Project: Integer Programming vs. Expert Systems: An Experimental Comparison

Maxime CHAMBREUIL
McGill ID: 260067572
maxime.chambreuil@mail.mcgill.ca

Contents

1 Introduction

This report is based on the article, written in 1990 by Vasant Dhar an Nick Ranganathan, from the Leonard N. School of Business, New York University. The goal of their paper was to compare an in-production Expert System to assign teachers to courses in a term, with the Integer Programming formulation, implementation and solution. On the other hand, this report is part of Discrete Optimization 1 course at McGill University, so we will mostly pay attention to the IP case and forget about the Expert System one.

We will first dicover the problem, which is faced every semester by university staff: assign an adequate teacher to a course, given the fact there are 2 terms in a schoolyear. Secondly, we will take a look at the IP formulation of this problem, and finally study how it has been solved by the authors.

2 The problem

Every year in each university, assigning a course to a term and a teacher to a course is a huge problem, even without considering the schedule and the classroom.

Indeed, you would have to deal with:

- the skills of a teacher: whether or not he/she can teach a course
- their preferences: difference between a course he/she can teach and want to teach
- the workload of each teacher: legacy problem
- the public of each course: undergraduate/graduate
- the specificity of a course: separated in different section
- etc...

So the automation of this process can save a lot of time and money, but we need to collect a big amount of data to initiate the process. Concerning those data, you can find their hierarchy in Figure 1. This schema comes from the article but does not match the given explanation: The ”Undergrad course” box should be linked with the ”Course” one, instead of the ”Term”.

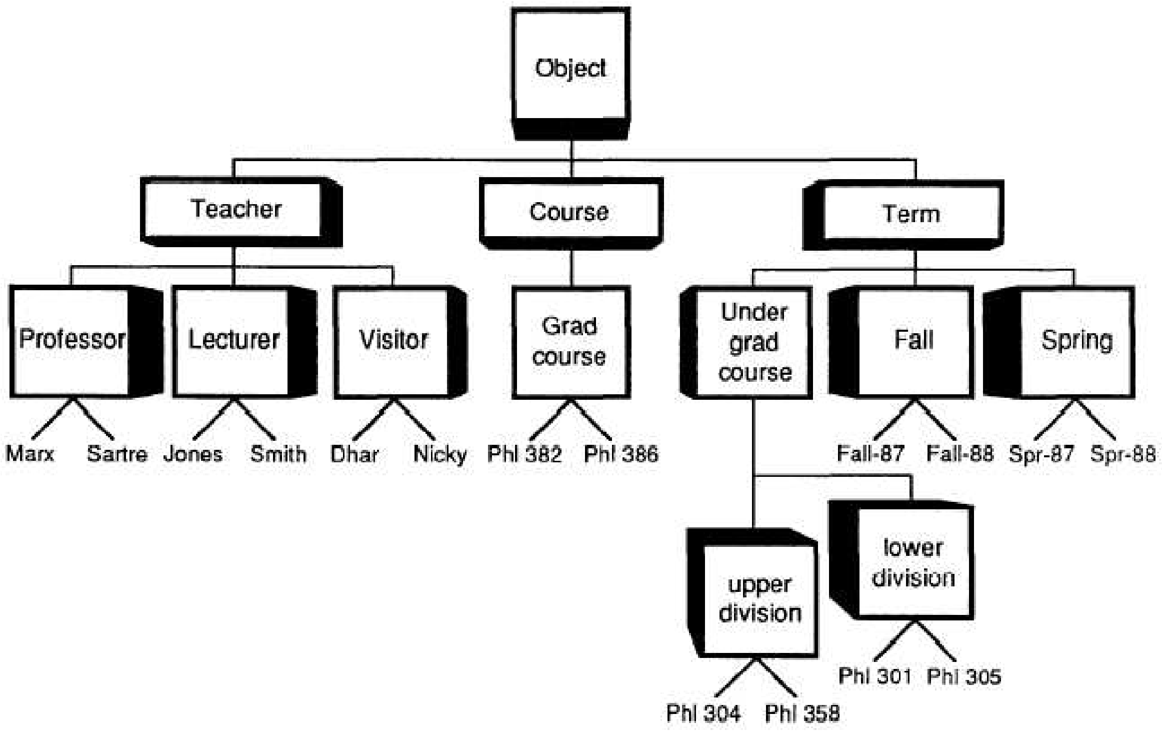

Figure 1: Data Hierarchy (Source: Article)

The problem of revising a solution will not be covered in this report, but you can find it in the original paper.

