The International Journal of Applied Management and Technology

A refereed journal published by the School of Management, Walden University

The International Journal of Applied Management and Technology (iJAMT) is published biannually in May and November. Publication is via digital media and available for viewing or download from the journal's web site at http://www.ijamt.org.

ISSN 1554-4740
www.ijamt.org | info@ijamt.org

Submissions are welcomed from scholars, scholar-practitioners, and advanced graduate students. Please visit the journal's web site at www.ijamt.org for detailed information.

The International Journal of Applied Management and Technology is a peer reviewed publication.

MODEL-BASED STUDENT ACADEMIC PLANNING

Avi Dechter, California State University

Abstract

Planning and managing the progress toward the degree is a daunting challenge for many college students due to complex and often confusing graduation requirements and institutional rules. As a result, many students take far longer than is necessary or desirable to graduate. In this paper we view the problem of completing a degree program in a timely manner as a constrained optimization problem and consider its modeling using both constraint programming and integer programming methodologies. We also discuss how constrained optimization tools might be used to help students, as well as university policy makers, to manage issues of academic planning

Keywords

Academic planning, Time-to-degree, Constraint programming, Integer programming

Introduction

Most college degree programs in the U.S. are intended, as the term "four-year college" implies, to be completed in four years. Yet, many college students now take far longer than that to attain their degrees. Data collected by the U.S. Department of Education (NCES, 2004) shows that only 33% of first-time freshmen entering four-year bachelor's degree programs in the U.S. complete their degrees within four years. The 5-year and 6-year average graduation rates are 50% and 56%, respectively. Graduation rates vary widely among different U.S. institutions and 4-year rates can be as low as 10%.

The lengthening of the time to graduate has been attributed primarily to the changing characteristics of the student population (Astin & Oseguera, 2005). Prominent among these changes is the increase in number of students who are ill- prepared academically and, therefore, are required to take remedial courses and are more likely to have to repeat courses before passing them successfully. There has also been an increase in the number of students who, because of financial or other circumstances, cannot maintain the full course load required for timely completion of the degree requirement. Given these factors, which are largely beyond the control of higher education institutions, the importance of helping students in planning and monitoring their progress toward graduation has increased. Poor planning leads to further delays in completing a degree program as it may result in students taking more courses than are needed to satisfy the requirements for the degree and to incorrect scheduling decisions. Unfortunately, in order to avoid such errors students and their academic advisors must confront a rather complex scheduling problem.

The pursuit of a college degree is a major endeavor that shares many characteristics with what commonly is referred to as a project, e.g., the construction of a new warehouse. Degree programs, just like projects, are defined in terms of a set of activities (courses) that must be completed (passed) in order to reach a well-defined end result (graduation). Course prerequisites are akin to the precedence relationships that characteristically exist among the project activities (e.g., the construction of the wall must precede the installation of windows in that wall), and institutional or self-imposed course-load restrictions affect the time to complete a degree program in the same way limited resources (e.g., labor) affect the time it would take to complete a project. Table 1 summarizes the parallels between a degree program and a project.

Table 1: Parallels between a Project and Degree Program

Project	Degree Program
Activities	Courses
Precedence relationships	Course prerequisites
Resource limitations	Course-load limits
Goal: minimize completion time	Goal: minimize time-to-degree

The practice of project management has benefited tremendously from mathematical modeling (Williams, 2003). Mathematical models are the basis of the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT), which have helped project managers in controlling projects' completion times and costs. These models and their extensions are also at the core of widely available computer software for project scheduling and management.

College students (and their academic advisors) would greatly benefit from such models in planning and controlling their progress toward graduation. Unfortunately, despite the similarities between the two problems, project scheduling models are not directly applicable for the academic planning problem due to a fundamental difference between the two problems. The set of activities that must be accomplished to complete a project is typically known (or assumed to be given) so that the model is only concerned with scheduling of the activities. In contrast, most degree programs are structured so that they can be completed by taking different sets of courses (due to the wide inclusion of elective courses). The model in this case must handle both decisions, something standard project scheduling models are not designed to do.

Project scheduling under resource constraints is a notoriously hard combinatorial optimization problem that has received large amount of attention in the Operations Research literature (Brucker *et al.*, 1999; Kolish & Padman, 2001). There are two general approaches for modeling and solving such problems. The traditional approach is to use integer programming (IP) which is rooted in Operations Research (Patterson & Huber, 1974; Stinson *et al.*, 1978). A more recent approach uses constraint programming (CP), an emergent software technology for declaring and solving constraint satisfaction and constrained optimization problems (Caseau & Laburthe, 1997; Baptiste *et al.*, 2001), rooted in Computer Science. Constraint programming is free from many of the restrictions imposed by integer programming (in particular that only linear inequalities may be used to represent constraints) and makes modeling of the problem more "natural." Solutions techniques for solving integer programming models, on the other hand, are more developed and can solve, using current technology, larger problems. In this paper we use both approaches to discuss the modeling of the academic planning problem and its potential uses.

