Brahim Hnich   Patrick Prosser   Barbara Smith (Eds.)

Modelling and Reformulating Constraint Satisfaction Problems

Fourth International Workshop
Sitges (Barcelona), Spain, 1 October 2005
Proceedings

Held in conjunction with the
Elventh International Conference on
Principles and Practice of
Constraint Programming (CP 2005)

Optimization Models for Generating Graduation Roadmaps

Avi Dechter[1], Rina Dechter[2]

[1] California State University, Northridge
adechter@csun.edu

[2] University of California, Irvine
dechter@ics.uci.edu

Abstract. Most bachelor's degree granting institutions in the U.S. are "four-year" colleges in name only. The majority of students who enter such colleges take far longer than four years to graduate. In an effort to help students to plan their studies more effectively and reduce time-to-degree, the California State University has introduced "graduation roadmaps," which are sample plans that demonstrate how a particular degree program may be completed within a given amount of time. In this paper we develop integer programming and constraints programming models for facilitating the process of generating graduation roadmaps, which also allow customizing these roadmap for the preferences of individual students.

INTRODUCTION

A recent report ([Carey, 2004]), citing data collected by the U.S. Department of Education's Graduation Rate Survey, shows that only 37% of first-time freshmen entering four-year bachelor's degree programs in American universities and colleges complete their degrees within four years. Only 63% of the students complete their degrees in six years – the more common time frame used to report graduation rates. Graduation rates vary wildly among institutions with overall six-year graduation rates ranging from less than 10% to almost 100%. Understandably, graduation rates and their companion measure of institutional effectiveness – the time to bachelor's degree completion (time-to-degree) – are of major concern to many universities and colleges.

The California State University, where graduation rates have been significantly lower than national averages (although somewhat better than its peer public institutions), has been under pressure to improve its graduation rates. The overall four-year graduation rate at CSU is just 8% and the six-year graduation rate is 40%. A CSU Task Force on Facilitating Graduation, which had been formed to address this problem, considered many policy options that are available to the institution ([CSU, 2002]). Prominent among the task force's recommendations was that the CSU campuses should develop 4-year, 5-year, and 6-year "graduation roadmaps" for all academic degree programs. These roadmaps "should be term-by-term depictions of the courses in which students should enroll over the entirety of their academic careers." The rationale for this recommendation is two-fold: (1) such roadmaps would serve as examples for students to assist them in planning their own individual pathways for graduation, and (2) the development of the roadmaps would require departments to review their curricula and their scheduling practices and to make sure that students can, in fact, graduate on a 4-, 5-, or 6-year timetable.

Pursuing a college degree has many of the characteristics of a project. Like the building of a warehouse or the development of computer software, completing a college degree is a "one-time endeavor with a well-defined end result, defined in terms of identifiable activities that must be accomplished in order to bring the project to completion" [Meredith, 2000]. Enrolling in and successfully completing courses are the activities that define a degree program; course prerequisites are similar to the precedence relationships common among project activities; and the timely completion of the degree, just like most projects, is a primary objective. Table 1 summarizes the parallels between a degree program and a project.

Project	Degree Program
Activities	Courses
Precedence relationships	Prerequisite requirements
Resource limitations	Study-load limits
Goal: minimize completion time	Goal: minimize time-to-degree

Table 1. Parallels between a Project and Degree Program

There are, however, two significant differences between completing a college degree and a "standard" project. First, a standard project is defined in terms of a single set of activities, all of which must be accomplished in order to complete the project. In contrast, a given college degree can be earned by taking many different sets of courses. Second, precedence constraints in projects involve a fixed set of activities all of which must be completed before the activity can start. Course prerequisites can often be met by completing different sets of courses. For these reasons, established project planning and scheduling techniques cannot be used directly for degree planning.

