eClasSkeduler: A Course Scheduling System for the Executive Education Unit at the Universidad de Chile

Jaime Miranda
Department of Management Control and Information Systems, School of Economics and Business, Universidad de Chile, Santiago, Chile, jmirandap@fen.uchile.cl

Each October, the Executive Education Unit at the Universidad de Chile develops its course schedules for the following year. By 2008, the complexities of increasing enrollments and course offerings had rendered its manual timetabling process unmanageable. Inconvenient and inflexible scheduling decisions were causing discontent among instructors and students, making the need for a more efficient system of assigning classrooms patent. Three characteristics distinguish the unit’s situation from the classic university course timetabling problem. First, its courses vary in duration, ranging between 15 and 30 weeks. Second, its course start dates are spread over the academic year. Finally, each course’s start date is flexible and must fall within a window defined by the earliest and latest start dates. This paper presents an automated computational system that generates optimal timetables and classroom assignments for all the unit’s courses, minimizing both operating costs and schedule conflicts. When we compared the schedules it generated with the unit’s manually generated timetables, we found that our system yielded average cost savings of 35 percent; in addition, it reduced execution times (for generating schedules) from two weeks to less than 30 minutes.

Key words: education systems: planning education systems; operations information systems: decision support systems, integer programming.

History: This paper was refereed. Published online in Articles in Advance February 16, 2010.
Figure 1: The graph illustrates the sustained growth in annual enrollments and number of courses taught.

has led to a corresponding increase in rescheduled courses (Table 1).

A range of critical organizational problems had also begun to surface under the prevailing system, causing much frustration and wasted time for EEU administrators, instructors, and students. These problems, which included assignment of classrooms with insufficient capacity, schedule conflicts (overlaps) between courses taught by a specific instructor, frequent changes in classroom assignments, and excessive assignment of special classrooms such as laboratories (labs), were often not detected promptly and were extremely difficult to resolve.

These problems affected the EEU negatively, and perceptions of disorganization damaged its image. The timetable conflicts increased operating costs because they necessitated course rescheduling, compensation mechanisms for affected students, and rental of classrooms outside of EEU premises. Rates for these unscheduled rentals were 200 percent higher than standard (classroom) rates. The EEU also incurred an additional cost of transferring students to the off-premise locations; in addition, if the rental required a lab, the expense of equipping it also had to be added to the cost.

To address these difficulties, the EEU embarked on developing “eClasSkeduler,” an automated computational system that generates optimal course timetables and classroom assignments. The objective of this paper is to describe this system and the use of an integer programming (IP) model within it to provide an optimal solution that minimizes operating costs and schedule conflicts. Implementing this system has reduced the time required for the timetable construction and report generation (i.e., ETTR, the second stage), which previously took an average of two weeks and now averages less than 30 minutes. Additional benefits include greater automation of the scheduling process, faster and higher-quality timetabling, the ability to explore multiple scheduling scenarios, and more efficient use and assignment of the EEU’s resources.

Problem Description and Requirements

In its simplest form, course timetabling is defined as the assignment of a set of courses to different time slots and classrooms while satisfying certain requirements (Hinkin and Thompson 2002). In this paper, we discuss a model that addresses a course timetabling solution for a university.

Three characteristics distinguish the EEU’s situation from the classic university course timetabling problem. First, because the EEU’s courses differ in the depth of treatment of their subjects, they vary in duration, ranging between 15 and 30 weeks. Introductory-level courses generally have fewer classes than specialization courses. Second, the courses have different start dates, which are spread over the entire academic year. Finally, each course’s start date must fall within a window defined by the earliest and latest start dates. A start date is flexible—it can be any day within a previously defined three-week window. These characteristics imply that the number of courses starting or ending in any given week can fluctuate. The EEU can schedule a course for
any date within the planning period if the end date is also within the same period. Because of these characteristics, different courses may start and end in any given week. Therefore, the solution to the timetabling problem must be formulated so that it schedules the entire academic year simultaneously. It cannot schedule a single base week and then replicate it over the entire calendar; each week must have its own timetable.