We begin, in the next section by discussing the structure of degree requirements and by showing, using a small example, that the time needed to complete the program depends on the courses selected to satisfy the requirements. Next we develop both a CP model and an IP model for the example. To illustrate the structure and complexity of a "real life" problem, we then describe in detail the requirements and rules governing a specific degree program at the California State University, Northridge and discuss its modeling. Lastly, we discuss the potential benefits of using optimization technology for degree planning and point to some possibilities of future research.

The Structure of Degree Requirements

Academic requirements can be quite complex and they vary widely in structure from one academic institution to another. However, a typical degree program is defined in terms of a set of **course requirements** as well as the specification of course prerequisites. The purpose of the course requirements is to guarantee that the student's academic plan covers specific subjects or areas of knowledge. A typical course requirement consists of a list of courses, all of which or, more typically, a subset of which must be completed successfully by the student in order to satisfy the requirement.

The purpose of specifying prerequisite requirements for a course is to ensure that courses are taken by the student in logical sequence – for example, leading from introductory courses to more advanced courses – and that the students posses the necessary knowledge for getting the full benefits from the course. In general, prerequisites are courses, test scores and other conditions that must be completed or satisfied before (or sometimes simultaneously with) taking a specific course. Often, however, the prerequisites for a course are specified in terms of a list of courses, all of which or a subset of which must be completed successfully before the course may be taken.

To illustrate the effect of the selection of courses on the time to complete a degree program, consider a small, fictitious degree program based on a group of 12 courses that are listed, along with their prerequisites, in Table 2. There are just two course requirements. To satisfy Requirement 1, a student must select at least 3 courses from courses C4, C5, C6, and C7. To satisfy requirement 2, the student must take at least 2 courses from courses C8, C9, C10, C11, and C12. Suppose further that a student may enroll in no more than 3 courses in a single term.

Table 2: Courses Requirements for Sample Degree Program

Course	Prerequisites
C1	None
C2	None
C3	C2
C4	C2
C5	C3
C6	C1
C7	C5 and C6
C8	C5
C9	C4
C10	C7 and C8
C11	C6
C12	C11

It is useful to represent the relationships among the 12 courses in term of a precedence diagram as shown in Figure 1, where nodes represent courses and arrows represent prerequisite requirements.

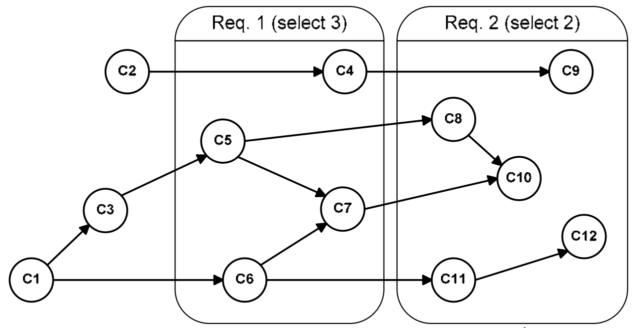

Figure 1: A Precedence Diagram and Course Requirements for Sample Program

Clearly, there are many course selections that would simultaneously satisfy the two course requirements and the prerequisite requirements. Consider two possible plans, Plan A and Plan B. Under Plan A, Courses C5, C6, and C7 are selected to satisfy Requirement 1 and courses C8 and C10 are selected to satisfy Requirement 2. Under plan B, Courses C5, C6, and C7 are selected to satisfy Requirement 1 and courses C11 and C12 are selected to satisfy requirement 2. These selections are highlighted on the precedence diagrams in Figure 2. The courses that are prerequisites of the selected courses must also be taken and are highlighted as well.

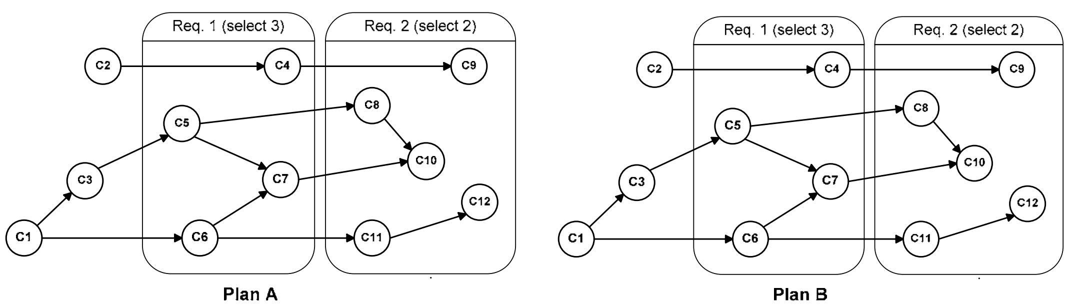

Figure 2: Two Course Selection Plans for the Sample Degree Program

Both plans consist of 7 courses and both clearly satisfy the two course requirements as well as the prerequisite requirements. However, they differ in terms of the total time if would take to complete them. Plan A would require at least 5 terms to complete because it includes 5 courses (C1, C3, C5, C8, and C10) that must me taken in sequence. The longest sequence of this type in Plan B consists of just 4 courses (C1, C6, C11, and C12) and, therefore, this plan can be completed in 4 terms (assuming, as the case here is that the course load limit is sufficiently large and is not restrictive). Course schedules showing how Plan A may be completed in 5 terms and Plan B in 4 terms are shown in Figure 3. As this example demonstrates, the time to complete a degree program depends on the courses selected by the students to satisfy the course requirements.