In this paper we address the modeling of the degree planning problem using both integer programming and constraint programming. Integer programming (IP) is an extension of linear programming where variables are restricted to take only integer values. Integer programming has been for several decades an established tool for solving combinatorial optimization problems such as resource-constrained project scheduling problems (e.g., [Patterson, 1974], [Stinson, 1978]). Constraint Programming (CP) is an emergent software technology for declaring and solving constraint satisfaction and constrained optimization problems, including scheduling problems ([Baptiste, 2001]). Unlike integer programming, it is not restricted to linear constraints.

The remainder of the paper is organized as follows. In the next section we discuss the structure of academic degree requirements and, with the help of a small example, we demonstrate some of the unique characteristics of degree planning. We then develop both an IP model and CP model for solving this example and demonstrate the advantages of the constraint programming modeling approach. Lastly, we discuss the potential benefits of using optimization technology for degree planning and suggest future research.

THE STRUCTURE OF DEGREE REQUIREMENTS

The requirements for a degree are typically specified in terms of **required** and **elective** courses. A required course is a course that every student in a degree program must take to qualify for the degree. Typically the required courses are grouped in clusters of related courses, all of which must be taken. For example, the requirements for the BS degree in accountancy at CSUN include a cluster of **required business courses** and a cluster of **required accounting courses**. While usually a required course is a specific course that must taken, students may be offered more than one way to satisfy a particular requirement. For example, the required business courses for the BS in accountancy include a choice between MKT 304 and MGT 360.

Elective courses are typically specified in terms of “baskets” of courses from which the student must select a specified number of **units** (or a given number of courses if all the courses in the basket carry the same number of units). For example, the requirements for the BS in Business Administration (BSBA) with option in Marketing at CSUN included a basket of Option Elective Courses consisting of 14 courses (all are 3 unit courses) from which the student must elect 6 units. As in the case of required courses, a basket of elective courses may include a group of courses only one of which may be used to satisfy the total units required from this basket. For example, the BSBA with option in Supply Chain Management allows selecting either BUS 491 or SOM 498 (but not both) as one of the two courses that have to be elected from a basket of 10 Option Elective Courses.

In addition to the course requirements for a degree, the student’s plan must satisfy all course prerequisites. The prerequisite requirements are designed to ensure that courses are taken in a logical order that promotes the student’s learning. As demonstrated above, prerequisite requirements could be quite detailed and involve not just courses, but also minimum grade requirements and passing of examinations. We will assume, however, that prerequisite requirements for a course specifies sets of courses all of which the student must have completed before enrolling in that course. There is often more than one way to satisfy the prerequisite requirements For example, for MATH 255A the prerequisite requirement is MATH 105 or both MATH 102 and MATH 104. SOM 409, whose prerequisites are given as SOM 306 and either SOM 307 or MATH 340, provides a different example. (Co-requisites for a course are just like prerequisites except that the student may enroll in them contemporaneously with the course.)

A particular course may be part of more than one requirement. For example, ECON 160 (Microeconomics Principles), which is one of the Lower-Division Business Core required courses for all BSBA students at CSUN, is in the basket of elective courses that may be used to satisfy the Social Sciences General Education requirement of the university. Also, a course may be at the same time a prerequisite for another course and a required (or elective) course in a particular degree program. On the other hand, prerequisite requirements may require students to take courses that are not otherwise required by their programs.

To illustrate the problem of degree planning and scheduling, consider the requirements for small, fictitious, degree program as described in Table 2.

Course	Units	Prerequisites Requirements
Prerequisite Courses – Select as needed		
C1	4	None
C2	3	None
C3	2	C4
C4	3	None
Required Courses		
0C5	3	C1
C6 **- or -** C7	4 4	C1 and either C2 or C3 C4
C8	3	C4
Electives – Select at least 6 units from the following		
C9	3	C5 and C10
C10	3	C6
C11 **- or -** C12	3 3	C7, C8 and C10 C13
C13	4	C8
Minimum number of units required for degree – 24		

Table 2. Sample Requirements for Degree

A network representation of the requirements for our fictitious degree is provided in Figure 1. The nodes of the network represent the courses and the directed arcs represent prerequisite requirements in the obvious way. Additional constructs are used to represent the clustering of the courses and the choices available to students in satisfying the various requirements.