The problem must also incorporate other requirements and conditions to ensure that the solution is appropriate. Each course consists of two classes, either a lecture or lab, per week. A lecture can be given in any classroom; a lab class must be held in a computer lab. The two weekly classes must be held in three-hour time slots on different days between Monday and Saturday; the days must remain fixed throughout the course’s duration. Moreover, all classes for a given course must be scheduled over consecutive weeks, leaving no intermediate weeks without a class. The classroom assigned to a course must also be the same for the entire planning period. However, this restriction is relaxed for lab classes. Dates for the latter are defined a priori for each course at the start of the planning horizon.

A key aspect of the timetabling requirements is that for any given course, there is a set of conflict-excluded courses (i.e., excluded courses) that must not be scheduled on the same day. The courses in the set are those that a student would be most likely to take or an instructor would be most likely to teach during the same academic term. The EEU administration determines the excluded course sets and communicates them to the timetable programmers prior to scheduling.

In summary, generating the EEU course and classroom timetables is extremely complex. With the growing number of courses and students, plus the need to schedule closed courses on an ongoing basis, manual timetabling was no longer viable, and its continued use would lead to scheduling errors and inefficiencies in resource allocation.

Literature Review

The twin problems of timetabling and classroom assignment that educational institutions face have been widely studied in the literature (de Werra 1985, Mooney et al. 1996, Schaerf 1999a, Asratian and de Werra 2002, Lewis 2008). Both problems are classified as NP-hard (Schaerf 1999b, Pongcharoen et al. 2008). Many techniques to solve them are available, including a series of optimization models based primarily on linear and integer linear programming (Daskalaki et al. 2004, Daskalaki and Birbas 2005, MirHassani 2006, Hernández 2008) and various types of heuristics and metaheuristics (Elmohamed et al. 1998, Rossi-Doria et al. 2002, Lewis 2008, Pongcharoen et al. 2008). Many researchers have used specific heuristics such as tabu search (Alvarez-Valdés et al. 2002), simulated annealing (Elmohamed et al. 1998), ant colony algorithms (Socha et al. 2003), genetic algorithms (Wang 2003), multineighborhood local search (Gaspero and Schaerf 2003), multiagent methods (Oprea 2007, Strnad and Guid 2007), and hyperheuristics (Burke et al. 2003, 2007; Rattadilok et al. 2005). Other approaches use hybrid heuristics that combine two or more of the above techniques (White and Zhang 1998, Chiarandini et al. 2006) or offer designs built around constraint programming (Goltz and Matzke 1999, Abdennadher and Marte 2000, Rudová and Murray 2003).

de Werra (1985) describes a series of problems that arise in the mathematical modeling of course scheduling, instructor assignment, and examination timetabling. By adopting a theoretical perspective, de Werra’s paper lays the basis for developing future applications that address similar problems and actual study cases.

Some researchers have systematized their solution approaches and implemented computational systems to support decision makers at specific educational institutions. Bloomfield and McSharry (1979) propose a system that uses a heuristic procedure to solve a course scheduling problem. They discuss its low implementation costs and significantly reduced execution times. Whereas the equivalent manual process took three weeks to deliver a solution, their system took two days.

Ten years later, Chahal and de Werra (1989) implemented a system on a PC to resolve a course scheduling problem at an adult training school. They solve their network-flow model using a heuristic procedure that generates initial solutions in a matter of minutes.
Johnson (1993) implemented a computational course scheduling system at the Loughborough University Business School. The paper describing the system, which allows the user to estimate the effects and consequences of a given scheduling solution, shows that efficient management of course databases leads to better-quality schedules. The schedules generated are not necessarily feasible or optimal; however, Johnson stresses their system's usefulness in permitting greater automation of the processes.

Ferland and Fleurent (1994) present SHAPIR, a system that uses heuristic algorithms to solve an IP model and allows the user to manually adjust the schedules it generates. Three educational institutions of different sizes have tested the system; in each case, the results were good and execution times were reasonable.

Stallaert (1997) developed a system to automate course timetabling at UCLA's Anderson School of Management. In this approach, the courses are separated into two sets, core and noncore. The system uses an IP model whose objective function maximizes the schedule preferences of the core-course instructors; it then schedules the noncore courses using a heuristic procedure.

