., 19 (1968),.pp. 59-67.

[8] C. L. LIU, Introduction to Combinatorial Mathematics, McGraw-Hill, New York, 1968. [10] V. STRASSEN, Gaussian elimination is not optimal, Numer. Math., 13, pp. 354-356.