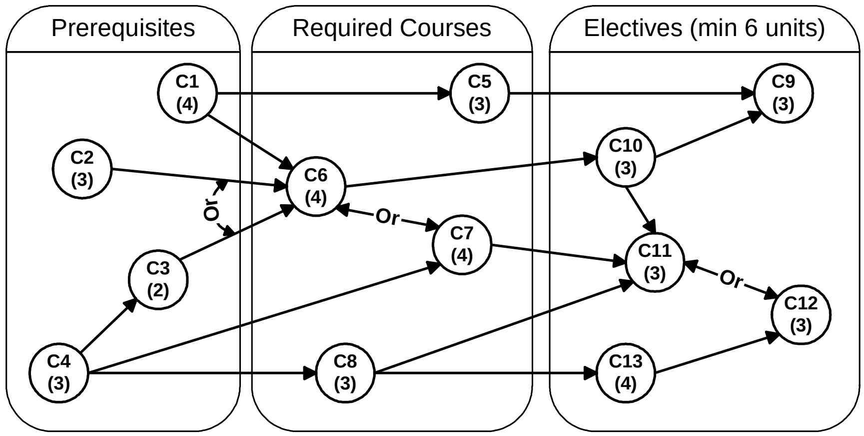

Fig. 1. A Network Representation of Degree Requirements

Degree planning requires making two distinct but interdependent decisions: the selection of a subset of courses to take and the scheduling of the selected courses for specific terms. The selected courses must satisfy the degree requirements. The scheduling of the courses must satisfy additional constraints (study load constraints and perhaps limited availability of courses). We refer to a subset of courses that satisfy the requirement for a degree as a degree plan. The total number of different degree plans in this case is 58. Two of these plans are shown in Figure 2, where the courses selected for each plan are highlighted.

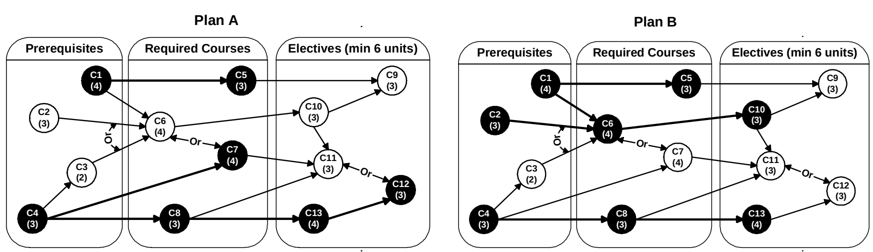

Fig. 2. Two Course Degree Plans for the Example

Plan A calls for taking 7 courses totaling 24 units. Plan B consists of 8 courses totaling 27 units. However, if there are no limits on the number of units a student may take in a single term, plan B may be completed in fewer terms than plan A. This is because the longest path of highlighted nodes (selected courses) in the graph of plan A consists of 4 courses while the longest path in the graph of plan B consists of only 3 courses. A schedule for completing Plan A in its shortest time-to-degree of 4 terms and a schedule for completing Plan B in its shortest time-to-degree of 3 terms are shown in Figure 3.

As this example demonstrates, a smaller program (in terms of the number of units needed to complete a degree) is not necessarily shorter. In fact, Plan A in the only 24 unit plan that satisfies the requirements, which means that in order to minimize the time-to-degree the student must take more than 24 units. For this reason, it would generally not be correct to separate the planning decision from the scheduling decision: they must be considered together. For example, a strategy that would focus on first finding minimum-unit programs is not guaranteed to find plans that are optimal from the time-to-degree standpoint. The problem of simultaneously selecting and scheduling courses so as to minimize the length of the program is more complex to model and solve than standard project scheduling problems where the only issue is that of scheduling activities. When resource constraints are added, complexity increases even further. In the next section we develop both an integer programming model and a constraint programming model for solving the graduation roadmap problem.