Foulds and Johnson (2000) developed SlotManager, which has a similar structure to our model, for use with a microcomputer. It has three components—interface, database, and model base. Its interface enables a user to enter information, e.g., type of student group, list of classrooms, and course characteristics, into the system using a series of menus. Its database is relational and contains the information that defines the schedules for a given semester or year. Finally, the model base contains the system's "intelligence," which solves instructor-schedule conflicts, classroom-capacity violations, and resource underuse. Sprague and Carlson (1982) originally presented this structure.

Hinkin and Thompson (2002) created SchedulExpert, a system that also incorporates an IP model that they implemented for course timetabling at Cornell University's School of Hotel Administration. The set of decision variables in the model assigns courses to patterns, which are combinations of class days, time slots, durations, and classroom-equipment requirements. The type of variable used is similar to the type that we used in our formulation. Their model addressed all the schedule-conflict conditions that Cornell imposed; in addition, the time required to generate the timetables dropped from several weeks to a few hours.

Other studies (Carter 2001, Dimopoulou and Miliotis 2001, Alvarez-Valdés et al. 2002, Haase et al. 2004, Oprea 2007) have reported diverse decision support systems that incorporate mathematical programming models for generating course timetables and classroom assignments. In each case, the system contributed numerous improvements to the decision-making process.

System Description

According to the classification defined by Liang et al. (2008), we can consider eClasSkeduler as supporting decision making based on mathematical models because it generates an optimal solution that is consistent with a series of constraints. It uses a menu-based interface that allows the user to modify all the necessary parameters (Figure 2).

The input information module is a database that stores all information relating to courses, classrooms, and instructors. For each course, it includes data items such as the following:

1. Estimate of student enrollment, which is derived from an historical demand forecast, and the estimate is adjusted as registration proceeds;
2. Course duration;
3. Window defined by the start dates;
4. Lab class dates; and
5. Corresponding excluded courses. The database stores each classroom's capacity, rental cost, and any technological resources that it contains.

The user interface module serves as the system control mechanism. It transforms the input data kept in the input information module as specified by the requirements and formats of the IP optimization model. It also handles the definition of the weight coefficients that set the relative importance of the different objectives in the objective function. Finally, it allows various technical parameters to be defined, including the maximum number of iterations, the gap
between the linear relaxation and the best integer solution, and the numerical tolerances.

The optimization module contains the code for the IP model (see the appendix) and a directory that stores the specific models of previously solved instances. These may be classified according to the combination of objectives defined in each case. A CPLEX 9.0 solver (ILOG 2003) is used to manipulate and execute the models and find the optimal solutions.

The report generation module receives the solution results as input; it processes and transforms them to produce management reports, including reports on course schedules for specific fields of study, the use of classrooms, and schedules for individual instructors. It also generates performance indicators for management control and resource allocation such as unused classroom capacity for each time slot, extra lab rentals, and the number of schedule conflicts between excluded courses.

**Optimization Module: Description**

The optimization module contains an integer linear programming model (see the appendix) that defines the optimal timetables for all EEU annual courses. The model incorporates all the EEU’s objectives, as well as the requirements and constraints for ensuring its normal operation. It attempts to assign available EEU classrooms efficiently relative to cost and capacity, avoiding the use of external premises and schedule conflicts between excluded courses wherever possible.

The model assigns each course a unique timetable pattern defined by a vector with three components—a classroom, a time slot for two different days, and a period of consecutive weeks over which the course is taught. For a course to be assigned to a given timetable pattern, its classroom-capacity component must be equal to or greater than the number of
registered students, the period of consecutive weeks must equal the course duration, and the first class must fall within the window defined for the course start date. If the pattern meets all these conditions, it is classified as a feasible pattern for that course.

The model’s objective function expresses four fundamental objectives reflecting the various concerns of the EEU—see Equation (1) in the appendix. The first term of the function represents total classroom rental cost. This factor depends on a classroom’s characteristics such as years of use and installed equipment; the latter typically includes data show projectors, personal computers, electronic blackboards, and audiovisual equipment.

