Some Results on The Design of Balanced Academic Curricula*

Carlos Castro
Carlos.Castro@inf.utfsm.cl

and

Eric Monfroy
Eric.Monfroy@inf.utfsm.cl

Departamento de Informática, Universidad Técnica Federico Santa María
Valparaíso, Chile

Abstract

In this paper, we present a mathematical programming model for designing balanced academic curricula. The solution to this model gives a curriculum where the academic load of each period is as similar as possible. Based on the use of several very efficient solving techniques, in the last years we have solved real-life instances that show the applicability of this approach.

Keywords: Academic Curriculum Design, Combinatorial Optimization.

1 Introduction

When designing an academic curriculum a lot of factors are taken into account. Based on the objectives defined for the career under consideration, experts have to propose the courses covering the fundamentals in each domain. Roughly speaking, each domain is represented by a set of courses and some precedence relationships are also imposed among them. Indeed, the academic load of each course, representing the amount of effort required to successfully follow the course, is expressed in some way (generally called credit). Some explicit restrictions can also be imposed when developing a curriculum. For example, number of academic periods, maximum/minumum number of courses per period, and maximum/minumum academic load per period. Once this information is available, the assignment of courses to each academic period is carried out using a trail and error approach until an adequate curriculum is reached.

Assuming that a balanced academic load favors academic habits and facilitates the success of students, we are interested in designing balanced academic curricula. Considering that the academic load of each

*The first author has been partially supported by the National Science and Technology Fund of Chile, FONDECYT, under grant No. 1070268.

period is given by the sum of the credits of each course taken in a period, the problem of designing balanced academic curricula consists in assigning courses to periods in such a way that the academic load of each period will be balanced, i.e., as similar as possible. In order to obtain such a balanced curriculum we think that a quantitative approach, instead of the traditional trail and error approach, can be very useful.

In the last years, we have solved several real-life instances using complete solving techniques, incomplete solving techniques, as well as some schemes of cooperation among these techniques. Indeed, this problem has been used for other authors to show the interest of using reformulation when solving hard combinatorial problems.

This paper is organized as follows: section 2 describes the problem we are interested in. Section 3 presents an integer programming model for the balanced academic curriculum problem. In order to better explain this model, we present, in section 4, a very simple example. In section 5, we present the solution to this example and we also describe three real cases that we have solved using several modern optimization techniques. Finally, in section 6, we conclude the paper and give some perspectives for further works.

2 The Balanced Academic Curriculum Problem

As a general framework we consider administrative as well as academic regulations of the Technical University Federico Santa María.

Academic Curriculum An academic curriculum is defined by a set of courses and a set of precedence relationships among them.

Number of periods Courses must be assigned within a maximum number of academic periods.

Academic load Each course has associated a number of credits or units that represent the academic effort required to successfully follow it.

Prerequisites Some courses can have other courses as prerequisites.

Minimum academic load A minimum amount of academic credits per period is required to consider a student as full time.

Maximum academic load A maximum amount of academic credits per period is allowed in order to avoid overload.

Minimum number of courses A minimum number of courses per period is required to consider a student as full time.

Maximum number of courses A maximum number of courses per period is allowed in order to avoid overload.

In this work, we concentrate on three particular instances of the balanced academic curriculum problem: the three Informatics careers offered by the Technical University Federico Santa María.

3 Integer Programming Model

In this section, we present an integer programming model for the balanced academic curriculum problem. Classical examples and very good explanations on the subject of modeling in integer programming can be found in [12].

- Parameters

  Let

  m : Number of courses
  n : Number of academic periods
  α_i : Number of credits of course i; $\forall i = 1, \ldots, m$
  β : Minimum academic load allowed per period
  γ : Maximum academic load allowed per period
  δ : Minimum amount of courses per period
  ϵ : Maximum amount of courses per period

- Decision variables

  Let

$$x_{ij} = \begin{cases} 1 & \text{if course } i \text{ is assigned to period } j; \forall i = 1, \ldots, m \; \forall j = 1, \ldots, n \\ 0 & \text{otherwise} \end{cases}$$

  c_j: academic load of period j; $\forall j = 1, \ldots, n$
  c: maximum academic load for all periods

