ETM
ENGINEERING & TECHNOLOGY MANAGEMENT

# *Title: Student Scheduling at a Charter School*

Course Title: Operations Research
Course Number: ETM 540
Instructor: Dr. Tim Anderson
Term: Fall
Year: 2013
Author(s): Fannie Black, David Tucker, Chih-Jen Yu, and Tim Calderwood

# Table of Contents

## INTRODUCTION

A local charter school in Portland, Oregon, is experiencing staffing issues relative to teacher scheduling while maintaining optimal classroom size. The student population at the charter school has been growing over the last few years, and with increased classroom size there is a need to optimize scheduling to minimize the cost to operating the school.

The charter school was informally founded by parent volunteers in 1974 and became a 501(c)(3) in 1994. Its mission is "To enhance the diverse strengths of our youth and families in partnership with the community through cultural identity and education." Operating as an academic institution, the charter school additionally offers a variety of social services outside of education to the community. These services support topics of domestic violence, housing, and employment, to name a few.

The key problem that this work is to address is that the school is experiencing staff scheduling while maintaining an optimal classroom size that supports the mission of the organization. The problem is compounded due to the increase in student population which has been expanding in recent years.

The current scheduling tool is done manually and is labor intensive to implement. There is a lot of trial and error when creating the master class schedule, as well as teacher and room availability. The charter school aims to balance out the type and level of a class when building the schedule. Individual student schedules are created based on their needs. A student advisor reviews their transcripts to determine what credits they have completed, and creates a schedule based on their needs to obtain a high school diploma. If a student has goals to pursue higher education, those are taken into consideration as well when creating their schedule. This individualized approach to scheduling students into classes is one of the reasons why existing class scheduling programs used by the local school district does not work for the charter school.

The research objectives of this paper looks to develop a model to assist with teacher and student scheduling. The model developed and research results are expected to act as a pilot program for the charter school and could be implemented at other charter schools as well.

## LITERATURE REVIEW

Based on the research objectives, this project is aimed at handling teacher and student scheduling problems. Literature review was conducted to discuss relevant scheduling and timetabling papers with a focus on student scheduling which mainly deal with assigning students to proper course or class in accordance with graduation requirements and resource constraints.

For scheduling problems, Dantzig's model is considered as the first paper using integer programming with set covering model. The objective of his model is to minimize the cost of the scheduled shifts with two restrictions. The first constraint is that employees assigned should be greater than or equal to the desired staffing level. The second constraint specifies the numbers of employees to be a non-negative integer. The formulation is briefly described as follows: [1]

Min Z = $\sum_{t \in T} C_t X_t$

Subject to $\sum_{t \in T} a_{tp} X_t \geq r_p$ for p ∈ P,
$X_t \geq 0$ and $X_t \in$ integer for t ∈ T,

The variables are defined as follows:
P: the set of indices associated with planning periods,
T: the set of indices associated with allowable shifts,
$a_{tp} = 1$, if period p is a working period for shift t,
$a_{tp} = 0$, otherwise,
$c_t$ : The cost of having an employee work shift t,
$r_p$ : The desired staffing level in period p,
$X_t$ : The number of employees working shift t

Based on Dantzig's model, some of the literature indicated that various integer programming models can be used in retail stores, call centers, and university timetabling problems, which are briefly listed below.

1. Retail Store Case (Kanna Miwa & Soemon Takakuwa 2010 ):[2]
   Problem: To find optimal number of clerks based on operations types, operation frequency, and staffing schedule
   Objective: Minimize daily total working time
   Constraints: Expected number of operations at each time interval
   Model: Integer programming

2. Call Center Case (Katherine Marie Perry 2012):[3]
   Problem: Finding the optimal solution for the call-center scheduling problem
   Objective: (1) Minimize the sum of staff scheduled above demand for each time period (2) Minimize staff size
   Constraints: Staff for each time period greater than or equal to the forecast
   Model: Integer programming (set covering)

3. University Timetabling Case (S. Daskalaki 2004): [4]
   Problem: Building a university timetable satisfying university features and requirements
   Objective: The objective function is specified to minimize to the cost of assigning course to the period of the day
   Constraints: Uniqueness, completeness, consecutiveness, repetitiveness, pre-assignment
   Model: Integer programming