The second term is the opportunity cost of assigning a classroom to a course with fewer registered students than the room’s capacity. The difference between course enrollments and nominal classroom capacity is defined as unused capacity. The EEU administration valued the cost of an empty seat based on the loss that the EEU suffered when a student could not register in a closed course because of a lack of classroom capacity. Note that because closed courses are not scheduled, we could not know the number of their enrollments with certainty. Therefore, the objective function should minimize the unused capacity across the entire course schedule to ensure the maximum possible number of available seats. This objective seeks to ensure that the EEU’s infrastructure is used efficiently, assigning classrooms with excess capacity only when absolutely necessary.

The third term in the objective function is the total rental cost of external classrooms. This factor includes both lecture rooms and computer labs, which as noted earlier average three times the cost of comparable space within the EEU’s premises. The fourth term is the total cost associated with the schedule conflicts between excluded courses. The EEU administrators also quantified this because the EEU must hire an extra instructor when a conflict occurs. Note that they determined this cost and the opportunity cost of an empty seat based on historical data. Thus, all four objective function terms were economically quantified, resulting in a single cost function to be minimized.

Implementation Results

In this section, we analyze the results of an experiment we conducted to study the strengths of the system; we compared the timetables it generated for the academic years 2007 through 2009 with those developed using the EEU’s manual process. In each case, the manual schedules had been drawn up the previous October; our model used the same data available to the EEU’s schedulers. The planning horizon for each year was 40 weeks, with six weekly time slots (six days and one block per day). In the analysis, we looked at a series of performance indicators (Table 2) that reflect various management and resource-allocation criteria; we included all the objectives in the IP objective function and weighted them equally.

The time needed for timetable construction and report generation (ETTR) was cut dramatically from an average of two weeks to less than 30 minutes (Table 3), thus enabling the EEU to respond quickly to requests from private companies for closed courses. High-speed scheduling also means that multiple scenarios using different objective weights can be analyzed and compared.

In comparison with the manual system, our system provided a substantial decline in classroom rental

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETTR</td>
<td>Total time required to generate the course timetables and classroom assignments.</td>
</tr>
<tr>
<td>Total rental cost</td>
<td>Rental cost associated with the assigned classrooms.</td>
</tr>
<tr>
<td>Unused classroom capacity</td>
<td>Total number of empty seats in all classrooms assigned by the system. This value is calculated as the difference between the capacity of a classroom and the number of enrollments in the course assigned to it.</td>
</tr>
<tr>
<td>Number of courses with schedule conflict</td>
<td>Number of excluded courses that conflict with each other.</td>
</tr>
<tr>
<td>Number of external lecture rooms</td>
<td>Number of external lecture rooms assigned by the system. These classrooms are located off EEU premises.</td>
</tr>
<tr>
<td>Number of external labs</td>
<td>Number of lab rooms assigned by the system that are located off EEU premises.</td>
</tr>
</tbody>
</table>

Table 2: The performance indicators include criteria for resource-assignment efficiency as well as operability criteria. The results of the comparisons demonstrated that our system provided significantly reduced execution times.
costs because two of the four objective function terms indicated lower total operating costs. eClasSkeduler boosts efficiency in using economic resources, thus also increasing profits. Table 4 shows the decline in the total rental costs.

Our system also resulted in a significant reduction in unused classroom capacity. Its method of assigning courses to classrooms left fewer empty seats per academic year; thus, it reduced rental costs because they are directly proportional to nominal capacity (Table 5). Under the manual system, the number of available classrooms grew each year, resulting in an increasing underuse of total nominal capacity.

Using the timetabling system, we also observed a decline in the number of excluded course-schedule conflicts (Table 6). Students had more flexibility to register for advanced courses in a given subject field, thus increasing their satisfaction levels.

eClasSkeduler also generated schedules that required fewer external classrooms and labs. The manual schedules assigned 18 external classrooms and eight additional labs between 2007 and 2009; in contrast, our system generated schedules requiring only three classrooms and one lab (Tables 7 and 8).