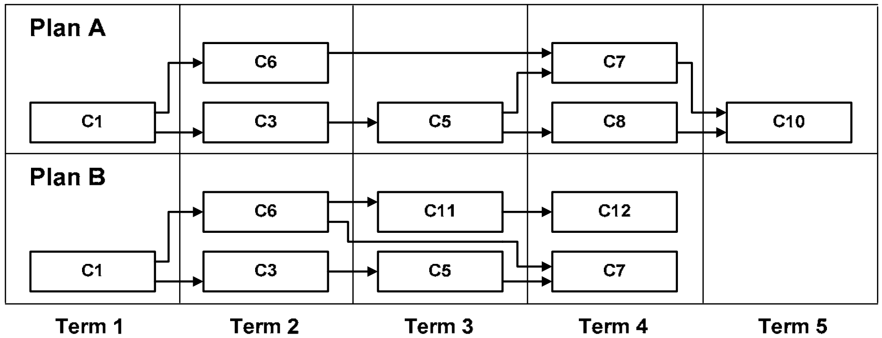

Figure 3: Minimum Length Schedules for Plan A and Plan B

Close inspection of this degree program reveals that Plan B is not optimal and that, in fact, this degree can be completed in three terms by selecting courses C4, C5, and C6 to satisfy Requirement 1 and courses C9 and C11 to satisfy Requirement 2. The resulting plan, Plan C, together with a course schedule showing how it may be completed in 3 terms is shown in Figure 4. Notice that while Plan C allows the completion of the program in fewer terms than either Plan A or Plan B, it requires taking more courses (8 courses compared to 7). This demonstrates the fact that smaller plans (in terms of their total course load) are not necessarily shorter. In this case, in fact, there is no 7 course plan that could be completed in fewer than 4 terms. Therefore, it would generally not be correct to separate the course selection decision from the course scheduling decision: they must be considered together. In the next section we develop both a constraint programming model and an integer programming model for selecting and scheduling courses for this program so the program is completed in the smallest possible number of terms.

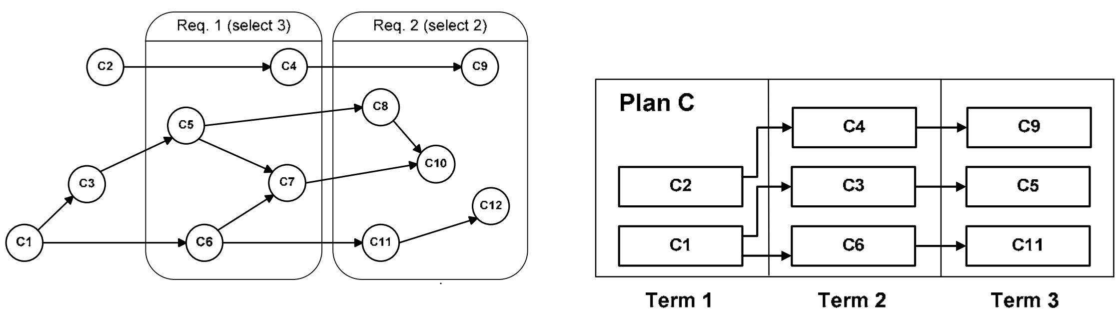

Figure 4: Course Selection and Schedule for Plan C

Modeling the Academic Planning Problem

We start this section by developing a constraint programming (CP) model for the sample academic planning problem discussed in the preceding section. We then show how the CP model may be modified to create an integer programming (IP) model. Our objective is to illustrate the similarities and differences of these two modeling approaches. Both are similar in that they require the specification of (decision) variables, constraints, and (in the case of optimization) an objective function. The models are written in OPL, which is a language designed to express, in a rather straightforward manner, both constraint programming and integer programming models (Van Hentenryck, 1999).

Let `Courses = {C1,C2,...,C12}` be the set of all the courses available for the student and let `Terms = {1,2,...,nbTerms}` represent a “planning horizon” consisting of `nbTerms` consecutive academic terms (e.g., semesters, quarters). The parameter `nbTerms` should be sufficiently large to ensure that degree requirements can be satisfied within that number of terms. In our case we may choose, for example, `nbTerms = 12`, which would allow taking all 12 courses available for the student, and never having to take more than one course in any one term. We next develop a CP model.