- Objective function

$$Min \;\; c = Max\{c_1, \ldots, c_n\}$$

- Constraints

  - The academic load of period j is defined by:

$$c_j = \sum_{i=1}^{m} \alpha_i \times x_{ij} \; \forall j = 1, \ldots, n$$

  - All courses i must be assigned to some period j:

$$\sum_{j=1}^{n} x_{ij} = 1 \; \forall i = 1, \ldots, m$$

  - Course b has course a as prerequisite:

$$x_{bj} \leq \sum_{r=1}^{j-1} x_{ar} \; \forall j = 2, \ldots, n$$

– The maximum academic load is defined by:

$$c = Max\{c_1, \ldots, c_n\}$$

This can be represented by the following set of linear constraints:

$$c_j \leq c \; \forall j = 1, \ldots, n$$

– The academic load of period j must be greater than or equal to the minimim required:

$$c_j \geq \beta \; \forall j = 1, \ldots, n$$

– The academic load of period j must be less than or equal to the maximum allowed:

$$c_j \leq \gamma \; \forall j = 1, \ldots, n$$

– The number of courses of period j must be greater than or equal to the minimum allowed:

$$\sum_{i=1}^{m} x_{ij} \geq \delta \; \forall j = 1, \ldots, n$$

– The number of courses of period j must be less than or equal to the maximum allowed:

$$\sum_{i=1}^{m} x_{ij} \leq \epsilon \; \forall j = 1, \ldots, n$$

4 Example

In order to better explain the general model developed in section 3, we now present a reduced version taken from an informatics career offered by the Technical University Federico Santa María. Table 1 presents 18 courses that are assigned to 4 academic periods, each course identified by its code, and it also presents the number of credits and the precedence relationships among them.

Academic period	Code	Credits	Req1	Req2
01	DEW100	1		
01	FIS100	3		
01	HCW310	1		
01	MAT190	4		
01	MAT192	4		
	Total	13		
02	FIS101	5	FIS100	MAT192
02	IWI131	3		
02	MAT191	4	MAT190	
02	MAT193	4	MAT190	MAT192
	Total	16		
03	FIS102	5	FIS101	MAT193
03	HW1	1		
03	IEI134	3	IWI131	
03	IEI141	3	IWI131	
03	MAT194	4	MAT191	MAT193
	Total	16		
04	DEW 0	2	DEW101	
04	HCW311	2	HCW310	
04	IEI132	3	IEI134	
04	IEI133	3	IEI134	
	Total	10		
	Total	55		

Table 1: A reduced academic curriculum.

Considering a maximum of 16 credits per period, a minimum of 3 credits per period, a maximum of 6 courses per period, and a minimum of 1 course per period, we now develop the integer programming model for this career.

- Parameters

  Let

  m : 18 (number of courses)
  n : 4 (number of academic periods)
  α_1 : 1 (number of credits of DEW100)
  α_2 : 3 (number of credits of FIS100)
  α_3 : 1 (number of credits of HCW310)
  α_4 : 4 (number of credits of MAT190)
  α_5 : 4 (number of credits of MAT192)
  α_6 : 5 (number of credits of FIS101)
  α_7 : 3 (number of credits of IWI131)
  α_8 : 4 (number of credits of MAT191)
  α_9 : 4 (number of credits of MAT193)
  α_{10} : 5 (number of credits of FIS102)
  α_{11} : 1 (number of credits of HW1)
  α_{12} : 3 (number of credits of IEI134)
  α_{13} : 3 (number of credits of IEI141)
  α_{14} : 4 (number of credits of MAT194)
  α_{15} : 2 (number of credits of DEW0)
  α_{16} : 2 (number of credits of HCW311)
  α_{17} : 3 (number of credits of IEI132)
  α_{18} : 3 (number of credits of IEI133)
  β : 3 (minimum academic load allowed per period)
  γ : 16 (maximum academic load allowed per period)
  δ : 1 (minimum amount of courses per period)
  ϵ : 6 (maximum amount of courses per period)

- Decision variables

  Let

  x_{ij} : Assignment variables, where

  $$x_{ij} = \begin{cases} 1 & \text{if course } i \text{ is assigned to} \\ & \text{academic period } j; \forall i = 1, \ldots, 18 \ \forall j = 1, \ldots, 4 \\ 0 & \text{otherwise} \end{cases}$$

  c_j : academic load of period $j; \forall j = 1, \ldots, 4$
  c : maximum academic load for all periods