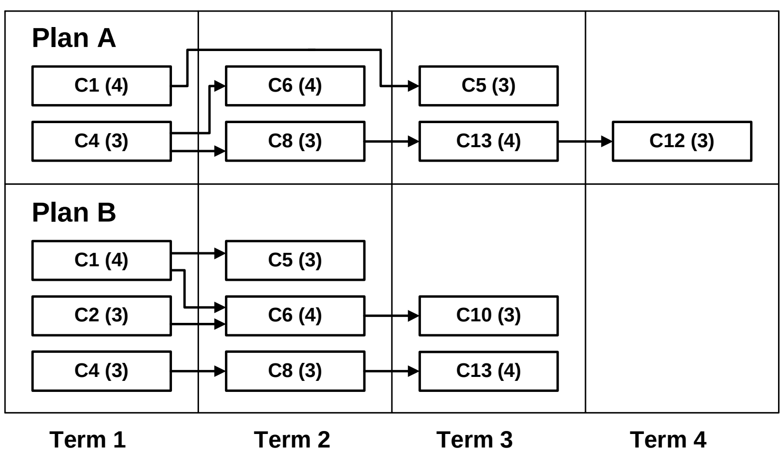

Fig. 3. – Minimum Length Schedules for the Two Plans

Modeling the degree planning problem

Traditionally, resource-constrained project scheduling problems have been modeled as integer programming problems. More recently, constraint programming has been used to model and solve a variety of scheduling problems. It has been claimed (e.g., [Caseau, 1997], [Van Hentenryck, 2002]) that constraint programming offers a more flexible and natural way for expressing "real life" constraints than integer programming. In some cases, constraint programming schemes yield computational advantages as well. In this section we develop, in parallel, an integer programming (IP) model and a constraint programming (CP) model for the fictitious example of the preceding section. By considering both approaches, we wish to determine the effectiveness of each approach to our domain and, at the same time, to provide another real-life benchmark through which the two approaches can be compared and contrasted.

In order to make the example more representative, we assume that no more than 6 units may be scheduled in any term and. Both modeling approaches are similar in that they require the specification of (decision) variables, the cost (objective) function, and the constraints.

Decision Variables

The IP model requires the use of binary (zero-one) decision variables indicating whether or not a particular course is scheduled for a particular term. Defining these variables requires specifying a sufficiently large "planning horizon," T, such that it can be shown that the degree program can be completed in T terms or less. In our example, an easy choice of a value for T is 13, the total number of courses available for selection, which would be the number of terms it would take to complete the program if all the courses are taken and only one course is taken per term. It is easy to show that the degree program can be completed in less than 8 terms so we reduce the value assigned to T to 8. Thus, we define the variables as follows:

$$x_{it} = \begin{cases} 1 & \text{if course } i \text{ is scheduled for term } t \\ 0 & \text{otherwise} \end{cases} \qquad 1 \le i \le 13, 1 \le t \le 8$$

It is convenient to define in the IP model an additional binary variable, y_i, for each of the courses, indicating whether the course is selected for the program:

$$y_i = \begin{cases} 1 & \text{if course } i \text{ is selected for the program} \\ 0 & \text{otherwise} \end{cases} \qquad 1 \le i \le 13$$

The relationships between the two sets of variables require the following constraints:

$$y_i = \sum_{t=1}^{8} x_{it} \qquad 1 \le i \le 13$$

These constraints also guarantee that each course is scheduled to no more that one term.