Decision Variables

The decision variables should reflect the two basic decisions involved in academic planning: (a) which courses should be selected for the plan and (b) for which term each of the selected courses should be scheduled to be taken. For each course c in Courses, we associate a binary variable (i.e., that may be assigned either the value of 1 or the value of 0) `take[c]` to represent the course's selection decision:

$$\texttt{take[c]}=\begin{cases}1 & \texttt{if course c is selected for the program}\\ 0 & \texttt{otherwise}\end{cases}$$

To represent the scheduling decision, we associate with course c in Courses an integer variable term[c] that may be assigned values in the range `0,1,...,nbTerms` such that:

$$\texttt{term[c]}=\begin{cases}\texttt{t} & \texttt{if course c is scheduled for term t in Terms}\\ 0 & \texttt{if course c is not selected for the program}\end{cases}$$

The consistency between these two sets of variables is guaranteed by including in the model the following constraints:

```
forall(c in Courses)
   take[c] = (term[c] > 0);
```

This last set of constraints simply means that a course should be marked as being selected for the program if and only if it is scheduled to be taken in some term during the planning horizon. It is based on the convention that an expression such as `(term[c] > 0)` is evaluated to 1 if is true and to 0 if it is not true.

Objective Function

The objective is to minimize the time-to-degree, that is, the number of consecutive terms needed to accommodate all the selected courses. For this purpose we define an additional integer variable, `lastTerm`, with possible values in the range `0,1,...,nbTerms` to represent the index of the latest term for which any course is scheduled. Thus, the objective function can simply be stated as:

```
minimize lastTerm
```

To guarantee that no course is scheduled beyond `lastTerm` the model must include the following set of constraints:

```
forall(c in Courses)
   term[c] <= lastTerm;
```

Constraints

In addition to the constraints discussed above, the model must include constraints to ensure that the plans they produce satisfy the academic requirements, that is, the two course requirements, all the prerequisite requirements and the per-term course-load limits. These constraints will now be discussed in order.

Course requirements. Let `ReqA = {C4,C5,C6,C7}` and `ReqB = {C8,C9,C10,C11,C12}` be the sets of courses available to satisfy the two course requirements of the program. To satisfy these requirements, the model must include the following constraints:

```
sum(c in ReqA) take[c] >= 3;
sum(c in ReqB) take[c] >= 2;
```

Prerequisite requirements. Let `PrerequisitePairs` be the set of all pairs of courses `<c,p>` such that p is a prerequisite of c. For each of these pairs, the model must ensure that if course c is included in the academic plan, course `p` must also be included and, further, that `p` must be scheduled for a term that precedes the term for which `c` is scheduled. On the other hand, if `c` is not selected, then `p` may still be included (or not included) in the plan and the term for which it is scheduled is not restricted. Constraints programming languages routinely handle such conditional constraints and in the CP model the prerequisite constraints are succinctly recorded as

```
forall(<c,p> in PrerequisitePairs)
   take[c] => 0 < term[p] < term[c];
```

The notation => indicates logical implication.

Course-Load Limits. The number of courses taken in each term may not exceed 3. In the CP model the following set of constraints:

```
forall(t in Terms)
   sum(c in Courses)(term[c] = t) <= 3;
```

The complete CP model is given in Figure 5.

```
minimize lastTerm
subject to {
   forall(c in Courses)
      take[c] = (term[c] > 0);
   forall(c in Courses)
      term[c] <= lastTerm;
   sum(c in ReqA) take[c] >= 3;
   sum(c in ReqB) take[c] >= 2;
   forall(<c,p> in PrerequisitePairs
      take[c] => 0 < term[p] < term[c];
   forall(t in Terms)
      sum(c in Courses)(term[c] = t) <= 3;
};
```

Figure 5: The Complete CP Model.

The CP model can be solved by any standard constraint programming solver but not by a specialized integer programming solver because many of its constraints are not linear equalities or inequalities. We next show how the CP model may be modified to create a corresponding IP model.

Representing the course-load constraints by linear inequalities require that we include in the model, for each course c in Courses and each term t in Terms a binary variable `scheduled[c,t]`defined as follows:

$$\texttt{scheduled[c,t]}=\begin{cases}1 & \texttt{if course c is scheduled for term t}\\ 0 & \texttt{if course c is not selected}\end{cases}$$

With these new variables, the course-load constraints (stating that no more than 3 courses may be taken in each term) can be replaced by the following linear inequalities:

```
forall(t in Terms)
   sum(c in Courses) scheduled[c,t] <= 3;
```

However, to maintain consistency between the new variables and the original variables, the following two set of constraints must be included in the IP model:

```
forall(c in Courses)
   take[c] = sum(t in Terms) scheduled[c,t];
forall(c in Courses)
   term[c] = sum(t in Terms) t*scheduled[c,t];
```

Notice that the inclusion of these (linear) constraints in the model permits the removal of the first set of (non-linear) constraints in the CP model because they also guarantee that a course is marked as being selected if and only if it is scheduled to be taken in some term of the planning horizon.

