MAGHO[1] – A tool for teachers and physical resources Assignment, and for student's scheduling

Vilmar Pedro Votre

UPM - Univ. Presbiteriana Mackenzie, R. Consolação 896, São Paulo, Brazil.

vilmar@mackenzie.br

Abstract. Magho is a comprehensive tool for graduate course timetabling. It was used in UPM for the generation of term course timetables and produced fast and reliable results. It is capable of finding the best teacher for each course offered given his interest in lecturing such course and his availability, simultaneously it finds the best schedule for the course given the term structure, avoiding overlaps and gaps. It determines the ideal number of offerings of each course per term, balancing the classrooms size and avoiding excessive offerings or under enrollment. Also, it can help students, giving the best enrollment suggestion when they have to attend classes they failed from previous terms which are pre-requisite for current or future ones. Last, it also optimizes the use of classrooms where the scheduling goal is sufficient space and minimum walking.
*1 - MAGHO comes from MAckenzie Gerador de HOrarios (Timetable Generator)

Key words *- Timetable for graduate level, Lagrange Multipliers, A* Algorithm Application fort timetable, weak-method application for timetable.*

1 – 5WHs[3] for Magho

1.1 Why – the nonexistence of available tools in Brazil with the desired quality; and the UPM's Academic System request for a good tool for timetable's generation. (UPM has more than 30.000 students, distributed in nine schools - Jan 2008).

1.2 What – teacher´s timetabling, student's enrollment and classroom's assignment for lessons. Magho approaches in a similar format the three problems solved in the end of every term at UPM. It supports the teacher's timetabling, the best suggestion of enrollment for students, making classroom balance when a course has two or more offerings, and, finally, classroom's assignment for every lesson. These statements lead to Figure 1, for Magho operation (Cardoso 1997), actors omitted, for shortness.

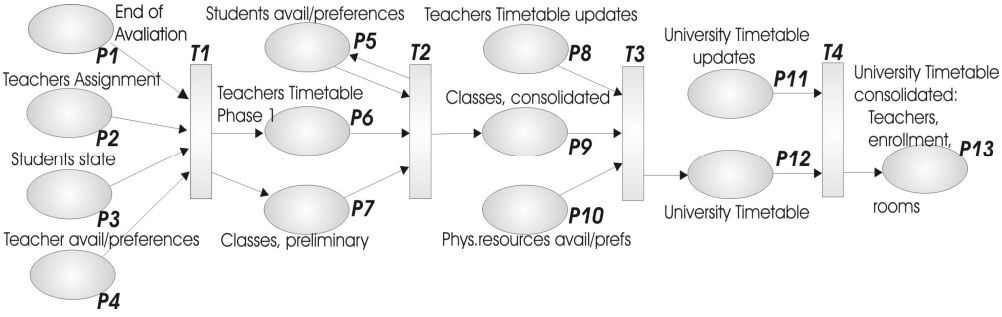

Figure 1 – Condition event Petri Net

Shortly: the term end allows estimation of students number for each discipline. Marking P2 (P1, P3 and P4 marked) fires T1, what marks P6 and P7. Students have a time slice for inputing availabilities and preferences for time slots. Then T2 fires, marking P9. When T3 fires there is the whole University tunning. Firing T4 generates outputs for effective enrollment, teachers payment and rooms scheduling.

1.3 Who - The specification is greatly of the General UPM Secretary – eng. Nelson Callegari. The implementation was my work, except for the web based interface developed by Vinicius Miana.

1.4 Where – the implementation is initiated at UPM-FCI (Faculty of Computation and Informatics). Now, I'm an full time teacher of FCI, with the main work of developing the timetable sub-system of the Academic System.

1.5 When - Magho is in development since 2005 (I was the FCI principal from 2001 to 2007). At 2006 end the teacher's assignment was in a good stage. In 2007 room's assignment and student support were solved. At the moment (January 2008) his C++ code has about 7.000 lines. The next step is the development of a Motif (Heller, 1991) interface for coordinator and for University administration of physical resources (actually is in pick-up a number style).

1.6 How - The interface is of two types: one, for heavy users (1.5, above); other, an unique dynamic web form for time slice availability and for disciplines and courses preferences, operated: by teachers; administrative staff (physical resources); by students, as illustrated by Fig. 2.

Magho is developed in Linux, C++ Gnu, under Anjuta environment. Communication is via dynamic form under Internet - CORBA (Vogel 1999) as middleware - and by CSV (Comma Separated Values) with IBM DB2 – in the core of the Academic System.