3 The Integer Programming formulation

As you may have suspected, the IP problem is huge in his definition. We will first present the subscript, the variables, the coefficients, the constraints to finish with the objective function.

3.1 Subscripts

- i = 1, 2, . . . , n instructors
- j = 1, 2, . . . , m courses
- t = 1, 2 terms (fall or spring)
- k = 1, 2, . . . , p categories of courses (for balancing curriculum)

3.2 Variables

$$x_{ijt} = \begin{cases} 1 & \text{if faculty i is assigned to course j in term t} \\ 0 & \text{otherwise} \end{cases}$$

3.3 Coefficients

- ρ_j = load of course j depending on its size and type
- T_i = teaching load requirement for teacher i
- lb_{jt} = lower bound on sections of course j to be taught in term t
- ub_{jt} = upper bound on sections of course j to be taught in term t
- lb_j = lower bound on total sections of course j to be taught in a year
- ub_j = upper bound on total sections of course j to be taught in a year
- y_j = number of sections of course j to be taught in the entire year
- G_t = the number of graduate level courses normally offered during term t
- c_{ijt} = cost of assigning teacher i to course j in a term t
- $g_j = \begin{cases} 1 & \text{if j is a graduate level course} \\ 0 & \text{otherwise} \end{cases}$
- $u_j = \begin{cases} 1 & \text{if j is an upper division course} \\ 0 & \text{otherwise} \end{cases}$
- $l_j = \begin{cases} 1 & \text{if j is an lowerer division course} \\ 0 & \text{otherwise} \end{cases}$
- $w_j = \begin{cases} 1 & \text{if j is an writing course} \\ 0 & \text{otherwise} \end{cases}$
- $C_{jk} = \begin{cases} 1 & \text{if course j is in category k} \\ 0 & \text{otherwise} \end{cases}$
- $t_{ijt} = \begin{cases} 1 & \text{if course j is proposed as a tutorial by i in term t} \\ 0 & \text{otherwise} \end{cases}$
- $AB_j = \begin{cases} 1 & \text{if course j has the A-B sequence} \\ 0 & \text{otherwise} \end{cases}$
- $f_i = \begin{cases} 1 & \text{if i is a faculty member} \\ 0 & \text{otherwise} \end{cases}$

3.4 Constraints

- The number of teachers assigned to a course in each term should be between the lower and upper bounds on the number of sections of that course :

$$lb_{jt} \leq \sum_{i=1}^{n} x_{ijt} \leq ub_{jt}$$

- The total number of sections of course j taught in the year should be:

$$lb_j \leq \sum_{t=1}^{2} \sum_{i=1}^{n} x_{ijt} \leq ub_j$$

- Each teacher must satisfy some minimal teaching load:

$$\sum_{t=1}^{m} \rho_j x_{ijt} \geq T_i$$

- Only professors can teach graduate courses:

$$x_{ijt} \leq 1 - (1 - f_i) g_j$$

- No professor can teach more than G graduate courses per year:

$$\sum_{j=1}^{m} \sum_{t=1}^{2} x_{ijt} g_j \leq G$$

- Course sequence continuity A-B constraint (Someone can only teach a B course in spring if they taught the A course in the fall):

$$AB_j(x_{ij2} - x_{ij1}) \leq 0$$

- There must be at least U upper division writing courses offered each term:

$$\sum_{j=1}^{m} x_{ijt} w_j u_j \geq U$$

These are the most relevant constraints, others are based on similar ideas.

3.5 Objective function

Among all possible goal, the authors have chosen to minimize deviations from teachers's desired courses to match their preferences:

$$\text{Minimize } z = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{t=1}^{2} c_{ijt} x_{ijt}$$

Considering our formulation, we obtain a problem with 700 binary variables and 300 constraints.

4 The Integer Programming implementation

4.1 Architecture

In 1990, the authors used the ZOOM library of the XMP package. After few researches, it appears that ZOOM/XMP is a linear and integer programming library written in Fortran.