The representation of the prerequisite requirements requires the use of one of IP's many "formulation tricks." In this case we replace the "natural" but decidedly non-linear constraints of the CP model with the following two sets of linear constraints:

```
forall(<c,p> in PrerequisitePairs)
   take[p] >= take[c];
forall(<c,p> in PrerequisitePairs)
   term[p] - term[c] <= -1 + (nbTerms +1)*(1 - take[c]);
```

The first set of constraints guarantees that if `take[c] = 1` (i.e., `c` is included in the plan) then `take[p] = 1` (i.e., `p` is also included). The second set guarantees that if `take[c] = 1` then the term for which `p` is scheduled must be strictly earlier than the term for which `c` is scheduled. The coefficient `(nbTerms +1)` is sufficiently large so that if `take[c] = 0` the constraint becomes vacuous. The complete IP model is given in Figure 6.

```
minimize lastTerm
subject to {
   forall(c in Courses)
      take[c] = sum(t in Terms) scheduled[c,t];
   forall(c in Courses)
      term[c] = sum(t in Terms) t*scheduled[c,t];
   forall(c in Courses)
      term[c] <= lastTerm;
   sum(c in ReqA) take[c] >= 3;
   sum(c in ReqB) take[c] >= 2;
   forall(<c,p> in PrerequisitePairs)
      take[p] >= take[c];
   forall(<c,p> in PrerequisitePairs)
      term[p] - term[c] <= -1 + (nbTerms +1)*(1 - take[c]);
   forall(t in Terms)
      sum(c in Courses) scheduled[c,t] <= 3;
};
```

Figure 6: The Complete IP Model

As Figures 5 and 6 illustrate, the CP model is a more compact representation of the problem than the IP model and it is also more "natural." due to the fact that it is not restricted to using only linear constraints. This advantage of constraint programming (Van Hentenryck, 2002) becomes more pronounced in "real-life" situations where degree requirements and regulations are far more complex.

The two models developed in this section were implemented using ILOG OPL Studio

(ILOG, 2000). The CP model was solved by the ILOG Solver, which is a general purpose constraint solver, and the IP model was solved by CPLEX, a general purpose integer programming solver. Both confirm that the shortest time to complete this degree program is 3 terms and that completing the program within that time frame requires taking a minimum of 8 courses. They also confirm that while the requirements may be satisfied by as few as 7 courses, doing so would prolong the completion of the program.

Academic Planning Modeling at CSUN

The requirements of “real-life” degree programs are far more complex than those of the made-up example used in the previous section. In this section we describe the requirements the B.A. degree in Child and Adolescent Development (CADV) at the California State University, Northridge (CSUN), one of approximately 50 bachelor’s degree programs offered by the university. We also report on our experience in modeling these requirements. The graduation requirements described in this section were taken from the 2004-2006 CSUN undergraduate/graduate catalogue (CSUN, 2004).

Graduating with a B.A. degree from CSUN requires earning a minimum of 120 units of course credit while satisfying course requirements in three major components of the university’s undergraduate program: A **General Education (GE) program** and the **Title 5 Requirements in American History and Government,** both of which are common to all students, and the **Major Requirements**, which are specific to the student’s selected area of concentration.

The GE Program consists of a minimum of 52 unit of course credit distributed among six areas (also referred to as GE sections) as follows:

Section A: Basic Subjects…………………………………………... 12 units
Section B: Natural Sciences………………………………….…………9 units
Section C: Humanities…………………………………………….……… 9 units
Section D: Social Sciences…………………………...………………9 units
Section E: Applied Arts and Sciences……………………..………… 4 units
Section F: Comparative Cultural Studies………………………………9 units

Each of the sections of the GE Program is further broken into one or more “subsections” which are more specific requirements that must be satisfied in order to satisfy the section. Each of the subsections is associated with list of courses available to satisfy it, as well as the rules that govern the selection, which vary from one section to another. The selection and scheduling of GE courses is further subject to the **Upper-Division General Education Requirement** that mandates that at least 9 units of GE course credit, selected from at least two different GE sections without restriction of subsections, must be in upper-division level courses (300-level and above). These courses may not be taken sooner than the semester in which junior standing (60 units) is achieved by the student. The GE sections and subsections (including the Upper-Division GE requirement), the number of courses listed in each subsection, and the selection rules are shown in Table 3 below.

The Title 5 Requirements in American History and Government, which are prescribed by California law, consists of three requirements: (1) American history, institutions and ideals, (2) the Constitution of the United States, and (3) the principles of state and local government as established in California. The number of courses available for each requirement and their rules are also included in Table 3.

The requirements of the CADV major consist of 42 units of course credit distributed among three groups of courses as follows:

Lower-Division Required Courses…………………..……………….. 13 units

Upper-Division Required Courses………………………………………. 31-33 units
Upper-Division Elective Courses…………….…………………….….... 12 units

The Upper-Division required courses are further divided into four “sections.” The details of the requirement are given in Table 4.