Table 3: The data show that the time required for our system to generate schedules could be measured in minutes, as the ETTR time illustrates. In comparison, the manual system required an average of two weeks to produce a complete course timetable and classroom assignment.

<table>
<thead>
<tr>
<th>Instance</th>
<th>Number of courses</th>
<th>Number of classrooms</th>
<th>Number of excluded course sets</th>
<th>Number of lab classes</th>
<th>ETTR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td>30</td>
<td>20</td>
<td>12</td>
<td>112</td>
<td>19.6 min</td>
</tr>
<tr>
<td>2008</td>
<td>38</td>
<td>25</td>
<td>20</td>
<td>170</td>
<td>22.6 min</td>
</tr>
<tr>
<td>2009</td>
<td>55</td>
<td>25</td>
<td>30</td>
<td>182</td>
<td>28.7 min</td>
</tr>
</tbody>
</table>

Table 4: The data illustrate the total cost (in US$) of rental costs in schedules generated by the manual process and by eClasSkeduler. The difference averaged 35 percent.

<table>
<thead>
<tr>
<th></th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEU’s manual process</td>
<td>60,300</td>
<td>69,420</td>
<td>90,230</td>
</tr>
<tr>
<td>eClasSkeduler</td>
<td>44,500</td>
<td>41,290</td>
<td>55,440</td>
</tr>
<tr>
<td>Difference (%)</td>
<td>26</td>
<td>41</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 5: The data show the decline in unused classroom capacity (i.e., empty seats) for each timetabling system; this decline averaged 52 percent over the three-year period.

<table>
<thead>
<tr>
<th></th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEU’s manual process</td>
<td>314</td>
<td>413</td>
<td>575</td>
</tr>
<tr>
<td>eClasSkeduler</td>
<td>251</td>
<td>132</td>
<td>138</td>
</tr>
<tr>
<td>Difference (%)</td>
<td>21</td>
<td>61</td>
<td>74</td>
</tr>
</tbody>
</table>

Table 6: Comparing the results of the manual system with our system, we see that the number of excluded courses with schedule conflicts declined each year.

<table>
<thead>
<tr>
<th></th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEU’s manual process</td>
<td>9</td>
<td>12</td>
<td>34</td>
</tr>
<tr>
<td>eClasSkeduler</td>
<td>1</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>Difference (%)</td>
<td>8</td>
<td>7</td>
<td>19</td>
</tr>
</tbody>
</table>

Table 7: Using eClasSkeduler, the number of external lecture rooms required decreased significantly each year; none was required in 2009.

<table>
<thead>
<tr>
<th></th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEU’s manual process</td>
<td>9</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>eClasSkeduler</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Difference (%)</td>
<td>7</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 8: Using eClasSkeduler required significantly fewer external labs each year.

<table>
<thead>
<tr>
<th></th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEU’s manual process</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>eClasSkeduler</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Difference (%)</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

Our system virtually eliminated the logistical problems of equipping off-premise classroom space and providing transportation, thereby sparing the EEU all the associated costs and the damage to its corporate image.
Discussion of Results
The objective function of the model minimizes the total cost expressed as a weighted sum of its various terms—see Equation (1). In our experiment, we gave each term a weight of 1 in line with the desire of EEU administrators to minimize the total cost of the timetable scheduling. Note, however, that if we ranked each term in accordance with its contribution to the final cost function, the most important objective would be the one with the largest contribution. The EEU administrators validated our results and concluded that the timetables generated made efficient use of resources at minimum cost. In the light of this performance, the EEU implemented the timetables that eClasSkeduler produced for 2009. We did not directly approach the search for the most appropriate weighting factors for each term in the objective function; however, we intend to incorporate a method of determining the relative weights for these terms in the decision-making process in a future study. Existing research reports several interesting solution approaches that address the multiobjective nature of certain applications and business problems (Bana e Costa et al. 2008, Reeves and Hickman 1992, Teodorović and Krčmar-Nožić 1989), suggesting that this is a fertile area for further research and development.

