An Integer Programming Model for the School Timetabling Problem

Geraldo Ribeiro Filho[•]
UNISUZ/IPTI – Av. São Luiz, 86 cj 192
01046-000 –- República –- São Paulo – SP – Brazil

Luiz Antonio Nogueira Lorena
LAC/INPE –- Cx Postal 515
12201-970 – São José dos Campos – SP – Brazil

Abstract

This work presents a binary integer programming model applied to the process of fixing a sequence of meetings between teachers and students in a prefixed period of time, satisfying a set of constraints of various types, known as school timetabling problem. Pairs of teachers and classes associated to timeslots are modeled as binary integer variables weighted by parameters referring to teachers preferences. Conflicting pairs relative to each timeslot are modeled as constraints. Computational results over real test problems are presented and compared to previous results obtained with metaheuristics applied to the same instances.

Keywords: timetabling, integer programming.

1. INTRODUCTION

The timetabling problem consists in fixing a sequence of meetings between teachers and students in a prefixed period of time (typically a week), satisfying a set of constraints of various types. A large number of variants of this problem have been proposed in the literature, which differ from each other based on the type of institution involved (university or high school) and distinct constraints [1]. A typical timetable instance requires several days of work for manual solution.

Several techniques have been developed to automatically solve the problem [2]. We therefore can find algorithms based on integer programming [3], algorithms based on the reduction of the problem to a well-studied problem known as *graph coloring* [4], approaches based on artificial intelligence techniques [5], and metaheruristics like *simulated annealing*, *tabu search*, *genetic algorithms* and *ant colony* [6][7][8][9].

We consider in this paper a problem known as *school timetabling*: to obtain the weekly scheduling for all the classes of a typical Brazilian fundamental school, avoiding teachers attending two classes at the same time, and vice versa.

[•] Address: Av. Francisco Rodrigues Filho, 399 – Mogi das Cruzes – SP – 08773-380 – Brazil
E-mails: geraldo@ipti.org.br geraldorf@uol.com.br

The context of Brazilian schools has some particular aspects that must be cleared. Before high school, Brazilian students must attend eight years of study divided in two levels, first the “basic” level and then the “fundamental” level. Each one of these levels are divided in two so called “cycles”, each one formed by two years. The problem instances used in this paper belongs to the fundamental level, that is, from the 5^{th} to 8^{th} years. A typical school may have more than one group of students, here called “classes”, attending the same year. For instance, a school could have classes known as 5A, 5B, 6A, 6B, 7A, and so forth. Each year has its curriculum of subjects, each subject with its own number of weekly lectures, each lecture scheduled to one timeslot of the week timetable.

The main objective of this work is to study the performance of the model as the basis for the eventual future development of a software tool to help administrative staff of public schools. Regarding the timetabling, some particular characteristics observed in Brazilian schools are: full use of available rooms, closed timetabling (at any timeslot all rooms are occupied), usual timeslot conflicts of classes and teachers, and soft constraints for teachers, that is, preferences to some determined timeslots and avoid waiting timeslots, or windows.

Another objective of this work is to compare the model results to those obtained over the same instances in previous work using the Constructive Genetic Algorithm (CGA) [10].

The following sections of this document, in this order, describe the model, the experiments and results, and present a conclusion, including perspectives for future work.

2. THE INTEGER PROGRAMMING MODEL

Given all possible teachers-class pairs based on previous knowledge to which classes each teacher will give lectures, and the number of timeslots, we present a binary integer programming model in which the variable x_{ij} represents the *i-th* teacher-class pair assigned to the *j-th* timeslot. Also, we can build a conflict graph $G=(V,E)$ where V is the vertex set of all teacher-class pairs and E is the edge set in which there is an edge for every pair of vertices with the same teacher or the same class. The edges represent conflicting teacher-class pairs that can not be assigned to the same timeslot.

The model can be given as follows:

$$Max \sum_{i=1}^{n} \sum_{j=1}^{p} a_{ij} x_{ij} \qquad (1)$$

Subject to

$$\sum_{j=1}^{p} x_{ij} = 1 \quad , i=1, ..., n \qquad (2)$$

$$x_{ij} + x_{kj} \le 1 \quad , j=1, ..., p \quad and \quad (j,k) \in E \qquad (3)$$

$$\sum_{i=1}^{n} x_{ij} \le n_r \quad , j=1, ..., p \qquad (4)$$

$$x_{ij} \in \{0,1\} \quad , i=1, ..., n \quad and \quad j=1, ..., p \qquad (5)$$

Where n is the number of teacher-class pairs, p is the number of available timeslots, x_{ij} are binary decision variables which solution value set to one represents the *i-th* pair assigned to the *j-th* timeslot.