The reason the IP model must use binary variables it that they enable, as will be shown later, expressing the prerequisite requirements as linear constraints. Since constraint programming is not restricted to the use of linear constrains, it allows a much more natural choice of decision variables that indicate the term to which each course is scheduled. In the CP model we associate with each course *i* a variable s_i defined as follows:

$$s_i = \begin{cases} t & \text{if course } i \text{ is scheduled for term } t,\ t = 1, 2, \ldots 8 \\ 0 & \text{if course } i \text{ is not selected for the program} \end{cases} \qquad 1 \le i \le 13$$

The Objective Function

The objective is to minimize the time-to-degree, that is, the value of the index *t* of the latest term for which any course is scheduled. For this purpose we define, in both modeling approaches, an additional integer variable, *Z*, to represent this value, so that the objective function is simply stated as

$$\text{Minimize } Z$$

To guarantee that no course is scheduled later than *Z*, the IP model must include the following set of constraints:

$$\sum_{t=1}^{8} t x_{it} \le Z \qquad 1 \le i \le 13$$

Similarly, in the CP model the following constraints must be satisfied:

$$s_i \le Z \qquad 1 \le i \le 13$$

The Constraints

Both IP and CP models must include constraints that will ensure that the solutions they produce satisfy all the degree requirements (i.e., required courses, elective courses, prerequisite requirements and minimum unit requirement) as well as term study-load limits. These constraints will now be discussed in order.

Required Courses

Courses C5 and C8, which must be included, require the following constraints in the IP model:

$$y_5 = 1$$
$$y_8 = 1$$

Similarly, in the CP model we need to include

$$s_5 > 0$$
$$s_8 > 0$$

Since either C6 or C7 may be selected to satisfy the remaining requirement, the IP model must include the following constraint:

$$y_6 + y_7 \geq 1$$

In the CP model, which permits logical, as well as mathematical constraints, this requirement may be simply specified as

$$(s_6 > 0) \text{ or } (s_7 > 0)$$

Elective Course Requirement

This requirement has two components: the list of electives from which courses may be selected (in our case, courses C9 through C13) and a minimum number of units that the selected courses must represent (6 units in this example). Courses C11 and C12 may not both be counted toward satisfying the unit requirement. The constraints that would ensure that both courses are counted should not, however, exclude the possibility that both courses are included in the program (perhaps to satisfy other requirements). In the IP model, this can be accomplished by defining two auxiliary variables, y'_{11} and y'_{12} to represent whether C11 and C12, respectively, are used to satisfy the elective requirement. These variables are defined as follows:

$$y'_i = \begin{cases} 1 & \text{if course } i \text{ is used to satisfy the elective requirement} \\ 0 & \text{otherwise} \end{cases} \qquad i \in \{11,12\}$$

Now the elective requirement can be represented in the model by including the following constraints:

$$
\begin{aligned}
&y'_{11} \le y_{11} \\
&y'_{12} \le y_{12} \\
&y'_{11} + y'_{12} \le 1 \\
&3y_9 + 3y_{10} + 3y'_{11} + 3y'_{12} + 3y_{13} \ge 6
\end{aligned}
$$

The first two constraints ensure that courses C11 and C12, respectively, cannot be used to satisfy the elective requirement if they are not included in the program. The third constraint ensures that at most one of then is used to satisfy the requirement, and the last constraint takes care of the minimum units that need to be selected.

Representing the elective course requirement in the CP model does not require any auxiliary variables and can be accomplished by the following constraint:

$$3(s_9 > 0) + 3(s_{10} > 0) + \max\left(3(s_{11} > 0), 3(s_{12} > 0)\right) + 3(s_{13} > 0) \ge 6$$

This constraint requires an explanation. In constraint programming when an expression is true, it given a value of 1 and if it is false it evaluated to 0. Thus, the expression $(s_9 > 0)$ has a value of 1 if course C9 is included in the schedule. The expression $\max\left(3(s_{11} > 0), 3(s_{12} > 0)\right)$ makes sure that, if both C11 and C12 are included in the schedule, the units of just one of them will be counted toward satisfying the requirement.