- Objective function

$$Min \quad c = Max\,(c_1, c_2, c_3, c_4)$$

- Constraints

$$
\begin{array}{lll}
\sum_{j=1}^{4} x_{ij} = 1 & \forall i = 1,\ldots,18 & \text{(all courses must be assigned)} \\
\sum_{i=1}^{18} x_{ij} \geq \delta & \forall j = 1,\ldots,4 & \text{(minimum number of courses per period)} \\
\sum_{i=1}^{18} x_{ij} \leq \epsilon & \forall j = 1,\ldots,4 & \text{(maximum number of courses per period)} \\
c_j \leq \gamma & \forall j = 1,\ldots,4 & \text{(maximum number of credits per period)} \\
c_j \geq \beta & \forall j = 1,\ldots,4 & \text{(minimum number of credits per period)} \\
c_j = \sum_{i=1}^{18} \alpha_i \times x_{ij} & \forall j = 1,\ldots,4 & \text{(academic load of each period)} \\
c_j \leq c & \forall j = 1,\ldots,4 & \text{(maximum academic load)} \\
x_{bj} \leq \sum_{r=1}^{j-1} x_{ar} & \forall j = 2,\ldots,4 & \text{(prerequisites between courses)}
\end{array}
$$

Of course, constraints concerning minimum and maximum number of courses and credits per period are satisfied by the curriculum under consideration, so this assignment represents a possible (but maybe not optimal) solution to the model. This solution is represented by the following values for the decision variables:

Course assignment (x_{ij})																		
i/j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

We now explain how constraints are satisfied by this solution.

- All courses i must be assigned to some period j:

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \forall i = 1,\ldots,m$$

This is verified by the sum of the elements in each column of the following assignment matrix:

Course assignment (x_{ij})																		
i/j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
$\sum$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

- The number of courses of period j must be greater than or equal to the minimum allowed:

$$\sum_{i=1}^{m} x_{ij} \geq \delta \qquad \forall j = 1, \ldots, n$$

This is verified by the sum of the elements in each row of the following assignment matrix:

i/j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	$\sum$	max	min
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	6	1
2	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	4	6	1
3	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	6	6	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	4	6	1

- The academic load of period j must be greater than or equal to the minimim required:

$$c_j \geq \beta \qquad \forall j = 1, \ldots, 4$$

The total number of credits per period is obtained by the multiplication of the assignment matrix and the vector containing the number of credits of each course ($\{\vec{x_{ij}}\} \cdot \vec{\alpha}$):

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 3 \\ 1 \\ 4 \\ 4 \\ 5 \\ 3 \\ 4 \\ 4 \\ 5 \\ 1 \\ 3 \\ 3 \\ 4 \\ 2 \\ 2 \\ 3 \\ 3 \end{pmatrix} = \begin{pmatrix} 13 \\ 16 \\ 16 \\ 10 \end{pmatrix}$$

These sums also verify the minimum academic load per period allowed $\beta = 3$.

- The maximum academic load is defined by:

$$c = Max\,(c_1, \ldots, c_4)$$

This can be representd by the following linear constraints:

$$c_j \leq c \qquad \forall j = 1, \ldots, 4$$

Academic load per period (c_i)		
i	c_i	$c = Max\,(c_1, c_2, c_3, c_4)$
1	13	16
2	16	16
3	16	16
4	10	16

- Course b has course a as prerequisite:

$$x_{bj} \leq \sum_{r=1}^{j-1} x_{ar} \qquad \forall j = 2, \ldots, n$$

For example, considering the precedence relationship between course MAT191 and course MAT190:

$$x_{82} \leq x_{41}$$
$$x_{83} \leq x_{41} + x_{42}$$
$$x_{83} \leq x_{41} + x_{42} + x_{43}$$

5 Experimental Results

Solving the reduced version of the balanced academic curriculum problem, presented in section 4, we obtain the optimal solution presented in table 2. This solution was obtained using `lp_solve` [1], a software for solving integer linear programming models. Details about techniques for solving integer programming models can be found in [8].

As real-life instances we have considered the three informatics careers at the UTFSM, involving 8, 10, and 12 academic periods, and 36, 42, and 62 courses. This means, 36×8, 42×10, and 62×12 binary variables, respectively. A first attempt to solve these problems using integer programming techniques was done several years ago on the curriculum of the four-year career involving [11]. Thanks to the improvement of these techniques in the last years, nowadays we can solve all these instances.

Considering the success of constraint programming [5] for solving combinatorial search problems, we were also interested in using this technique. In fact, using the Oz [2] language [10], that implements this kind of technique, we were able to get the optimal solution for the three cases under consideration [1].

We have also implemented a Genetic Algorithm [6] to apply the incomplete approach. We have obtained very good solutions in a very short time, but still not the optimal solutions obtained by the complete techniques.

Finally, we have also applied hybrid algorithms to solve these instances obtaining very promising results [3, 2, 4].