Table 3: General Education Course Requirements at CSUN

Section	Subsection	Course List	Rules
A – Basic Subjects	A.1 – Written Composition	4 courses	1. Select a minimum of 3 units from each subsection 2. All courses must be completed within the first 4 semesters or 60 units, whichever comes first 3. The writing course must be completed no later than the semester in which 45 units are completed 4. The writing course and oral communication course should be taken simultaneously or within two consecutive semesters 5. The critical reasoning course should be taken after completing the mathematics course
	A.2 – Critical Reasoning	7 courses	
	A.3 - Mathematics	8 courses	
	A.4 – Oral Communication	5 courses	
B – Natural Sciences	B.1 – Biological and Physical Sciences	19 courses	1. Select at least 9 units from entire section 2. Select at least one lecture course from each subsection 3. Select at least one lab or field study course associated with one of the lecture courses selected
	B.2 – Earth Sciences and Astronomy	16 courses	
	Available lab and field study courses	18 courses	
C - Humanities	Super List	4 courses	1. Select at least 3 units from each of subsections C.1, C.2, and C.3 2. The "Super List" courses may be applied to any of the three subsections
	C.1 – Literature	25 courses	
	C.2 – Fine Arts	34 courses	
	C.3 – Philosophy and Religion	21 courses	
D – Social Sciences	(not subsections)	61 courses from 14 disciplines	1. Select at least 9 units from entire list 2. The courses selected must be from at least 2 disciplines
E – Applied Arts and Sciences	(not subsections)	85 courses from 23 disciplines	1. Select at least 4 units from entire list 2. The courses selected must be from at least 2 disciplines
F – Comparative Cultural Studies	F.1 – History of Western Civilization	9 courses	1. Select at least 3 units from each of subsections F.1, F.2, and F.3 2. Three units of one foreign language course may be applied to either subsection F.2 or subsection F.3
	F.2 – International Cross-Cultural Studies	53 courses	
	F.3 – Intra-National Cross-Cultural Studies	38 courses	
	F.4 – Foreign Language	56 courses	
Upper-Division General Education Requirement	(not subsections)	199 courses	At least 9 units of the GE coursework, selected from at least two different sections (A through F, without restrictions of subsections), must be in Upper-Division level courses, taken no sooner than the semester in which Junior Standing (60 units) is being achieved
Title V Requirements in American History and Government	(1) – American History, Institutions and Ideas	7 courses	1. Requirement (1) may be satisfied by taking one course in its course list 2. Both requirements (2) and (3) may be satisfied by taking one course listed under requirement (2) 3. POL 403 would satisfy requirement (3) only
	(2) – the Constitution of the United States	5 courses	
	(3) – the Principles of State and Local Government	POLS 403	

Table 4: Course Requirements for the CADV Major

<table>
<tr><th colspan="2">Section</th><th>Subsection</th><th>Course List</th><th>Rules</th></tr>
<tr><td colspan="2">Lower-Division Introductory Courses</td><td>(no subsections)</td><td>4 Courses</td><td>Take all listed Courses</td></tr>
<tr><td rowspan="10">Upper-Division Required Courses</td><td rowspan="2">Concepts and Applications</td><td>Overview of Development</td><td>2 Courses</td><td rowspan="2">Select one course from each subsection</td></tr>
<tr><td>Modes of Inquiry</td><td>3 Courses</td></tr>
<tr><td rowspan="3">Domains of Development</td><td>Cognitive Development</td><td>4 Courses</td><td rowspan="3">1. Select one courses from 2 of the 3 subsections
2. Students must select CADV 350 or CADV 352 as one of these choices</td></tr>
<tr><td>Language Development</td><td>2 Courses</td></tr>
<tr><td>Social Development</td><td>3 Courses</td></tr>
<tr><td rowspan="3">Cultural and Linguistic Contexts</td><td>Required Course</td><td>CADV 460</td><td rowspan="3">1. CADV 460 must be taken
2. Select one course from each of the other subsections</td></tr>
<tr><td>Cultural Contexts</td><td>10 courses</td></tr>
<tr><td>Linguistic Context</td><td>9 courses</td></tr>
<tr><td rowspan="2">Professional Development</td><td>Group A</td><td>3 courses</td><td rowspan="2">1. All the courses in Group A must be taken
2. CADV 394B and CADV 494B are the parts of a two-semester sequence that start in the fall
3. Select one of the courses in Group B</td></tr>
<tr><td>Group B</td><td>2 courses</td></tr>
<tr><td colspan="2" rowspan="6">Upper-Division Elective Courses</td><td>Development</td><td>13 courses</td><td rowspan="6">1. Select any 12 units of coursework from all listed courses (without regard to areas of interests)
2. Courses are only counted once in the major, either as a required course or as an elective but not as both</td></tr>
<tr><td>Atypical Development</td><td>8 courses</td></tr>
<tr><td>Education</td><td>16 courses</td></tr>
<tr><td>Counseling</td><td>12 courses</td></tr>
<tr><td>Culture, Language and Development</td><td>21 courses</td></tr>
<tr><td>Research, Methodology and Assessment</td><td>7 courses</td></tr>
</table>

In addition to representing a much larger planning problem than the example discussed above (involving 500 courses and 30 individual course requirements compared to 12 courses and just two requirements), the requirements for earning a B.A. in CADV at CSUN are more intricate than those of our example:

Course requirements are mostly specified in terms of **number of units of credit** that must be taken to satisfy the requirement (as opposed to the **number of courses**) because courses offered by the university provide varying amounts of credit.

