Scheduling College Classes Using Operations Research Techniques

Barry E. King, Butler University, Indianapolis, IN
Terri Friel, Roosevelt University, Chicago, IL

ABSTRACT

This paper outlines efforts taken at Butler University's College of Business Administration to construct semester class schedules using a mixed integer linear programming problem to develop a schedule of classes and an assignment procedure to assign faculty to the schedule. It also discusses our experiences with the effort and suggestions for improvement.

INTRODUCTION

At Butler University, multiple sections of classes are scheduled in an attempt to give students a wide choice of options in constructing their individual class schedule. Many constraints are imposed such as not having senior level classes on Fridays so seniors can better participate in internship activities and not having junior and senior classes in the same discipline offered at the same day and time. There is also desire from administration to spread the classes out during the day to avoid an unusual number of classes being offered during the prime late morning and early afternoon hours thus better using classroom facilities.

BACKGROUND

The literature is full of course scheduling research. An early work by Shih and Sullivan [4] discusses course scheduling via zero-one programming but the work was hampered by the lack of adequate solvers of the time. Mooney, Rardin, and Parmenter [3] discuss large-scale classroom scheduling at Purdue University. Dinkel, Mote, and Venkataramanan [1] and Fizzano, and Swanson [2] discuss heuristics for classroom and course scheduling. Wang [5] uses a genetic algorithm to solve a course scheduling problem.

Our work departs from these previous efforts in that it takes a two-stage optimization approach to the problem faced at Butler University and solves the problem with solvers available through SAS.

Each of the two programs has a data import and development stage, a solution stage, and a reporting stage all written in the SAS programming language.

THE CLASS CONFLICTS SCHEDULER

The class conflicts scheduler is a mixed integer linear programming problem with only binary variables where a variable indicates that a section of a class is to be offered at a certain day and time slot or not offered at that day and time slot.

Characteristics of the problem include:

CLASS CONFLICTS

- No sections of the same class at the same time.
- For classes with more than one section, at least one will start on Monday and at least one will start on Tuesday to spread the classes across the week primarily to accommodate internships.
- No junior or senior level classes within a discipline at the same time to avoid students in the same discipline being locked out of complementary classes due to a time conflict.
- No junior and senior level classes between management science and accounting at the same time. The management information systems minor draws heavily from accounting students.
- Certain other classes not to be offered at the same time such as Statistics and Information Technology, both sophomore level prerequisite courses.
- For classes with two or more sections, have at least one section in the morning and one in the afternoon so that athletes can schedule around their practice times.
- Similarly, for classes with only one section, have the class scheduled before 2:15 to accommodate athlete practice schedules.

TIMES TO BE AVOIDED

- No senior level classes on Monday, Wednesday, Friday schedules to accommodate a free schedule on Fridays for internships.
- Other classes such as international management classes not to be offered at certain times in order not to conflict with liberal arts and sciences offerings in international studies.

- No three-hour classes to be offered at the same time four or five days a week in order not to conflict with four and five hour courses.

TIMES TO BE FIXED

- Sections to be offered at fixed times in order to spread the offerings throughout the week when the program returned sections grouped together. Running the program is an iterative effort, refining the number of fixed time slots at each iteration.
- Classes to be offered at fixed times in order to accommodate adjunct faculty who frequently have time constraints.

SECTION COUNTS

Set the number of sections of each of the three-hour classes to be taught for the semester being constructed. This defines the course offerings for the semester.

NUMBER OF SECTIONS IN EACH TIME PERIOD

Set the minimum and maximum number of sections to be offered in a day and time period in order to spread the workload evenly. University administration would like to see more early morning and late afternoon sections being offered in order to use classroom space to better advantage.

THE DATA

SAS data steps were written to develop the MPS data to be fed to SAS's PROC OPTMILP mixed integer linear programming solver. Approximately 7,000 constraints and 1,900 binary variables were generated to construct a schedule of classes for 92 sections. The program takes about two and a half minutes CPU time to run on a Dell Optiplex GX620 tower computer executing 6000 iterations.

Observations about the results include:

- The schedule avoids all the time conflicts that could not easily be avoided by hand scheduling.
- More early morning and late afternoon sections were scheduled that better used classroom space but are unpopular with faculty.
- More Monday, Wednesday, Friday sections are scheduled in order to better use classroom resources but also are unpopular with faculty.

ASSIGNMENT OF FACULTY TO THE SCHEDULE OF CLASSES

A second program, an assignment algorithm, was developed to assign faculty to the now-constructed schedule of classes that has days and times associated with each section of each course to be offered.

Discipline coordinators (department heads) were asked to assign a skill utility to each instructor for each course he or she teaches with 5 being the highest and 1 being the lowest.

Additionally, each instructor was asked to assign a preference to each course he or she taught with 5 being highly preferred and 1 being not very well preferred.

The skill and preference value were combined to create an overall utility where individual preferences were given twice the weight of the skill value in order to give greater emphasis to what the instructor wants to teach.

SAS's PROC ASSIGN is used to perform the assignment to cover each class while maximizing overall utility. Dummy instructors with a utility of 1 for all classes are needed to obtain feasibility since there are more sections to be covered than there are regular faculty. These dummy assignments are then covered by adjuncts on an *ad hoc* basis.

