UTILIZATION OF INTEGER PROGRAMMING FOR THE OPTIMIZATION OF CLASS TIMETABLES

José Vicente Caixeta Filho
João Carlos Vianna de Oliveira
Campus "Luiz de Queiroz"
Universidade de São Paulo
Av. Pádua Dias, 11 - 13.418-900 Piracicaba, SP - BRAZIL

1. Aim

The high schools usually have, at the beginning of each academic year, problems to define the timetables for their classes, mainly due to the restrictions in the time availability of the teachers. In view of that, it is proposed a modeling structure to help in this kind of time tabling, based on mathematical programming algorithms.

2. Modeling structure

The proposed model is basically founded upon a 0-1 integer programming structure, which means that the decision variables for the problem will represent, under optimal conditions, the subjects that should occur for each class, along the week, by each different teacher. Since this type of structure could imply in a high number of variables, resulting in a combinatorial explosion, it was decided to interpret the decision variable as representative of the teacher-subject status, i.e., because not necessarily all the teachers are able to be responsible for all subjects, this status would consider only the possible combinations. Therefore, for this study, when referring to a *subject*, this considers implicitly the combination *teacher-subject*. In view of that, the proposed modeling structure considers the following definitions:

n = number of subjects, m = number of classes, o = number of periods during the day in which subjects are taken, p = number of weekdays in which subjects are taken;

X_{ijkl} = 0-1 decision variable, representing the possible occurrence of a subject i, for the class j, in the period k, in the weekday l;

c_{ijkl} = pseudo-cost associated to the decision variable X_{ijkl}, and inicially set equal to 1.0;

A_{ijkl} = parameter type 0-1, indicating the availability of a subject for a class in a determined period of a weekday;

TOT_{ij} = total number of periods during the week in which the subject i should be offered to the class j;

R_{ikl} = parameter representing the number of classes having the same subject i, in a determined period k of a weekday l.

It is also assumed that the schools have as their main objective to achieve the maximum possible number of subjects, that is:

$$\text{Max} \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{o}\sum_{l=1}^{p} c_{ijkl} X_{ijkl} \qquad (2.1)$$

and the following constraints taken into consideration:

$$X_{ijkl} \leq A_{ijkl} \qquad (2.2)$$

that is, a subject will be offered or not in a determined period of some weekday, respecting the availability of that subject;

$$\sum_{i=1}^{n} X_{ijkl} \leq 1.0 \qquad (2.3)$$

that is, in each class, it could happen a maximum of one subject in a determined period of each weekday;

$$\sum_{j=1}^{m} X_{ijkl} \leq R_{ikl} \qquad (2.4)$$

that is, each subject can be offered in a determined period of a weekday to one or more classes;

$$\sum_{k=1}^{o}\sum_{l=1}^{p} X_{ijkl} \leq TOT_{ij} \qquad (2.5)$$

that is, each subject can be offered according to the weekly frequency expected for each class;

$$\sum_{i=a}^{a}\sum_{j=1}^{m} X_{ijkl} + \sum_{i=b}^{b}\sum_{j=1}^{m} X_{ijkl} \leq 1.0, \text{ for } 1 \leq a,b \leq n \qquad (2.6)$$

that is, for the cases in which a teacher is responsible for more than one subject, these correspondent times must not be coincident.

3. Discussion of the results

Once it was assumed that the use of this modeling structure would be responsibility of the high schools themselves, the compatible mathematical programming software tested for that should be available for microcomputers, the most common computational platform obtainable in that category of schools.

To select the software, it was used a survey conducted by Sharda (1992), taking into account the potential power of this type of tool with respect to manipulation of large and dense matrices, which certainly demand great amount of RAM memory as well long processing times. In view of that, it was utilized the OSL (IBM, 1991) code, incorporated to the optimization language GAMS (Brooke et al., 1992), under various microcomputer configurations.

Specifically for the most representative model processed for this study, built for a representative high school at the São Paulo State, Brazil, it was generated a matrix with 7,671 rows, 34,801 columns and 93,482 non-zero elements, which required, according to the solving strategy used, an average of 20 Mb of RAM memory in a microcomputer 486 DX with a clock of 66 MHz.

The obtained results attested the efficiency of GAMS/OSL, specially regarding the linear solution. However, the search for the integer solution was not equally efficient, mainly due to the fact the algorithm used for that is based on branch-and-bound techniques.

Given the integer solution is attained from the initial linear solution (which could be got through different algorithms available in OSL, including the very convergent ones for some classes of problems, such the interior point methods), the execution time consumed was considered satisfactory.

However, to get a greater efficiency to achieve the integer solution, the following solving strategies available for OSL were tested:

a) variation in the criteria for absolute and relative optimization. When the integer solution (*I*), in terms of the value of the objective function, is not exactly equal to the linear solution (*L*), the relaxation of the tolerance for valid absolute (*I* - *L*) and relative (*I*/*L*) intervals may imply more reasonable solutions for the problem.
b) variation in the values of some pseudo-costs. If the obtained integer solution does not consider some key-subjects of the school, priorities can be given to force the presence of the corresponding variables into the basis. This can be done with the use of solving strategies available for GAMS/OSL (such as strategies 8, 16 or 32) or, eventually, by altering the values of the pertinent c_{ijkl} in the objective function.
c) relaxation of the constraints. Identified which constraints are the most binding ones, those ones could be merely removed or have their respective RHS relaxed. It is recommended, however, that this strategy be applied only to the inequality constraints.

It has also to be considered that the evaluation of those results can be facilitated if a friendly report generator is available, once the organization of those results with the default status of GAMS/OSL can be a very complex task. GAMS/OSL has some alternatives for that, as well permits the integration with routines developed under other languages.

4. References

Brooke A, Kendrick D, Meeraus A (1992) GAMS: a user's guide - release 2.25. Scientific Press, San Francisco.

IBM Corporation (1991) Introducing the Optimization Subroutine Library - release 2. IBM, Kingston.

Sharda R (1992) Linear programming software for personal computers: 1992 survey. OR/MS Today 19(3): 44-60.