Prerequisite Requirements

The prerequisite requirements for a course have to be satisfied only if that course is included in the degree plan. For example, since C1 is a prerequisite of C5, a **conditional constraint** must be added, of the form "if C5 is scheduled for period t_5, then C1 must be scheduled for a period $t_1 < t_5$." In the IP model this would achieved by adding the following two constraints to the problem:

$$
\begin{aligned}
&y_1 \ge y_5 \\
&\sum_{t=1}^{8} t x_{1t} - \sum_{t=1}^{8} (t-1) x_{5t} \le 9(1 - y_5)
\end{aligned}
$$

The first constraint guarantees that if C5 is included in the plan, C1 must also be included. The second constraint guarantees that if C5 is included in the plan, then the term for which C1 is scheduled must be earlier than the term for which C5 is scheduled. The coefficient 9 (that is, the planning horizon +1) is sufficiently large so that if $y_5 = 0$ the constraint becomes redundant.

In the CP model, the constraint that would satisfy the relationship between C5 and C1 can simply be written as:

$$\left(s_5 > 0\right) \Rightarrow \left(0 < s_1 < s_5\right)$$

where the symbol $\Rightarrow$ indicates logical implication. Similar constraints are needed, in both the IP and the CP models, for all pairs of courses between which there is a simple prerequisite relationship like that between C5 and C1.

The case of course C6, where either C2 or C3 would satisfy a prerequisite requirement, is somewhat more involved. Constraints are needed to ensure that if C6 is selected for the program, then at least one of the courses C2 and C3 are scheduled for an earlier term. These constraints, however, should not go so far as to require that both C2 and C3 satisfy this condition. In the IP model, two auxiliary variables, $y''_{6,2}$ and $y''_{6,3}$, are needed to represent whether C2 and C3, respectively, satisfy this prerequisite requirement for C6. These variables are defined as follows:

$$y''_{6,j} = \begin{cases} 1 & \text{if course } j \text{ satisfies the prerequisite requirement of C2} \\ 0 & \text{otherwise} \end{cases} \qquad J \in \{2,3\}$$

Now the elective requirement can be represented in the model by including the following constraints:

$$\begin{aligned}
& y''_{6,2} \le y_2 \\
& y''_{6,3} \le y_3 \\
& y''_{6,2} + y''_{6,3} \ge y_6 \\
& \sum_{t=1}^{8} t x_{2,t} - \sum_{t=1}^{8} (t-1) x_{6,t} \le 9\left(1 - y''_{6,2}\right) \\
& \sum_{t=1}^{8} t x_{3,t} - \sum_{t=1}^{8} (t-1) x_{6,t} \le 9\left(1 - y''_{6,3}\right)
\end{aligned}$$

The first two constraints indicate that courses C2 and C3, respectively, cannot satisfy the prerequisite requirement for C6 if they are not included in the program. The third constraint ensures that at least one of them satisfies the prerequisite requirement, and the last two constraints make sure that for C2 (respectively C3) to satisfy the requirement it must be scheduled for a term that is earlier than the term for which C6 is scheduled.

As before, representing this requirement in a CP model is far simpler. It does not require any auxiliary variables and can be accomplished with a single constraint:

$$\left(s_6 > 0\right) \Rightarrow \left(0 < s_2 < s_6\right) \text{ or } \left(0 < s_3 < s_6\right)$$

Minimum Unit Requirement

The requirement that a student must take at least 24 units to earn the degree is represented in the IP model by the following constraint:

$$\sum_{i=1}^{13} u_i y_i \geq 24$$

where u_i is the number of units of each course *i*.

Similarly, in the CP model we need

$$\sum_{i=1}^{13} u_i \left(s_i > 0\right) \geq 24$$

Study-Load Limits

The number of units taken in each term may not exceed 6. In the IP model the following set of constraints:

$$\sum_{i=1}^{13} u_i x_{it} \leq 6 \qquad 1 \leq t \leq 8$$

Similarly, in the CP model we need the constraints:

$$\sum_{i=1}^{13} u_i \left(s_i = t\right) \leq 6 \qquad 1 \leq t \leq 8$$

The complete CP model and IP model for this problem is given in Figure 4 and Figure 5, respectively.