There are other student scheduling articles showing that assignment model with binary decision variables can be used to optimize the scheduling requirement. For instance, an undergraduate student case needs to be assigned in management related courses. In order to find an optimal class schedule, the objective function can be established to maximize the total rating from the courses in the schedule. The constraints include each course to be taken only once, taking one class during any time slot, choosing class based on preferences, avoiding some particular courses, taking one of some course in the same subject, and taking exactly the required courses. The illustrated objective formulation is briefly shown below. [5]

$$X_{ij} = \begin{cases} 1, \text{ if course j is taken at time slot i} \\ 0, \text{ else} \end{cases} \quad \text{where } i = \{1, 2 \ldots 10\}, j = \{1, 2 \ldots .8\}$$

Maximize $Z = \sum_{i=1}^{10} \sum_{j=1}^{8} c_{ij}\, x_{ij}$ where $c_{ij}$ = rating of the course j at time slot i

The above literature review indicates that integer programming has been used to solve scheduling and timetabling problems with focus on minimization of numbers of staff, time, and cost variables. The assignment model with binary decision variables can be used to deal with student scheduling problems by maximizing the rating of the courses and considering school regulation, personal preference, and logical constraints.

## OBJECTIVE

The objective for the student scheduling model is to complete a schedule that fulfills graduation requirements for the student, prioritizing required courses ensuring that students can graduate.

## CONSTRAINTS

The constraints that were utilized in the student scheduling model are as follows:

**Class Capacity Constraint** – The number of students registered for the class cannot be greater than the capacity of the classroom where the class is to be located.

**Schedule Constraint** – Students can only be scheduled for one class per hour.

**Requirement Constraint** – Students must take all courses required for graduation.

**Binary Constraint** – All variable cells are to be binary, where a 1 represents that the course is scheduled and 0 represents that the course has not be scheduled.

Each of the above constraints will be included in the scheduling model.

## PRE-SCHEDULE DATA

In order for the model to work correctly it is required that the following information be documented and contained in a spreadsheet.

**Course to Room Assignment** - The room where the class will be held, identifying the location and capacity of each room. This assignment guarantees that facility will have enough desks and the proper capabilities to meet the requirements for the classroom.

**Student Identification** – The number of students must be created for each of the students to be scheduled for each classroom. This identification assigns the students a unique identifier, enabling other data to be linked back to the student.

**Student Academic History** – The courses that the student has completed must be documented, providing the model the previous courses completed. The following table identifies an example of the table. The courses that have been completed would be marked with a 1.

**Table 1: Historical Student Information**

| Historical Information | | | | | |
|---|---|---|---|---|---|
| Courses Completed | English | Math | Science | Social Studies | Elective |
| Student 1 | 1 | 0 | 0 | 0 | 0 |
| Student 2 | 1 | 0 | 1 | 0 | 0 |
| Student 3 | 0 | 1 | 0 | 1 | 0 |
| Student 4 | 1 | 1 | 0 | 1 | 0 |
| Student 5 | 0 | 1 | 0 | 1 | 0 |
| Student 6 | 0 | 1 | 1 | 0 | 1 |
| Student 7 | 1 | 1 | 0 | 0 | 1 |
| Student 8 | 0 | 0 | 0 | 0 | 1 |

**Graduation Requirements** – Identification of the courses that are needed for graduation and the priority of classes to be completed.

For the charter school, each student is required to complete a total of 24 courses for graduation, the course list is identified in the table below. With the completion of the 24 credits, students are required to meet the school district's proficiency for Oregon's essential skill (ES) standards for reading, writing, and math. Students are also required to develop a personal education plan to help guide them in pursuing their personal, academic, and career interests after graduation; participate in a career related learning experience connecting their learning beyond the classroom; demonstrate their career-related knowledge and skills (e.g., problem solving, personal management, teamwork, employment foundations, communication, and career development). Additionally, students are to complete an extended application of academic and specialized knowledge and skills within the student's personal and career interests after graduation.