Conclusions
This paper has presented a timetabling system that uses an IP model to solve the problem of scheduling classes and assigning classrooms at the EEU, a branch of the Universidad de Chile.

eClasSkeduler generated satisfactory results for all involved participants in the EEU. In particular, it curtailed operating costs and lowered unused classroom capacity—the two most important findings from the institution’s viewpoint because they lead to more efficient use of economic and infrastructure resources. In addition, the system produced fewer course schedule conflicts and off-premise classroom assignments, thus allowing students to have more options in selecting courses in the same academic term and increasing the EEU’s attractiveness to students. In addition, the decreased need to use external resources also reduced the need for students and instructors to travel from the EEU’s premises.

Based on these results, we conclude that incorporating our method into the EEU’s decision-making process would be beneficial. First, it would enable the EEU’s management to choose the best of a series of options by evaluating multiple scenarios through simple manipulation of the system parameters. Second, the system would considerably shorten execution times from two weeks under manual scheduling to mere hours. This would allow the EEU to respond rapidly to changing requirements, significantly reducing the risks and uncertainties inherent in the manual approach. Third, the system automates what is otherwise the extremely slow, tedious, and complicated task of scheduling by hand.

Appendix
We begin the presentation of our scheduling model by introducing the necessary notation. The set of courses is denoted \( C \) and the set of available classrooms (excluding labs) is defined as \( S \). A timetable pattern is a set of timetables and classrooms in which each element consists of a triplet defined by a classroom—a period consisting of consecutive weeks and a time slot. For example, if a course \( c \) is assigned the pattern 1903/6-25/Mon-Wed, its classes are all held in Room 1903, begin in Week 6, end in Week 25 (a duration of 20 weeks), and are given in the Monday and Wednesday time slots. The EEU has more than 5,000 unique patterns.

Note that courses can only be assigned to a pattern that is feasible; i.e., it must fulfill specific conditions. A pattern is feasible for a course \( c \in C \) if the restrictions listed below are satisfied.

(1) All classes are scheduled in the same classroom, which has sufficient capacity for the course’s enrollment. A classroom’s capacity is denoted \( s \in S \), and enrollment is represented by the parameter \( D_c \) for course \( c \).

(2) The first week of classes falls within the course’s time window.

(3) The number of scheduled weeks matches the duration of the course.

If we define \( P \) as the set of feasible patterns, then \( P(c) \) is the set of feasible patterns for course \( c \) and \( P = \bigcup_{c \in C} P(c) \). Let us denote \( t = 1, \ldots, T \) as any day in the academic calendar on which classes may be held. We
also define $a_{ps}$ as a parameter that takes the value 1 if pattern $p$ includes a class held on day $t$ in room $s$, and 0 otherwise. Furthermore, for each day $t$ and for each $n \in \mathbb{N}$, the set $TL(t, n) \subseteq P$ is the set of patterns $p$ such that day $t$ is the $n$th class of the pattern.

For each course $c$, the term $Lab(c)$ is the set of lab classes for course $c$ that are counted correlatively with the lecture classes. Thus, if course $c$ includes two lab classes, one in the third week and one in the ninth, the set of these classes is defined as $Lab(c) = (3, 9)$.

Finally, for each $c$, we also define a corresponding set $CI(c)$ of excluded courses that are not scheduled on the same day, if possible. Note that if $a \in CI(c)$ and $b \in CI(c)$ are excluded courses with respect to course $c$, they are also excluded relative to each other.

The model contains three sets of decision variables. The first set consists of binary variables that are activated if a given course is assigned to a specific timetable pattern. More specifically, for each course $c$ and pattern $p \in P(c)$, the variable

$$X_{cp} = \begin{cases} 1 & \text{if course is given as determined by the pattern } p, \\ 0 & \text{otherwise.} \end{cases}$$

The second set consists of nonnegative integer variables that represent the use of external lecture rooms and labs. Countervariables $z_t$ and $y_t$ track the number of extra lecture rooms and labs, respectively, needed on day $t$. In the present context, all labs have the same capacity, equipment, and other resources; in contrast, the capacities of lecture rooms vary and their costs therefore depend on their assignments. For each course $c$, countervariable $q_{ct}$ tracks the number of schedule conflicts on day $t$ between $c$ and its corresponding excluded courses. The third set is composed of nonnegative integer variables that count the number of schedule conflicts among excluded courses. Note that the last two decision variable sets were included given that on specific occasions not all of the operating requirements and conditions can be completely satisfied. In such cases, the model will incorporate slack variables.