Constraints (2) indicate that every pair must be assigned to exactly one timeslot, the constraints (3) avoid the assignment of two conflicting pairs to the same timeslot, constraints (4) indicate that in each timeslot there are at most n_r assignments, where n_r is the number of available rooms at the school, and constraints (5) indicate binary variables.

The constraints (4) were not implemented in our tests because the particular characteristics of the test instances guarantee full use of available rooms.

The objective function (1) considers weights a_{ij} to represent an eventual teacher's restriction for timeslots in work days (undesirable timeslots), and the teacher's precedence level among his pairs. The a_{ij} formulation is given by:

$$a_{ij} = \frac{1}{prec(t)} + (1 - restr(t, j)) \qquad (6)$$

Where

- the teacher t is allocated to some class in the *i-th* teacher-class pair,
- $prec(t)$ is the precedence level of the teacher t, and
- $restr(t, j)$ is a binary value indicating the teacher t restriction for the work day of the *j-th* timeslot.

Due to previous work with the test instances, the teachers with higher precedence were considered at level 1, immediately below were the teachers at level 2, and so forth. Formulation (6) shows that the higher the value of a_{ij}, the higher the teacher precedence and the most the teacher restriction is considered.

3. EXPERIMENTS

As one of objectives of this work was to compare results to previous work that used a heuristic, the tests were made over the same instances, described below in this section. Results are presented in table form and the tests were made using C language programs and the solver CPLEX version 7.5 [11]. Tests were performed on a 512 MB Pentium 4 HT microcomputer.

3.1 The instances

For the computational tests we have used four instances, corresponding to two typical Brazilian schools, named here Gabriel and Massaro. Morning, afternoon and evening periods were considered for the Gabriel school and only the morning period for the Massaro school.

When the tests were performed, the schools activities were already begun. The data used in the tests were taken from feasible solutions given by the school administrative staff. As the teachers' precedence levels and undesirable timeslots were unknown to us, this information was artificially generated.

The set of teachers was partitioned into three levels, according to the number of classes and overall time dedicated to the school. Teachers giving classes in less than 50% of the all timeslots in the week were considered at level three, between 50% and 75% were considered at level two, and those giving classes in more than 75% of the timeslots, had the precedence level considered one. Teachers in level one precedes the others and so on.

The teachers' undesirable timeslots were artificially identified considering their number of classes per week and the real solution manually obtained by the schools administrative staff.

Table 1 shows each instance characteristics such as number of teachers, classes, week timeslots, and the number of teachers restrictions for timeslots, total and for teachers in precedence level one.

Table 1 – Schools characteristics

School	Teachers	Classes	Timeslots	Restr.	Restr. (1)
Gabriel (Morning)	30	17	25	220	22
Gabriel (Afternoon)	38	17	25	377	22
Gabriel (Evening)	38	17	20	386	27
Massaro	18	11	20	122	10

3.2 Test results

Table 2 presents the integer programming model results compared to previous results over the same instances using an evolutionary heuristic. For each instance the table shows the best values obtained by the heuristic for the percentage of teacher restrictions attendance (total and for teachers in level one) and number of windows (total and for the teachers in level one). We call teacher windows the timeslots with no classes, between two timeslots with classes for that teacher. Windows must be avoided not only to save teacher time, but due to legal restrictions.

Opposite to the integer programming model, the heuristic was not deterministic and the table shows the best average of three runs over the same instance.

Table 2 – Results Comparison

School	Heuristic (best average - 3 runs)				Integer Programming			
	Restr. (%)	Restr. (%)(1)	Win	Win (1)	Restr. (%)	Restr. (%)(1)	Win	Win (1)
Gabriel (Morning)	93.18	87.88	33.33	7.33	92.7	81.8	40	8
Gabriel Afternoon)	95.05	84.85	32.00	3.33	98.70	95.50	42	3
Gabriel (Evening)	90.85	87.65	12.33	1.67	97.4	88.9	8	1
Massaro	93.72	96.67	4.00	0.67	98.4	100	4	1

It can be seen the table 2 that the integer programming model improve restrictions attendance (total and for level one teachers) in three out of four instances, but the total number of windows is considerably higher for two instances in the integer programming result. While the heuristic had a specific parameter to avoid windows, the integer programming model in its present form has no formulation to avoid teachers' windows.

Table 3 shows the computer time taken by CPLEX to solve the model. Curiously some instances are almost instantaneous while others are hard to solve.