**Table 2: Graduation Requirements**

| Credit Area | Credits Option 1 | Credits Option 2 |
|---|---|---|
| English | 4 | 4 |
| Math | 3 | 3 |
| Science | 3 | 3 |
| [i]Social Studies | 3 | 3 |
| Physical Education | 1 | 1 |
| Health | 1 | 1 |
| World Language | 3 | 2 |
| Career & Technical Education | 0 | 1 |

| | | |
|---|---|---|
| Electives | 6 | 6 |
| **TOTAL CREDITS** | **24** | **24** |

[i]Social Studies credit requirements are as follows: Government 0.5; Economics 0.5; US History 1.0; World History/Global Studies 1.0.

For the graduation requirements to be communicated to the model, each course would need its own unique identifier assigned to the course. The following table illustrates this expansion.

**Table 3: Expanded Graduation Requirements**

| Credit Area | Credits |
|---|---|
| English 1 | 1 |
| English 2 | 1 |
| English 3 | 1 |
| English 4 | 1 |
| Math 1 | 1 |
| Math 2 | 1 |
| Math 3 | 1 |
| Science 1 | 1 |
| Science 2 | 1 |
| Science 3 | 1 |
| Social Studies – Government and Economics | 1 |
| Social Studies – US History | 1 |
| Social Studies – World History | 1 |
| Physical Education | 1 |
| Health | 1 |
| World Language 1 | 1 |
| World Language 2 | 1 |
| Career & Technical Education | 1 |
| Elective 1 | 1 |
| Elective 2 | 1 |
| Elective 3 | 1 |
| Elective 4 | 1 |
| Elective 4 | 1 |
| Elective 6 | 1 |
| **TOTAL CREDITS** | **24** |

**Course Prioritizations** – Course sequence is upheld by providing weights for each course. The following table identifies how weights are assigned to the courses using example weights. In the model, course weights are required so that the solver utilizes courses that must be completed for graduation requirements over secondary courses and electives.

The charter school prioritizes core classes that students need as well as any support for essential skills (math, reading, and writing). Since students need four years of English, that is usually the highest priority.

**Table 4: Course Weights**

| Course | English | Math | Science | Social Studies | Elective |
|---|---|---|---|---|---|
| Score | 4 | 4 | 3 | 2 | 1 |

## MODEL FORMULATIONS

The following show the model formulations used in the GLPK Code of the section.

**Definitions**

$Classes = List\ of\ class\ names$

$Students = List\ of\ students$

$HistoricalInfo_{s,c} = List\ of\ students\ and\ the\ classes\ they\ took.$

$Period = The\ time\ each\ class\ is\ taught$

$PriorityScore_C = Priority\ of\ each\ class$

$ClassSize_C = Max\ size\ of\ class$

**Variables**

$StudentSchedule_{s,p,c} = Matrix\ of\ student\ , period, and\ classes, binary$

**Constraints**

For each Period and Student, sum up the number of classes, must be 1 or less -

$$\sum_c StudentSchedule_{s,p,c} \ \leq 1 \forall\ s\ as\ Students, p\ as\ Period$$

Students must satisfy all requirements -

$$\sum_p StudentSchedule_{s,p,c} + HistoricalInfo_{s,c} \leq 1\ \forall\ s\ as\ Students, c\ as\ Classes$$

Students in Class must be less than the class size –

$$ClassSize_C - \sum_s StudentSchedule_{s,p,c} \geq 0\ \forall c\ in\ Classes, p\ in\ Period$$

**Objective**

$$\sum_s \sum_p \sum_c (\text{StudentSchedule}_{s,p,c} * PriorityScore_c)$$

The objective of the model is accomplished with prioritizing the courses needed for graduation and by assigning a weight to the course and maximizing the summation of weighted class schedule.

## GLPK AND OUR MODEL

We used Microsoft Excel to set up our initial model to test formulations. Microsoft Excel was a good choice for this because it provides visual feedback for debugging purposes. However, there are several limitations to using excel that need to be mentioned. The first is Microsoft Excel only supports 200 decision variables. This limited the model to 8 students and 5 classes over 4 periods. The school for our model has over 90 students with 7 periods and with 30 different classes. The second limitation was being able to change data without having to rebuild the entire model.