Minimize Z

subject to

$$s_i \le Z \qquad 1 \le i \le 13$$

$$(s_5 > 0) \text{ and } ((s_6 > 0) \text{ or } (s_7 > 0)) \text{ and } (s_8 > 0)$$

$$3(s_9 > 0) + 3(s_{10} > 0) + \max(3(s_{11} > 0), 3(s_{12} > 0)) + 3(s_{13} > 0) \ge 6$$

$$(s_i > 0) \Rightarrow (0 < s_j < s_i)$$

$$\forall < i, j > \text{ s.t. } j \text{ is a unique prerequisite of } i$$

$$(s_6 > 0) \Rightarrow (0 < s_2 < s_6) \text{ or } (0 < s_3 < s_6)$$

$$\sum_{i=1}^{13} u_i (s_i > 0) \ge 24$$

$$\sum_{i=1}^{13} u_i (s_i = t) \le 6 \qquad 1 \le t \le T$$

$$s_i \in \{0,1,2,\ldots,8\} \qquad 1 \le i \le N$$

Fig. 4. The CP Model for the Example

Minimize Z

subject to

$$y_i = \sum_{t=1}^{8} x_{it} \qquad 1 \le i \le 13$$

$$\sum_{t=1}^{8} t x_{it} \le Z \qquad 1 \le i \le 13$$

$$y_5 = 1;\ y_8 = 1;\ y_6 + y_7 \ge 1$$

$$y'_{11} \le y_{11};\ y'_{12} \le y_{12};\ y'_{11} + y'_{12} \le 1$$

$$3y_9 + 3y_{10} + 3y'_{11} + 3y'_{12} + 3y_{13} \ge 6$$

$$y_j \ge y_i \qquad \forall < i, j > \text{ s.t. } j \text{ is a unique prerequisite of } i$$

$$\sum_{t=1}^{8} t x_{jt} - \sum_{t=1}^{8} (t-1) x_{it} \le 9(1 - y_i)$$

$$\forall < i, j > \text{ s.t. } j \text{ is a unique prerequisite of } i$$

$$y''_{6,2} \le y_2;\ y''_{6,3} \le y_3;\ y''_{6,2} + y''_{6,3} \ge y_6$$

$$\sum_{t=1}^{8} t x_{2,t} - \sum_{t=1}^{8} (t-1) x_{6,t} \le 9(1 - y''_{6,2})$$

$$\sum_{t=1}^{8} t x_{3,t} - \sum_{t=1}^{8} (t-1) x_{6,t} \le 9(1 - y''_{6,3})$$

$$\sum_{i=1}^{13} u_i y_i \ge 24$$

$$\sum_{i=1}^{13} u_i x_{it} \le 6 \qquad 1 \le t \le 8$$

$$x_{it} \in \{0,1\} \qquad 1 \le i \le 13, 1 \le t \le 8$$

$$y_i \in \{0,1\} \qquad 1 \le i \le 13$$

$$y'_{11}, y'_{12}, y''_{6,2}, y''_{6,3} \in \{0,1\}$$

Fig. 5. The IP Model for the Example

As Figures 4 and 5 illustrate, the CP model is a more compact representation of the problem than the IP model. This is due to the fact that the CP model, in contrast with the IP model, is not restricted to linear constraints. Some of the variables and constraints in the IP models represent “formulation tricks” whose only role is to enable the representation of the degree requirements as linear inequalities. The logic-based constraints of the CP model represent these requirements in much more “natural” way. This advantage of constraint programming ([Van Hentenryck, 2002]) becomes more significant in “real-life” situation where degree requirements can be very complex, especially if we wish the user (e.g., the student) to incorporate some of his/her specific constraints into the model.

We used OPL Studio to solve the two models developed in this section. IP models are solved by CPLEX and CP models are submitted to ILOG Solver for solution. Both show that a minimum of five terms are needed to complete the degree program. There are seven different plans that would result in completing the degree within that time frame. One of then (which requires the smallest number of units – 25) is shown in Figure 6.