The requirement rules are far more involved than the simple rule used in the example (i.e., select 2 courses from a list of courses). Rules often refer to more than one list (e.g., select one course from two of three course lists) and refer to additional course attributes such as the course's discipline (biology, political science, etc.) and course's level (lower-division vs. upper-division courses).

Required rules at CSUN are concerned not just with **what** courses may be selected but also with **when** should be taken (e.g., courses selected to satisfy the Upper-Division GE requirement may not be taken earlier than the semester when 60 credit units are being achieved).

At CSUN, a course may be available to satisfy more than one requirement and there are complex rules that determine when a course may, in fact, be counted toward more than one requirement and this option is not allowed.

The specification of prerequisites at CSUN's courses catalogue also take forms that are often more general that in our sample degree program. For example, prerequisites are often stated in terms of a conjunction, e.g., the prerequisite is course x and either course y or course z.

These differences make modeling of the academic planning problem at CSUN much more challenging than the one we used as an example. Nevertheless, this problem has been modelled as a constraint programming model as well as an integer programming model. The CP model was to large for the ILOG solver to solve in reasonable time but the IP model was solved by CPLEX within a few minutes of running time. An academic plan generated by the model that shows how the degree may be completed within 8 terms is shown in table 5.

Table 5: An 8-Term Academic Plan for a B.A. in Child and Adolescent Development

Term 1			Term 2		
Course	**Units**	**Requirements/Prerequisites**	**Course**	**Units**	**Requirements/Prerequisites**
AAS155	3	GE-A1	CHIN102C	4	(none)/CHIN201
ART111	3	GE-C2	CHS151	3	GE-A4
CHIN101C	4	(none)/CHIN102C	POLS155	3	T5-Sec2, T5-Sec3
MATH140	4	GE-A3, CADV-Intro/ CADV380, CADV352	FCS330	3	CADV-DevOverview/ CADV460. CADV380, CADV352, CADV350
PHYS161	2	GE-B1	PSY150	3	GE-D, CADV-Intro/PSY310
Total Units	16		**Total Units**	16	
Term 3			**Term 4**		
Course	**Units**	**Requirements/Prerequisites**	**Course**	**Units**	**Requirements/Prerequisites**
CADV380	3	CADV-InqModes/ CADV394B	CADV150	3	CADV-Intro
CHIN201	3	GE-F3	CADV250	3	CADV-Intro
ENGL405	3	CADV-LinguisticCont	CADV352	3	CADV-Elective/CADV394B
GEOL300	3	GE-B2/GEOL301	CD361	3	CADV-Elective
PHIL200	3	GE-A2	PSY310	3	CADV-Elective
Total Units	15		**Total Units**	15	
Term 5			**Term 6**		
Course	**Units**	**Requirements/Prerequisites**	**Course**	**Units**	**Requirements/Prerequisites**
CADV394 B	2	CADV-ProfDevA/ CADV494B	ANTH152	3	GE-D
CADV460	3	CADV-ContextsReq	CADV350	3	CADV-CogDev/CADV470
GEOL301	1	GE-B(lab), GE-UD	CADV494 B	2	CADV-ProfDevA
HIST150	3	GE-F1	GEOL100	2	GE-B2/GEOL102
POLS310	3	GE-D, GE-UD	HIST270	3	T5-Sec1
RS361	3	GE-C3, GE-UD	PAS110	3	GE-E
Total Units	15		**Total Units**	16	
Term 7			**Term 8**		
Course	**Units**	**Requirements/Prerequisites**	**Course**	**Units**	**Requirements/Prerequisites**
CADV470	3	CADV-ProfDevA	CADV452	3	CADV-ProfDevB
CHS432	3	CADV-CultContexts	FLIT370	3	GE-F2, GE-UD
FCS480	3	CADV-Elective	GEOL102	1	GE-B(Lab)
KIN185A	1	GE-E	Free Elect	3	(none)
LING417	3	CADVLanguageDev	Free Elect	3	(none)
PAS245	3	GE-C1	Free Elect	3	(none)
Total Units	16		**Total Units**	16	

Applications and Benefits of the Proposed Modeling Approach

As part of their efforts to improve graduation rates, many higher education institutions have implemented computerized "degree audit" programs that compare, at any desired point in time, the individual student's academic record with the requirements of his degree program to create a report that shows which of the graduation requirements have been satisfied and which of them remain to be satisfied. This information is intended to guide students in developing their academic plans for a more timely completion of the requirements. However, degree audit programs do not provide any guidance regarding which of the available courses for each requirement should be selected or when to take them or how these decisions might effect the time to graduate. They also are incapable of generating academic plans based on the student's circumstances and goals.