After reviewing several options, we decided to use the GLPK (GNU Linear Programming Kit) package with GMPL. The GLPK is intended for solving large-scale linear programming (LP) problems. Our test shows that using GLPK allows for more than 200 students with 10 periods and 40 classes, more than what our problem requires. We also found it was designed to have separate data files and allows for data import from various sources including ODBC connections. It also supports advanced reporting options as well. Finally, the model can be used in high level programming languages, for example C# or JAVA.

Our model is setup in three sections and an external coma separated data file. What follows is a description of what each section is used for. However, we leave it up to the reader to refer to the GLPK user guide for details on each of the commands used. A complete printout of the GLPK model can be found in appendix **A**.

### SETS, PARAMETERS, AND VARIABLES

In this section we defined the data storage part of the code. This is broken into three subsections. The first is used to define and load data that is dynamic, primarily the student historical data. The hard coded section is data that will not change from run to run for this model, such as class priority or class sizes. For a future feature, these could be loaded as well from a file. Finally, there are the variables for this model. This is a three-dimensional binary matrix that is used to decide which student takes which class during which period. A 1 means a student is scheduled for a class during a certain period. A 0 means not scheduled.

### CONSTRAINTS

We have three constraints listed in this section. Two other constrains, both binary constraints, were applied in the Sets, Parameters, and Variables section described above. The three constraints are set up to match the formulations described in the formulation section. The first given constraint is

> s.t. ScheduleConstraint {p in Period, s in Students}: sum {c in Classes}StudentSchedule[s, p, c] <= 1;

This is read as "subject to all periods and students, sum up all the classes taken by a student for that period". This sum must be less than or equal to one. This is used to make sure a student is not assigned to more than one class per period. It also assigns the results of this to a set called ScheduleConstraint. Please see Appendix A for the full set of constraints.

### OBJECTIVE SECTION

Our objective, as stated before, is to prioritize courses so that students can graduate on time. In case of this model, it reads as follows.

> maximize obj: sum{s in Students, p in Period, c in Classes}(StudentSchedule[s, p, c] * PriorityScore[c]);

The objective is to sum up all the classes taken and multiply the priority score defined for that class.

### DATA

The data section is where values do not change from run to run of our model. In the case of this model, we decided that class sizes and class priority will not change.

### EXTERNAL DATA FILE

The data we were supplied was in the format as follows.

**Table 5: Data File Example**

| Historical Information | | | | | |
|---|---|---|---|---|---|
| Courses Completed | English | Math | Science | Social Studies | Elective |
| Student 1 | 1 | 0 | 0 | 0 | 0 |
| Student 2 | 1 | 0 | 1 | 0 | 0 |
| Student 3 | 0 | 1 | 0 | 1 | 0 |
| Student 4 | 1 | 1 | 0 | 1 | 0 |
| Student 5 | 0 | 1 | 0 | 1 | 0 |
| Student 6 | 0 | 1 | 1 | 0 | 1 |
| Student 7 | 1 | 1 | 0 | 0 | 1 |
| Student 8 | 0 | 0 | 0 | 0 | 1 |

We wrote a C# program to convert the data into a usable form for our model. The converted external data file is a coma delimited file with the first row being the header information, and the rest of the rows are the historical data for each student. Each row of historical data contains the students name, the class name, and either a 0 or 1. A 1 means a student has taken the class before, a 0 means not taken. A student is listed multiple times, once for each class. The set of classes must then be the same for each student for this model. An example of this is seen below.

```
StudentName,Class,Value
Student4,English,1
Student4,Math,1
Student4,Science,0
Student5,English,0
Student5,Math,1
Student5,Science,0
```

The data we were given by the charter school is that there are 30 total classes to be taught over 7 periods a day. We were also told there are about 100 total students. This gives us 21,000 decision variables, which took about eight minutes to solve with our five constraints. Our max objective function value when solved is 3821. The following is the schedule for the five of the students.