We can see that the objective function, which is represented by Equation (1), combines four objectives. The first objective minimizes the total cost of the course-classroom assignment. The parameter $F_{cp}$ in this term is defined as the cost of scheduling course $c$ according to pattern $p \in P(c)$. The second minimizes unused classroom capacity to prioritize filling all seats. The term $Cap_s - D_t$ represents the difference between the capacity of a given classroom assignment and the number of enrollments in a particular course. The third objective minimizes the cost per day of using extra lecture rooms and labs. The parameters $g_t$ and $d_t$ are defined as the unit cost of using a lecture room and lab room, respectively, per day $t$. The fourth and last objective minimizes the number of schedule conflicts with excluded courses.

$$\min z = C_1 \sum_{c \in C} \sum_{p \in P(c)} F_{cp} X_{cp} + C_2 \sum_{c \in C} \sum_{p \in P(c)} (Cap_s - D_t) X_{cp}$$
$$+ C_3 \sum_{t \in T} [d_t y_t + g_t z_t] + C_4 \sum_{c \in C} \sum_{t \in T} q_{ct}. \quad (1)$$

Note that the objective function terms all have individual weight coefficients $C_i (i = 1, \ldots, 4)$ that reflect the relative importance of the different objectives as defined by the system user. The function includes criteria for resource-assignment efficiency to minimize operating costs and unused classroom capacity, as well as operability criteria to minimize course conflicts.

Equation (2) ensures that each course $c \in C$ is scheduled using only one of the possible feasible patterns defined in the set $P(c)$:

$$\sum_{p \in P(c)} X_{cp} = 1 \quad \forall c \in C. \quad (2)$$

The second set of constraints, which is represented by Equation (3), avoids schedule conflicts in the use of lecture rooms, ensuring that no more than one class is held in each classroom $s \in S$ on a given day $t \in T$. If more lecture rooms are needed, the slack variable $z_t$ is activated, and the objective function value is penalized.

$$\sum_{c \in C} \sum_{p \in P(c)} a_{ps} X_{cp} \leq 1 + z_t \quad \forall s \in S, \forall t \in T. \quad (3)$$

The third constraint set, which is represented by Equation (4), is similar to the second constraint set but applies to labs. It guarantees that the number of courses using a lab on a given day $t \in T$ does not exceed the number available, which is defined by the parameter $NLabs$. If more labs are needed, the slack
avoided, the variable $y_t$ is activated, and the objective function value is penalized.

$$\sum_{c \in C} \sum_{n \in \text{Lab}(c)} \sum_{p \in T_L(c, n) \cap P(c)} x_{cp} \leq NLabs + y_t \quad \forall t \in T. \quad (4)$$

The fourth and final constraint set, which is defined by the Equation (5), requires that wherever possible, for any day $t \in T$ and course $c$ not corresponding to an excluded course as defined by the set $C_l(c)$ be given on the same day. If a schedule conflict cannot be avoided, the variable $q_{cl}$ is activated, again imposing a penalty on the objective function value.

$$\sum_{j \in C_l(c)} \sum_{p \in P(j) \cap S} a_{jcp} x_{jp} \leq 1 + q_{cl} \quad \forall c \in C, \forall t \in T. \quad (5)$$