To solve our problem, ZOOM will first find a solution to the LP relaxation. Then we can deduce the integer solution using an heuristic procedure called Pivot and Complement (E. Balas and Martin C. H.). It can describe as a sequence of pivot steps, which put all slacks variables into the basis, using the Blend's rule. When a feasible integer solution is found, we improve it by flipping variables to the opposite bounds.

The input of ZOOM comes from the Expert System but they need to be preprocessed to suit the ZOOM interface. ZOOM generates a schedule, which can be presented in different form: summaries, "database", which can answer user's question, etc...

The whole process is summarized in Figure 2.

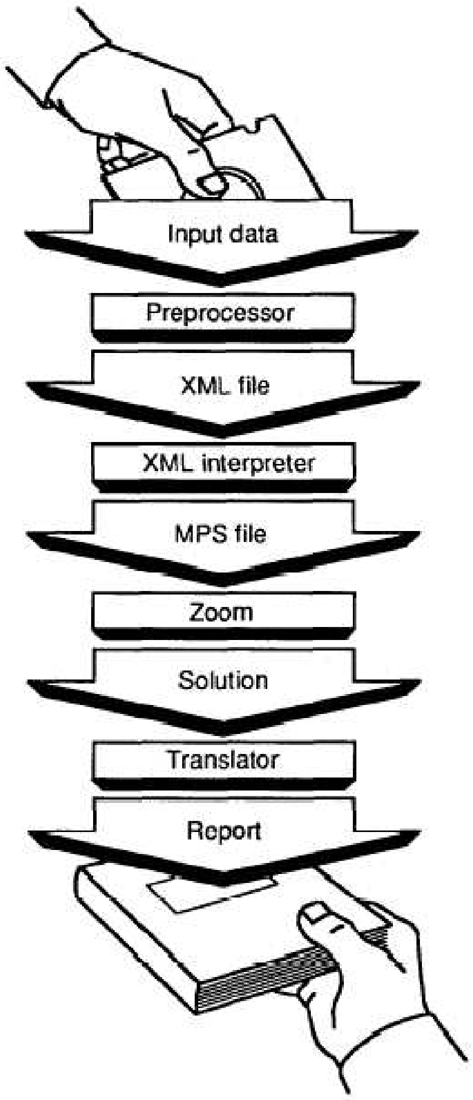

Figure 2: IP Process (Source: Article)

4.2 Performance

All the experiments were carried out on the same machine as the Expert System: a SUN-3 workstation. The response time of the IP varied from a few minutes to a few days.

When a solution is found, it comes quickly because of the pivot and complement heuristic. If this heuristic does not find a feasible solution, no solution is found at all, whereas the Expert System can provide a partial solution and indicates the holes.

4.3 Solution

In general, the IP solution is similar at 75% to the ES solution. The difference has been explained by the single objective limitations, compiled knowledge limitations and global optimization limitations.

In the IP formulation, we have specified only one objective function, in fact we have to minimize or maximize many expressions, minimizing the teacher load is an example. The problem is that we have already some difficulties to find a solution, so adding an objective function is computationnaly expensive.

The compiled knowledge limitations can be understood as the expressing power of number: the coefficient have to incorporate preferences, flexibility and trading criteria, so that any change in a value can give a totally different solution.

The global optimization limitations leads to a global optimum solution, which can unfortunately contain problem like no teacher is assigned to a section. This can be solved by adding constraints, which would give an even more complicated problem.

5 Conclusion

As a conclusion, we have seen that it is possible to use IP to solve a real problem of affecting. Unfortunately, because of his limitations, his difficulties to find a solution, his incapacity to propose a partial solution and his lack of flexibility, the IP formulation is not used and the ES is prefered.

6 References

- Vasant Dhar: vdhar@stern.nyu.edu
- ZOOM/XMP: http://cadswes.colorado.edu/directory/Tim_Magee.html
- XMP: http://www.adobe.com/products/xmp/main.html
- More recent tool to solve IP: http://support.sas.com/rnd/app/or.html
- E. Balas, C. H. Martin: Pivot and Complement – a Heuristic for 0–1 Programming, Management Science 26(1), 1980.