| | Period 1 | Period 2 | Period 3 | Period 4 | Period 5 | Period 6 | Period 7 |
|---|---|---|---|---|---|---|---|
| **Student 5** | Computer Apps | Biology | Environ Science | Native American Lit | Geometry | Algebra1 | Global Studies |
| **Student 19** | Global Studies | Algebra1 | Academic Literacy Read | Native American Lit | Geometry | Algebra1 | Global Studies |
| **Student 23** | Algebra1 | No Class | Geometry | Biology | Storytelling | Native American Lit | Computer Apps |
| **Student 42** | Storytelling | Global Studies | Biology | Environ Science | Geometry | Academic Literacy Read | Food Service Nutrition |
| **Student 59** | Academic Literacy Read | Algebra 1 | Computer Apps | Environ Science | Storytelling | Biology | Food Service Nutrition |
| **Student 93** | Geometry | Biology | Global Studies | Native American Lit | Computer Apps | Algebra 1 | Food Service Nutrition |

While attempting to resolve a difference between our Microsoft Excel model and our GLPK model, it was determined that our model has alternate optimal solutions. This may occur when the objective function has the same slope as one its binding constraints. Our model has such a setup. The following shows the results from different solvers, all with the same optimal value. [6]

| From Excel | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

| From GLPK | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| From R | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

## CONCLUSION AND FUTURE WORK

The model as created can help the charter school schedule students for a term. The schedule would need to be exported and documented in the completed term information, and the process repeated. The potential improvements that can be added to the model for improved efficiency are as follows:

**Instructor Staffing** – The model can be expanded to schedule teachers to courses, therefore improving the functionality of the model and increasing the application of the model to practice at the school.

**Room Size Constraints** – Allowing courses to be taught in multiple locations would increase the flexibility of the model. This can be accomplished by utilizing flexible space and furniture allowing rooms to be organized in ways that support multiple subjects and class sizes. [7]

**Completion of Pre-Course Requirements** – The model needs to be expanded to allow the course sequence to be accounted for and pre-course requirements to be included in the sequence. For example, students could only take English 1 and not English 2 concurrently, and cannot take English 2 until English 1 has been completed.

**Agile Workforce** – The utilization of an agile workforce and cross training of workers can improve the flexibility of the schedule if coupled with flexible room configurations that support multiple subjects and configurations. [8]

There are multiple improvements that the use of a scheduling model can bring within a charter school. The use of a model should only be directional for these schools, as receiving direct guidance from school officials supports the schools mission by providing alternative education for at-risk teens.