EXPERIENCE

Observations about the results after the first live use of the programs for the spring 2008 semester include;

- Concerns by discipline coordinators that the program was not making best use of the faculty. This can be attributed to the discipline coordinator not spending enough effort to construct the skill utilities for his or her area.
- Dissatisfaction with the spread of class assignments over the day. This is a result of the programs taking into account management's desire to have classes in the early morning and the late afternoon.

CONCLUSION

The class conflict scheduler seems to be working adequately now and no improvements are envisioned at this time.

Straightforward improvements to the current assignment program include

- Work with the discipline coordinators to better develop the skill utility assignments for the discipline.

- Take into account seniority based on years at the university so that senior personnel have a higher utility at teaching a preferred course than those with less seniority

APPENDIX

THE CLASS CONFLICTS SCHEDULER

$$Max\ Z = \sum_{c,s,t} X_{c,s,t} \qquad (1)$$

$$Subject\ to:$$

$$X_{c1,s,t} + X_{c2,s,t} \le 1 \quad \forall\ s,t \qquad (2)$$

$$\sum_{t} X_{c,s,t} = 1 \qquad \forall\ c,s \qquad (3)$$

$$\sum_{s} X_{c,s,t} \le 1 \qquad \forall\ c,t \qquad (4)$$

$$\sum_{c,s} X_{c,s,t} \le \max_t \quad \forall\ t \qquad (5)$$

$$\sum_{c,s} X_{c,s,t} \ge \min_t \quad \forall\ t \qquad (6)$$

$$X_{c,s,t} = 0 \qquad c,s \in C \qquad (7)$$

$$X_{c,s,t} = 1 \qquad c,s \in F \qquad (8)$$

$$\sum_{t} X_{c,s,t} \ge 1 \qquad t \in AM \qquad (9)$$

$$\sum_{t} X_{c,s,t} \ge 1 \qquad t \in PM \qquad (10)$$

$$\sum_{t} X_{c,s,t} \ge 1 \qquad t \in M \qquad (11)$$

$$\sum_{t} X_{c,s,t} \ge 1 \qquad t \in T \qquad (12)$$

$$\sum_{t} X_{c,s,t} \le 1 \qquad t \in D_k \qquad (13)$$

$$X_{c,s,t}\ binary$$

c is course, s is section of the course, and t is day and time slot.

Equation (1) is the objective function that forces the schedule. It will equal the total number of sections being scheduled.

Equations (2) handle conflicts between two courses such as course c_1 and course c_2. Only one of the courses can be offered at time t.

Equations (3) force at least one section of a course to be taught.

Equations (4) communicate that only one section of a course is to be taught at a day and time slot.

Equations (5) limit the number of sections to be taught at a particular day and time slot.

Equations (6) force a minimum number of sections to be taught at a particular day and time slot. Equations (5) and (6) give the analyst some power to force the spread of classes over days and times.

Equations (7) communicate a time conflict. A particular course and section combination may not be offered at the time indicated in the set of time conflicts C.

Equations (8) communicate a fixed time. A particular course and section combination must be offered at the time indicated in the set of fixed times F. This is useful for achieving spread of sections across the week.

Equations (9) communicate that at least one section of a course, when there is more than one section, must be offered in the morning.

Equations (10) communicate that at least one section of a course, when there is more than one section, must be offered in the afternoon.

Equations (11) communicates that at least one section of a course, when there is more than one section, must begin on a Monday.

Equations (12) communicate that at least one section of a course, when there is more than one section, must begin on a Tuesday.

Equations (13) communicate that no three-hour section is to be offered at the same time four or five days a week.

ASSIGNMENT OF FACULTY TO THE SCHEDULE OF CLASSES

$$Max\ Z = \sum_{i,c} U_{i,c} X_{i,c} \qquad (14)$$

$$Subject\ to:$$

$$\sum_{i} X_{i,c} = 1 \qquad \forall\ c \qquad (15)$$

$$X_{i,c}\ binary$$

i is instructor and c is course section.

Equation (14) is the objective function that seeks to maximize the utility of instructors assigned to course sections. Currently, for running the assignment for fall 2008, $U_{i,c}$ = 0.5(management skill assessment) + (instructor preference for course c). For example, if management assesses that an individual has skill level 4 for an introductory accounting class but the instructor has only level 1 interest in teaching the course, the utility for that instructor teaching the introductory accounting course is 0.5(4) + 1 = 3. The 0.5 multiplier deflates the influence of the management skill assessment thereby giving the individual preference more power in the utility.

Equations (15) communicate that every course section is to have one and only one instructor.

REFERENCES

[1] Dinkel, J., Mote, J., and M. Venkataramanan. “An Efficient Decision Support System for Academic Course Scheduling.” *Operations Research*, Vol. 37, # 6 Nov-Dec (1989), pp. 853-864.

[2] Fizzano, P., and S. Swanson. “Scheduling Classes on a College Campus.” *Computational Optimization and Applications*. Vol 16 #3 (2000), pp. 279-294.

[3] Mooney, E., Rardin, R., and W. Parmenter. “Large-Scale Classroom Scheduling.” *IEEE Transactions*. Vol 28 #5 May (1996), pp. 369(10).

[4] Shih, W., and J. Sullivan. “Dynamic Course Scheduling for College Faculty via Zero-One Programming.” *Decision Sciences*. Vol 8 (1977), pp. 711-721.

[5] Wang, Y. “Using Genetic Algorithm Methods to Solve Course Scheduling Problems.” *Expert Systems with Applications*. Vol. 25 (2003). pp. 39-50.