Table 3 – Computer time

School	Time (sec)
Gabriel (Morning)	4860
Gabriel (Afternoon)	1852
Gabriel (Evening)	4
Massaro	3

The large difference in computer time between the first two and the last two instances shown on table 3 may be caused by specific characteristics of the model and the instances. There are few feasible solutions available and depending on the solver internal processing the solution time may be largely variable.

The instance called Gabriel (Afternoon) was then transformed to try to produce more feasible solutions. This instance has five precedence level one teachers with full week classes, and half of their classes were assigned to five new added virtual teachers with no restrictions, reducing the instance average classes/teacher ratio. The transformed instance was solved two ways: the solver was set to seek all feasible solutions and to stop when find the first feasible solution. The experiment was repeated removing one new teacher each time. Table 4 shows time results, restrictions attendance and the first solution gap given by the solver.

Table 4 – Time comparison with more feasible solutions

New Teachers	Best solution			First solution found			
	Time (sec)	Restr. (%)	Restr. (%) (1)	Time (sec)	Restr. (%)	Restr. (%) (1)	Gap (%)
5	151	99.5	100.0	75	95.5	77.3	1.73
4	419	98.1	100.0	412	98.1	100.0	0.0
3	903	98.1	100.0	112	99.2	100.0	0.14
2	1013	99.5	100.0	123	99.2	100.0	0.14
1	2399	99.5	100.0	2502	99.5	100.0	0.0

As expected, the reduction of teachers with full week schedule increases the number of feasible solutions and reduces computer time in most cases. The time required to solve the instance with just one additional teacher being even greater than the time required to solve the instance original form, indicates there are other factors to consider analyzing performance.

The coefficient a_{ij} formulation and problem characteristics may induce the same objective function value for different solutions, and therefore not only the number of feasible solutions may influence the solver performance, but also the variation of the objective function evaluation.

4. CONCLUSION

The school timetabling problem is very challenging and several days of work are normally needed to manually solve these problems. We have proposed in this paper a binary integer programming model to solve real problems considering particular characteristics. It considers the usual feasibility problem of teachers and classes allocation avoiding conflicts, teacher restrictions for some timeslots and teachers precedence relative to their pairs.

The model results were compared to previous work using an evolutionary heuristic and the results were improved regarding teachers restrictions attendance. Large difference in computer time to solve instances depends on specific characteristics and demands research. Future work on the model should include some formulation to avoid teachers' windows, and exploration of a recent proposal of Lagrangean relaxation with clusters to deal with larger conflict graphs [12].

5. REFERENCES

[1] D. de Werra, *An introduction to timetabling*, in European Journal of Operations Research, 19, 1985, pages 151–162.

[2] E. K. Burke, S. Petrovic, *Recent research directions in automated timetabling*, in European Journal of Operational Research, 140(2), 2002, pages 266-280.

[3] S. Daskalaki, T. Birbas, *Efficient solutions for a university timetabling problem through integer programming*, in European Journal of Operational Research, 160(1), 2005, Pages 106-120.

[4] E. K.Burke, B. McCollum, A. Meisels, S. Petovic, R. Qu, *A graph-based hyper-heuristic for educational timetabling problems*, in European Journal of Operational Research , to appear, 2006.

[5] M. P. Carrasco and M. V. Pato, *A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem*, in European Journal of Operational Research, 153(1), 2004, pages 65-79.

[6] Z. N. Azimi, *Hybrid heuristics for Examination Timetabling problem*, in Applied Mathematics and Computation, *163(2), 2005, pages 705-733.*

[7] R. Alvarez-Valdes, E. Crespo, J. M. Tamarit, *Design and implementation of a course scheduling system using Tabu Search,* in European Journal of Operational Research, 137(3), 2002, pages 512-523.

[8] M. P. Carrasco and M. V. Pato, *A Multiobjective Genetic Algorithm for the Class/Teacher Timetabling Problem*, Lecture Notes in Computer Science, vol. 2079, Springer-Verlag, 2000, pages 3-17.

[9] O. Rossi-Doria, M. Sampels, M. Birattari, M. Chiarandini, M. Dorigo, L. M. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, T. Stützle, *A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem*, Lecture Notes in Computer Science, vol. 2740, Springer-Verlag, 2003, pages 329-351.

[10] G. Ribeiro Filho, L. A. N. Lorena, *A Constructive Evolutionary Approach to School Timetabling*, in Applications of Evolutionary Computing, Lecture Notes in Computer Science, vol. 2037, Springer-Verlag, 2001, pages 130-139.

[11] ILOG CPLEX 7.5 Reference Manual. Copyright by ILOG, France, 2001.

[12] G. M. Ribeiro, L. A. N. Lorena, *Lagrangean relaxation with clusters and column generation for the manufacturer's pallet loading problem*, in Computers and Operation Research , to appear, 2006.