## APPENDIX A: GLPK MODEL CODE

```
/* Model Sets, Parameters, and Variables */
/* Read the historical information from the CSV file.  Credit for helping with this solution */
/* goes to Heinrich Schuchardt, from his feedback on the Help-glpk@gnu.org board [9] */

set info, dimen 3; # s = student, c is class, b is either 0 or 1
set Classes  := setof{(s,c,b) in info} c;
set Students := setof{(s,c,b) in info} s;

param HistoricalInfo {s in Students, c in Classes}, binary := sum{(s,c,b) in info} b;

table test IN "CSV" "StudentSampleConv.csv": info <- [StudentName,Class,Value];

/* Hard coded data */
set Period;

param PriorityScore {c in Classes};
param ClassSize { c in Classes};

/*----- Model Variables -----*/
var StudentSchedule {s in Students, p in Period, c in Classes }, binary;

/*----- Model Constraints -----*/
/* Students Can only be scheduled for one class per Period */
s.t. ScheduleConstraint {p in Period, s in Students}: sum {c in Classes}StudentSchedule[s, p, c] <= 1;

/* Students in Class must be less than the class size */
s.t. ClassCapacityConstraint{c in Classes, p in Period}: ClassSize[c] - sum{s in Students}StudentSchedule[s,p,c]
>= 0;

 /* Students must satisfy all requirements */
s.t. RequirementsConstraint{s in Students, c in Classes}:  HistoricalInfo[s,c] + sum{p in Period}StudentSchedule[s, p, c] <= 1;

/*----- Model Objective -----*/
/* Maximize the schedule * the Priority Score for the class */
maximize obj: sum{s in Students, p in Period, c in Classes}(StudentSchedule[s, p, c] * PriorityScore[c]);

solve;

display  obj;

printf "\n"  > "test.csv";
printf "Object Value: %s\n\n\n",   obj >> "test.csv";
printf("Student Schedule\n\n\n,") >> "test.csv";
for {p in Period}
{
	for {c in Classes}
	{
	printf "%s,", p >> "test.csv";
	}
}

for {p in Period}
{
   for {c in Classes}
	{
	printf "%s,", c >> "test.csv";
	}
}

for {s in Students}
{
	printf "%s,", s >> "test.csv";
	for {p in Period}
	{
		for {c in Classes}
		{
```

```
            printf " %s,", StudentSchedule[s,p, c] >> "test.csv";
            }
    }
  printf("\n") >> "test.csv";
}

/* Hard coded data */
data;

set Period := One, Two, Three, Four, Five, Six, Seven;

param ClassSize  :=
  AcademicLiteracyRead180              10
  AdvisoryHomeroom                     10
  Algebra1                             10
  Algebra2                             10
  AnatomyPhysiology                    10
  Biology                              10
  ComputerApps                         10
  ConstructionCareerExploration        10
  CreativeArt                          10
  EnvironScience                       10
  ESLConsultation                      10
  FoodServiceNutrition                 10
  ForensicScience                      10
  Geometry                             10
  GlobalStudies                        10
  GovernmentEcon                       10
  Health                               10
  Math180                              10
  MulticulturalLit                     10
  NativeAmericanArt                    10
  NativeAmericanLit                    10
  PE                                   10
  ReadingandWritingforJustice          10
  ReadingSkillsWorkshop                10
  Spanish2                             10
  Storytelling                         10
  TestingSupport                       10
  USHistory                            10
  WritingSkillsWorkshop                10
  Spanish1                             10;

param PriorityScore:=
  AcademicLiteracyRead180               7
  AdvisoryHomeroom                      1
  Algebra1                              7
  Algebra2                              3
  AnatomyPhysiology                     3
  Biology                               8
  ComputerApps                          8
  ConstructionCareerExploration         4
  CreativeArt                           3
  EnvironScience                        7
  ESLConsultation                       2
  FoodServiceNutrition                  7
  ForensicScience                       5
  Geometry                              6
  GlobalStudies                         9
  GovernmentEcon                        5
  Health                                2
  Math180                               5
  MulticulturalLit                      3
  NativeAmericanArt                     4
  NativeAmericanLit                     6
  PE                                    2
  ReadingandWritingforJustice           3
  ReadingSkillsWorkshop                 4
  Spanish2                              3
  Storytelling                          9
```

```
  TestingSupport                      1
  USHistory                           2
  WritingSkillsWorkshop               1
  Spanish1                            5;
end;
```

## REFERENCES


[1] Gabor Dano, “Workforce scheduling with logical constraints: theory and applications in call centers”, thesis, Department of Mathematics Vrije Universiteit, Amsterdam July, p6, 2004.

[2] Kanna Miwa, Soemon Takakuwa, Optimization and Analysis of Staffing Problems at a Retail Store, Proceedings of the 2010 Winter Simulation Conference, IEEE, pp.1915-1916, 2010.

[3] Katherine Marie Perry, The Call Center Scheduling Problem using Spreadsheet Optimization and VBA, master thesis in at Virginia Commonwealth University, pp. 31-33, April 27, 2012.

[4] S. Daskalaki, T. Birbas, E. Housos, An integer programming formulation for a case study in university timetabling, European Journal of Operational Research, pp. 117–135, 2004.

[5] Janice K. Winch, Jack Yurkiewicz, “Student Class Scheduling With Linear Programming”, p.7, http://www.nedsi.org/proc/2013/proc/p121023002.pdf

[6] Dong-Joon Lim, (dongjoon@pdx.edu), "Re: ETM 540 - GLPK Constraint Confusion," E-mail to T. Calderwood(ctimothy@pdx.edu) 10 Dec 2013.

[7] Kim Hassell. "Flexible Classroom Furniture". American School & University. 84 (2): pp. 18-21, 2011.

[8] Hopp, Wallace, and Mark Oyen. "Agile workforce evaluation: a framework for cross-training and coordination". IIE Transactions. 36 (10): pp. 919-940, 2004.

[9] Heinrich Schuchardt, " Re: [Help-glpk] Generate Set from Param data," Dec. 4 2013.