With the necessary notation now defined, we show the IP model:

$$\min \quad z = C_1 \sum_{c \in C} \sum_{p \in P(c)} F_c x_{cp} + C_2 \sum_{c \in C} \sum_{p \in P(c)} (Cap_c - D_c) x_{cp} + C_3 \sum_{t \in T} \sum_{c \in C} \sum_{t \in T} d_{cyt} + g_{zct} + C_4 \sum_{c \in C} \sum_{t \in T} q_{cl}$$

s.t. \quad \sum_{p \in P(c)} x_{cp} = 1 \quad \forall c \in C,

$$\sum_{c \in C} \sum_{p \in P(c)} a_{jcp} x_{jp} \leq 1 + z_t \quad \forall s \in S, \forall t \in T,$$

$$\sum_{c \in C} \sum_{n \in \text{Lab}(c)} \sum_{p \in T_L(c, n) \cap P(c)} x_{cp} \leq NLabs + y_t \quad \forall t \in T,$$

$$\sum_{j \in C_l(c)} \sum_{p \in P(j) \cap S} a_{jcp} x_{jp} \leq 1 + q_{cl} \quad \forall c \in C, \forall t \in T,$$

$$x_{cp} \in [0, 1] \quad \forall c \in C, \forall p \in P(c),$$

$$z_t, y_t, q_{ct} \in \mathbb{Z}^+ \quad \forall t \in T, \forall c \in C.$$

Acknowledgments
I would like to express my special thanks to Francia Lara and to professors André Weintraub, Rafael Epstein, Guillermo Durán, Pablo Rey, Antoine Sauré, Mario Morales, Sigifredo Laengle, and Ariel La Paz for their valuable comments on the final version of this article, and to Juan Enrique Negri, director of the Executive Education Unit, for his support in carrying out this research. I gratefully acknowledge the helpful suggestions provided by the anonymous referees.

This research project was partially supported by the Millennium Institute “Complex Engineering Systems.” The Institute is a Center of Excellence supported by state funds.

References


Juan Enrique Negri Echeverria, Director, Executive Education Unit, School of Economics and Business, Universidad de Chile, Diagonal Paraguay 257, piso 19, Santiago, Chile, writes: “I am writing as the director of the Executive Education Unit, part of the School of Economics and Business at the University of Chile, in regard to our experience with a newly developed course scheduling and classroom assignment system.

“The main reason for the incorporation of this computational system into our timetable decision-making process was the complex situation facing the Unit around the end of 2007. Widespread discontent had surfaced among both academic staff and students with the Unit’s course schedules, which were marred by numerous errors and complications such as conflicts between courses typically taken simultaneously or given by the same instructor, assignment of classrooms with insufficient capacity, and conflicts between laboratory room assignments.

“The Unit’s principal scarce resource is its classrooms, and particularly computer laboratories. The occurrence of a course conflict meant that additional space would have to be subcontracted, often outside the Unit’s premises. This led to extra costs due to high rents (200% above normal levels), setting up and equipping the space (particularly for lab classes, which require installation of software), and the transfer of students to an off-site location. These scheduling problems were often not detected early on, resulting in damage to the institution’s image.
“The new computational scheduling system went into operation this year (2009). Since the 2009 timetables were defined in October 2008, and some courses actually began before the system was implemented, the schedules it generated could not be used in their entirety. However, as the superiority of the new scheduling became apparent, the timetables of all those courses that had not yet begun were modified.

“In my personal view, the computational system has delivered considerable qualitative and quantitative benefits in the short term. Qualitatively, greater satisfaction on the part of instructors and students has already been noted in the reduction in complaints received, thus strengthening the Unit’s image as well as improving the labor relations climate in the operations section, which is directly responsible for the schedules.

“Quantitatively, three benefits in particular are worth noting. First, there has been an increase in the efficient use of resources, both economically in terms of a significant decline in total costs, and physically in the form of better utilization of classroom space. This latter point has enabled us to offer a greater number of courses and respond better to non-scheduled requirements such as requests for closed courses by private companies. These courses are extremely important strategically for the Unit as they reinforce its relationship with the country’s major industries. Second, there has been a substantial reduction in the number of person-hours required to generate the timetables. The time involved has fallen from weeks under the manual system to a couple of hours, thus freeing up much of the staff formerly assigned to course scheduling for other important tasks. And third, with the ability to timetable rapidly, alternative scenarios can be examined and their cost variations estimated. To these benefits we may also add the fact that the entire process is now much more automated.

“In conclusion, I would like to express our gratitude to the operations research and management science professions, who have made possible the new scheduling system that is proving to be a great source of support in our operational decision-making and an efficient manager of the various objectives of the Unit.”