School of Science and Math Exam Scheduler (SSMES) Help File*

Spring 2019

Contents

*https://github.com/maxhirsch/ssmes

1 Introduction

The School of Science and Math Exam Scheduler (SSMES) is the result of a 2019 Mini-Term project at the North Carolina School of Science and Mathematics which aimed to reduce the number of exam conflicts at the school. Students involved in the Mini-Term were Daniel Carter, Edward Feng, Kathleen Hablutzel, and Max Hirsch. This documentation serves as a help guide in running the interface for the code which minimizes exam conflicts.

2 About the Minimization Algorithm

Below we outline the general algorithm for determining low-conflict exam schedules. It by no means produces the lowest-conflict schedule, but it maintains similar structure to previous exam schedules while reducing conflicts. The algorithm is more precisely described in the code on Github. In the algorithm, a *conflict* is when a student has more than one exam in the same exam block. Previously, NCSSM had three exams each day during exam week, allowing another type of conflict in which a student had three exams in a 24 hour period. Our solution was to remove one exam period from each day so that students could no longer have this conflict.

Algorithm 1 Creating low-conflict exam schedule

Result: Exam blocks with relatively few conflicts

```
procedure REDUCE CONFLICTS
    for n trials do
        Split departments
        Randomly assign departments to exam blocks
        do
            Swap departments between exam blocks if the swap reduces conflicts
        while conflicts reduced
    end for
end procedure
```

Algorithm 2 Splitting Departments

Result: Departments which may or may not be split into two

```
procedure SPLIT DEPARTMENTS(definite_split, restricted_split)
    for n trials do
        Split definite_split departments
        Split any non restricted_split departments if there are courses with no intra-department conflicts
    end for
end procedure
```

3 Available Settings

In this section, we describe each of the settings in SSMES.

3.1 Term Schedule

Under *Settings* → *Term Schedule*, the school schedule can be toggled from trimester to semester.

3.2 Total Random Trials

Under *Settings* → *Total Random Trials*, the number of random initializations of schedules can be set. The default value is 5. Larger values of this setting will result in fewer conflicts, but the minimization will take a longer time.

3.3 Random Seed

Under *Settings* → *Random Seed*, the random seed for Total Random Trials can be set. The default value is 42.

4 Entry Fields

In this section, we describe each of the fields on the main screen of SSMES.

4.1 Term

Term is the trimester (T1, T2, or T3) or semester (S1 or S2) for which you would like to create an exam schedule. The term must chosen in the list before any data can be loaded.

4.2 Data File

The data file field is where you input the name/location of the file which contains course enrollment data. You can browse for this data by clicking "Choose File" beside the entry field. The data file must follow a specific format which we outline below.

4.2.1 Data File Format

The data file with course enrollment data is a .csv file formatted in the following way:

StudentID,Trimester,Course1,Course2,...
1,1,0,1,...
2,1,0,0,...
3,1,1,0,...

If on a semester schedule, "Semester" would appear in the header line rather than "Trimester."

Course1, Course2,... represent the course numbers with two alphabetic characters followed by three numeric characters (e.g. CH401). Each row is the student ID followed by the term (1, 2, or 3 if a Trimester schedule; 1 or 2 if Semester), and a list of 1s and 0s according to whether or not respectively a student is taking that column's course. An example file can be found on the project Github.

4.3 Number of Exam Days

Number of Exam Days is the number of days on which exams will be held. If, for example, it is 4, then there will be $2 * 4 = 8$ exam blocks.

4.4 Courses and Excluded Courses

The Courses field will be populated when you load data in the Data File field. You can then exclude courses from the minimization (e.g. classes without exams or with final papers instead of exams) by clicking the course then clicking "To Excluded." You can move an excluded course back to the Courses field by clicking on the course in the Excluded Courses field and clicking "From Excluded." Once data is loaded from the Data File field, you can also load excluded courses into the Excluded Courses field by choosing an Excluded Courses file. This can be done with the Excluded Courses File field which is found below the Excluded Courses field. This file must be formatted in a particular way which we describe below.

4.4.1 Excluded Courses File

The Excluded Courses file should be formatted such that each excluded course is on its own line with no other characters surrounding it. For example:

MA470
RE120
MA472

and so on.

4.5 Departments

The Departments field will be populated when data is loaded in the Data File field. Departments can be split in the algorithm. We allow the user to explicitly split or not split departments by moving departments to the Definite Split Departments and Restricted Split Departments fields respectively in a manner similar to that of moving courses to excluded courses described in subsection 4.4. The splitting of departments can also be inferred according to the rule described in algorithm 2. This inference can be enabled by checking the Infer Splits checkbox below the Restricted Split Departments field.

5 Output

The result of the optimization is two files. The user will provide a filename, *fn*, and a popup will allow the user to select a folder to which the files should be saved. The files will be saved as *fn*_department_courses.json and *fn*_exam_blocks.json. *fn*_department_courses.json contains the departments and their associated courses (a split department will end with a suffix to denote this). *fn*_exam_blocks.json will contain each exam block with the departments in that exam block. Note that there is no actual ordering to the exam blocks, as the order of the blocks themselves will not affect the total number of conflicts.