Encoding the requirements for a degree as constraints in an optimization or constraint satisfaction model would be a natural extension of the functionality of a degree audit program. As we illustrated in the last section, such a model can quickly produce a complete academic plan that would satisfy all the requirements for a degree as well as other constraints such as limits on the course load that may be taken at one time. Such a model could also be used to update the student's plan based on his progress in the program as information about courses completed successfully simply become new constraints in the model. The model can be used to help the student consider academic planning options before committing to a decision. For example, the model can answer questions like whether taking a particular course during a particular term would extend the student's time to graduation or by how much the student's course load must be increased in order to meet a specific target graduation date.

Conclusion and Suggestions for Further Research

Constraint modeling has been applied to a wide variety of planning and scheduling problems, including industrial shop-floor job scheduling, project planning, and workforce scheduling. This paper presents an initial attempt to apply this approach to personal academic planning. We first explain and motivate the concept with the aid of a small, made-up, example and then validate it on a "real-life" test case representing a typical degree program. As we discuss in the preceding section, this modeling approach has a number of potential benefits and applications, all of which stem from the ability to use the model to find the effect of the requirements as well as the student's decisions on the time it would take to complete the program.

This research could be extended in two distinct but connected directions. First, academic institutions seem to vary widely in how they structure and state their requirements and their educational policies and it would be useful to get a broader perspective of existing practices and standards in order to better assess the usefulness of the proposed modeling approach. Graduation requirements and institutional rules from more universities should be analyzed and academic planning models should be developed and tested for them. A desirable outcome of such examination would be a set of generic test cases which would be representative of a large number of institutions and on which different modeling approaches and solution techniques may be evaluated and compared. Another possible outcome of such research would be a better understanding of what makes one set of graduation requirements and academic policies better than others from the point of making the process of academic planning more tractable and easier to satisfy in a timely manner.

The second direction of research would involve improving the computational performance of the models. For the proposed modeling approach to be practical, solving the model should take on average more than a few seconds. As we reported above, the CP model developed for our case study proved to be too complex for the ILOG solver, which in for most inquiries did not return a solution at all, and solving the corresponding IP model by CPLEX took several minutes. The scheduling problems that result form applying the proposed approach are large (i.e., they involve a large number of variables and constraints) and belong to a class of problems that are notoriously hard. Nevertheless, reducing the solution time can often be achieved by reformulating the model and/or by customizing the solution procedure to the specific characteristics and structure of the problem at hand.

References

Astin, A. W., & Oseguera, L. (2005). *Degree Attainment Rates at American Colleges and Universities (revised edition)*. Los Angeles, CA: Higher Education Research Institute, UCLA.

Baptiste, P., Le Pape, C., & Nuijten, W. (2001). *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*. Boston, MA: Kluwer Academic Publishers.

Brucker, P., Drexel, A., Mohring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: notation, classification, models, and methods. *European Journal of Operational Research*, 112, 3-41.

Caseau, Y., & Laburthe, F. (1997). A Constraint based approach to the RCPSP. Proceedings of the CP97 Workshop on Industrial Constraint-Directed Scheduling, Schloss Hagenberg, Austria.

CSU (2002). *Facilitating Student Success in Achieving the Baccalaureate Degree*. Report of the California State University task force on facilitating graduation. Available online at <http://www.calstate.edu/AcadSen/Records/Reports/FacilitatingGraduation.pdf>.

CSUN (2004). *California State University, Northridge Undergraduate/Graduate Catalog 2004-2006*. Available online at <http://www.csun.edu/catalog/archive.html>.

ILOG (2000). *ILOG* OPL *Studio 3.0 Optimization Programming Language Reference Manual*.

Kolisch, R., & Padman, R. (2001). An integrated survey of deterministic project scheduling. *OMEGA*, 29, 249-272.

NCES (2006). U.S. Department of Education, National Center for Education Statistics, Integrated Postsecondary Education Data System (IPEDS), 2003-2004 data. <http://nces.ed.gov/ipeds/>, retrieved 20 August, 2006.

Patterson, J. H., & Huber, W. D. (1974). A horizon-varying, zero-one approach to project scheduling. *Management Science*, 20, 990-998.

Stinton, J. P., Davis, E. W., & Khumawala, B. M. (1978). Multiple resource-constrained scheduling using branch-and-bound. *AIIE Transactions*, 10, 252-259.

Williams, T. (2003). The contribution of mathematical modeling to the practice of project management. *IMA Journal of Management Mathematics,* 14, 3-30.

Van Hentenryck, P. (2002). *The OPL Optimization Programming Language*. Cambridge, MA: The MIT Press.

Van Hentenryck, P. (2002). Constraint and integer programming in OPL. *INFORMS Journal on Computing*, 14, 345-372.

Avi Dechter is an Associate Professor in the department of Systems and Operations Management at the College of Business and Economics, California State University, Northridge. He received his Ph.D. in Management Science from UCLA and his research interests include scheduling and planning in service operations.

Correspondence for the author may be sent to Dr. Avi Dechter, College of Business and Economics, California State University, Northridge, 18111 Nordhoff Street, Northridge, CA 91330-8378. E-mail: adechter@csun.edu.