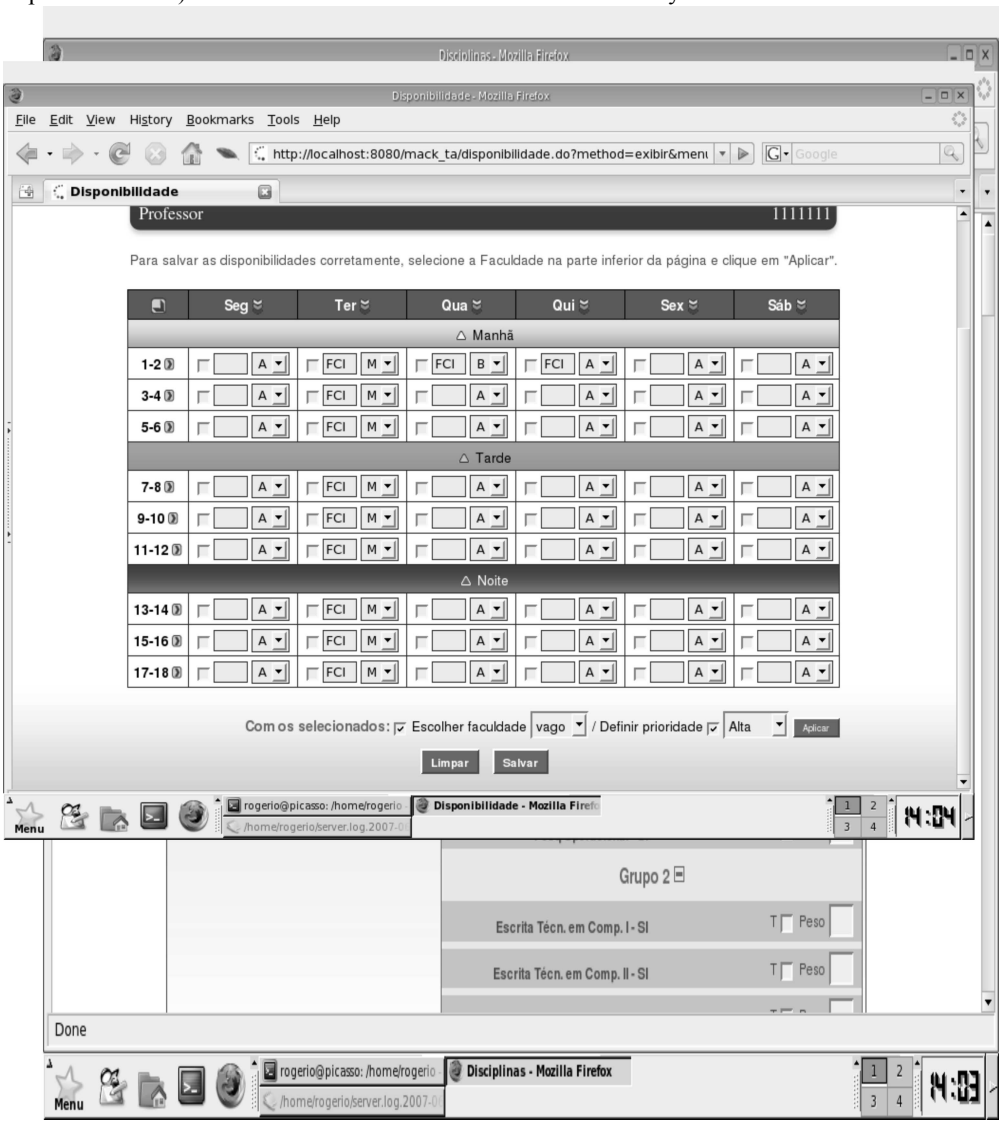

Fig. 2 – availability and preferences – teacher example

The teachers timetabling (an integer programming problem) is ambitious: Magho treats disciplines with labs based in parameters. Teacher candidates himself for lecture, lab, or both. He(she) also informs his(her) preference for disciplines and for time slots. The coordinator assigns another weight for each teacher, class-wise - dominant in the bound, guiding the assignment.

The algorithm used is A*, with a bound that assigns pounds for restrictions. Magho initiates by a diagnosis of best possible quality and balance viability of resources, for availability and preferences negotiation (Fisher,1981) prior to assignment. At assignment's end (typically of four to five hours of guidance for a school of 3.000 students and four courses) Magho applies Lagrange Multipliers (Ross 1975, Strang 1988) for assignments optimization – permutations of teachers for windows elimination and for violated restrictions - now operating in the teacher's dimension. The teacher´s availability is for the whole University, over the three periods (morning, afternoon and night), for six days of the week and various campi.

The physical resources search space, of moderate size, makes feasible to use a school-wide A* application. It assigns classrooms for each discipline, based on it´s class size (size varies due to dependencies). The students walk must be minimum.

Explanation: A* Algorithm (Clocksin & Mellish, 1981) is a branch & bound algorithm, what uses two principles: means-end analysis, borrowed from GPS (General Problem Solver), created by Allen Newel and Herbert A. Simon, announced at 1963; optimality principle, (Bellman, 1962). The first states that after your next step you must be near of the target. Se second, the principle of the dynamic programming, states that every sub-path of an optimal path is also optimal.