[1] tech.groups.yahoo.com/group/lp_solve
[2] www.mozart-oz.org

Academic period	Code	Credits	Req1	Req2
01	FIS100	3		
01	MAT190	4		
01	MAT192	4		
01	IWI131	3		
	Total	14		
02	DEW100	1		
02	FIS101	5	FIS100	MAT192
02	MAT191	4	MAT190	
02	MAT193	4	MAT190	MAT192
	Total	14		
03	IEI134	3	IWI131	
03	IEI141	3	IWI131	
03	MAT194	4	MAT191	MAT193
03	DEW 0	2	DEW101	
03	HW1	1		
03	HCW310	1		
	Total	14		
04	FIS102	5	FIS101	MAT193
04	IEI132	3	IEI134	
04	IEI133	3	IEI134	
04	HCW311	2	HCW310	
	Total	13		
	Total	55		

Table 2: Optimal solution for the reduced academic curriculum.

6 Conclusions

We have presented a quantitative approach for designing balanced academic curricula. Using solving techniques implemented in software available for free we have been able to solve very efficiently all instances we have considered in this work. Of course, more work could be done to extend the model presented in this paper. For example, we could consider several careers at the same time in order to assign common courses to the same academic period. We strongly believe that this kind of quantitative approach should be used in order to support several decision problems that arise in the educational field. Many fruitfull experiences have already been carried out for solving, for example, timetabling problems using a similar approach [9, 7].

References

[1] Castro, C., and Manzano, S. Variable and Value Ordering When Solving Balanced Academic Curriculum Problems. In *Proceedings of The Sixth Workshop of the Working Group on Constraints, ERCIM2001* (Prague, Czech Republic, June 2001).

[2] Lambert, T., Castro, C., Monfroy, E., aría Cristina Riff, M., and Saubion, F. Hybridization of Genetic Algorithms and Constraint P ropagation for the BACP. In *Twenty First International Conference on Logic Progr amming, ICLP 2005* (Sitges, Barcelona, Spain, Oct. 2005), M. Gabbrielli and G. Gupta, Eds., vol. 3668 of *Lectures Notes in Computer Science*, Springer-Verlag, pp. 421–423.

[3] Lambert, T., Castro, C., Monfroy, E., aría Cristina Riff, M., and Saubion, F. Hybridization of Genetic Algorithms and Cosntraint Propagation: application to the Balanced Academic Curriculum Problem. In *Fifth Workshop on Cooperative Solvers in Constraint Programming, COSOLV 2005* (Sitges, Barcelona, Spain, Oct. 2005).

[4] Lambert, T., Castro, C., Monfroy, E., and rédéric Saubion, F. Solving the Balanced Academic Curriculum Problem wit h an Hybridization of Genetic Algorithm and Constraint Propagation. In *The Eighth International Conference on Artificial In telligence and Soft Computing, ICAISC 2006* (Zakopane, Poland, June 2006), L. Rutkowski, R. Tadeusiewics, L. A. . Zadeh, and J. Zurada, Eds., vol. 4029 of *Lectures Notes in Artificial Intelligence*, Springer-Verlag, pp. 410–419.

[5] Marriott, K., and Stuckey, P. *Programming With Constraints: An Introduction.* Mit Press, 1999.

[6] Michalewicz, Z., and Fogel, D. *How to Solve It: Modern Heuristics.* Springer-Verlag, 1999.

[7] Mueller, T., and Barták, R. Interactive Timetabling. In *Proceedings of The Sixth Workshop of the Working Group on Constraints, ERCIM2001* (Prague, Czech Republic, June 2001).

[8] Nemhauser, G. L., and Wolsey, L. A. *Integer and Combinatorial Optimization.* Wiley-Interscience Series in Discrete Mathematics. Wiley, 1999.

[9] Schaerf, A. A survey of automated timetabling. Tech. Rep. CS-R9567, Centrum voor Wiskunde en Informatica, CWI, Amsterdam, The Netherlands, 1996.

[10] Smolka, G. The Oz Programming Model. In *Computer Science Today*, J. van Leeuwen, Ed., vol. 1000 of *LNCS*. Springer-Verlag, 1995, pp. 324–343.

[11] Vergara, C. Desarrollo de un Modelo de Programación Lineal para la Generación de Mallas Curriculares Balanceadas. Memoria de IEI, UTFSM, Valparaíso, Chile, Dec. 1994.

[12] Williams, H. *Model Building in Mathematical Programming*, 4 ed. J. Wiley and Sons, 1999.