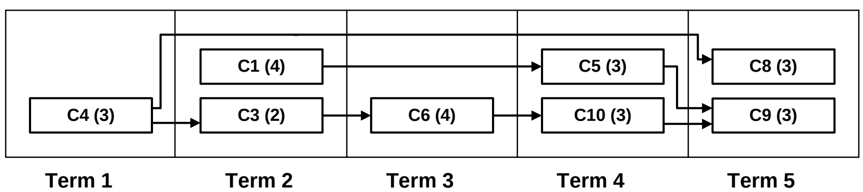

Fig. 6. A Minimum-Length Degree Plan

Conclusion and suggestions for further research

Completing the requirements for a college degree within a reasonable length of time is a daunting task for many students. Complex degree requirements and rules, the freedom afforded to students to choose from a large number of courses, and the need to satisfy prerequisite requirements for these courses, make it often difficult for students to plan their individual programs in a way that would reduce, if not minimize, their time-to-degree. In this paper we point to the similarities, as well as the differences, between degree planning and resource-constrained project planning and argue that degree planning could benefit from models similar to those used for project planning and scheduling.

With the help of a small example that, nevertheless, illustrates the essence and some of the complexities of the degree-planning problem, we show how the problems may be modeled and solved by two different approaches. Integer programming is the more traditional, operations research based, approach, which depends on “formulation tricks" to overcome the need to represent all the constraints in terms of linear equalities and inequalities. Constraint programming is a newer, computer science based, methodology, which allows a more straightforward and "natural" representation of the problem. Further study, currently underway, is needed to assess the practicability of such models for "real life" degree program requirements and to compare computationally the performance of the two approaches. Preliminary experiments show that both models generate optimal plans for real degree programs taken from the CSUN catalogue within acceptable amount of time. The computational aspects of these models, however, are not addressed in this paper and will be studied in the future.

Universities and perhaps other educational institutions may use models like those developed in this paper in several possible ways. They can be used to generate a large variety of "graduation roadmaps" of the type currently required by the California State University for all its degree programs. They can also be used as a basis for automated advisement systems that will allow students to develop and update their own, individual, degree plans. Finally, these models may be very useful for developers of new curricula who could use them to assess the expected time-to-degree of new and modified degree programs.

The models we have proposed in this paper take into account the effects of degree requirements, course prerequisites, and study-load limits on the time-to-degree. Implicitly, we have assumed that all courses are available for the student in each term. This assumption is often unjustified because courses may not be offered in each term and, even when they are offered, they may not have enough room to accommodate all the students who may wish to take them. Since course availability is determined by the course schedule, an interesting subject for further study would be investigating how degree planning models may be used to improve course scheduling decisions.

References

Baptiste P., C. Le Pape and W. Nuijten, 2001, Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems, Kluwer Academic Publishers: Boston

Carey, K., 2004, 'A Matter of Degree: Improving Graduation Rates in Four-Year Colleges and Universities,' A Report by the Education Trust

Caseau, Y. and F. Laburthe, 1997,'A Constraint Based Approach to the RCPSP,' Proceedings of the CP97 Workshop on Industrial Constraint-Directed Scheduling, Schloss Hagenberg, Austria.

California State University, 2002, 'Facilitating Student Success in Achieving the Baccalaureate Degree,' Report of the California State University Task Force on Facilitating Graduation.

California State University, Northridge, 2003, Graduation Rates Task Force Report

ILOG, 2000, ILOG OPL Optimization Programming Language Reference Manual.

Meredith J.R. and S.J. Mantel, Jr., 2000, Project Management: A Managerial Approach, John Wiley Sons: New York.

Patterson, J.H. and W.D. Huber, 1974, 'A Horizon-Varying, Zero-One Approach to project Scheduling,' *Management Science*, Vol.20, No.6, pp.990-998.

Stinson, J.P., E.W. Davis and B.M. Khumawala, 1978, 'Multiple Resource-Constrained Scheduling Using Branch-and-Bound,' *AIIE Transactions*, Vol.10, No.3, pp. 252-259.

Van Hentenryck, P., 2002, 'Constraint and Integer Programming in OPL,' *INFORMS Journal on Computing*, Vol.14, No.4, pp.345-372.