But the main gain (Boehm 1996, 1998) of Magho is in the student's enrollment, due to time restrictions, course's gaps, overlaps and prerequisites, specially when there are many offerings of the same lecture classes, and each with its own lab class schedule. Due to the restrictions, a great part of students can't determine a reasonable enrollment package. For implementation purposes, this search space is amenable to exhaustive search, so Magho applies the Prolog's weak method, with backtracking (Clocksin & Mellish, 1981) programmed over STL map (Josuttis 2007, Votre 1998). The performance is good: about 4 minutes for 3.000 students, the worst case is for one with a lot of disciplines that lead to 15385 combinations. This case is:

- current term: 3L (3^{th} term: class:L);
- failed courses from previous terms: 1^{rd} term (English-I 2 0, Ethic-I 2 0, Differential Calculus-I. 4 0, Operating Systems 2 2, Portuguese 2 0), 2^{th} term (English-II 2 0, Ethic-II 2 0, Differential Calculus-II. 4 0, Structured Programming-I 6 2, Computer Architecture 4 0) (pair: lecture lab);
- the student may attend failed courses in night and Saturday morning. His current term courses may only be attended in his own class (L). In his case there was night offerings (1J, 1K, 1L, 1M; 2J, 2K, 2L). There is also an offering of Differencial Calc. II, at Saturday. For every lecture class (theory) there are usually two to three different lab class schedule options. In his case Magho has searched 15385 possiblities, and suggested the enrollment of Table 1 (in portughese, the disciplines; 1M – lecture class; 1M21 - one lab class):

Table 1 – enrollment suggested for one student

	Monday	Tuesday	Wednesday	Thursday	Friday
13	1M Calc. Dif. I	2J Arquit. Computad	2L Prog. Estrut I	1J Escrita Tecn. I	2L21 Progr Est I
14	1M Calc. Dif. I	2J Arquit. Computad	2L Progr Estrut I	1J Escrita Tecn. I	2L21 Progr Est I
15	2L Ingles Tecnic II	1M21 Ambientes Oper	1L Ingles tecnic I	2J Arquit. Computad	2L Progr Estr I
16	2L Ingles Tecnic II	1M21 Ambientes Oper	1L Ingles tecnic I	2J Arquit. Computad	2L Progr Estr I
17	2L Progr Estr I	1J Escrita Tecn. I	1M Calc. Dif. I	1M Ambientes Operac	1L Etica e Cid I
18	2L Progr Estr I	1J Escrita Tecn. I	1M Calc. Dif. I	1M Ambientes Operac	1L Etica e Cid I

Week Class Hour: ..34 (Saturday 2-5 – morning Calc Dif II)

Note: Mackenzie period has six time slots (morning, afternoon, night; 13-18 are at night).

1.7- How much – the cost of the dynamic form development, Fig. 2, was of US$50.000,00. The coding was my work. I have a good knowledge of the problem; and I´m a good C++ programmer. As principal of FCI, the conditions for progress were few. At moment I´m working at good silence conditions.

2 - Conclusions and Comments

The results for the preliminary version are very favorable, under aspects of time and quality (Melicio 2007). Initially Magho is tuned for UPM, but in future it is intended to be offered to other brazilian universities. It also appears as a good experience for students of our computer courses, specially for Object Oriented design and implementation, do to its challenge aspects.

References

1. Bellman, R., Dreyfus, S. (1962), *Applied Dynamic Programming*, Princeton Univ. Press, Princeton, NJ., .
2. Boehm, Barry (1996) Anchoring the Software Process. IEEE Software, July.
3. ------ (1998) Developing Multimedia Applications with the WinWin Spiral Model LNCS, Volume 1301, May.
4. Cardoso, J. & Valette, R. (1997) - Redes de Petri. Ed. UFSC, Florianópolis, SC.
5. Clocksin, W. F. & Mellish, C. S (1981) - *Programming in Prolog* - Springer-Verlag
6. Fisher, R. and Ury, W. - (1981) Getting to Yes. Boston, MA: Houghton-Mifflin.
7. Heller, Dan (1991) - Motif Programming Manual. O´Reily Associates
8. Josuttis, Nicollay (2007) – The C++ Standard Library - A Tutorial and Reference. www.josuttis.com
9. Melicio, F. et all (2007) - THOR – A tool for school timetable. PATAT 2007
10. Ross, G. Terry (1975) – A branch and bound algorithm for the generalized assignment problem. Mathematical programming 8 (1975) 91-103
11. Strang, G. (1988) – Linear Algebra and it's applications. 3´ed. San Diego: Harcourt Brace Jovanovitch.
12. Vogel, Andreas et all (1999) – C++ Programming with CORBA – John Wiley & Sons, Inc.
13. Votre, Vilmar Pedro (1998) – C++ Explicado e Aplicado. (Independent Edition).