Course Timetabling via Genetic Algorithms: A Real Case

Ricardo T.A. de Oliveira
Department of Statistics and Informatics
Federal Rural University of Pernambuco
52171-900, Recife, Pernambuco, Brazil

Filippo César G. Régis
Department of Statistics and Informatics
Federal Rural University of Pernambuco
52171-900, Recife, Pernambuco, Brazil

Paulo Renato A. Firmino
Center for Science and Technology
Federal University of Cariri
63048-080, Juazeiro do Norte, Cear, Brazil

Tiago A.E. Ferreira
Department of Statistics and Informatics
Federal Rural University of Pernambuco
52171-900, Recife, Pernambuco, Brazil

ABSTRACT

Courses timetabling has been one of the main problems for planning, maintaining and optimizing educational institutions. However, the intriguing mathematical problem which usually result from the attempt of promoting optimal courses timetabling has prevented a widely dedication of education managers to this area. The present paper aims to summarize the usefulness of approximate techniques ($e.g.$ genetic algorithms) for dealing with courses timetabling. In particular, the successful application of the resulting algorithm in a Brazilian university center is highlighted.

General Terms

Optimization, Timetabling, Scheduling Problem

Keywords

Timetabling, Genetic Algorithms, Scheduling Problem

1. THE COURSES TIMETABLING PROGRAMMING PROBLEM

The elaboration of courses timetabling can be mathematically classified as a NP-hard combinatorial problem [5]. Thus, the greater the number of classrooms, disciplines, students, and professors, the so much greater the required time for obtaining an optimal solution is. Therefore, depending on these parameters the computation of exact solutions is challenging and numerical methods are required. In this way, one can find a number of researches in literature based on approximate techniques such as Tabu Search [7, 4], Evolutionary Algorithms [9], Scatter Search [8], Path Relinking, Variable Neighborhood Search, and Hybrid Heuristics [10, 2]. Actually, timetabling approaches are in development since last decades. Events like the Practice and Theory on Automated Timetabling and the International Timetabling Competition have been paramount for this purpose.

The parameters reflecting organizational characteristics of the educational system (*e.g.* the number of classrooms, disciplines, students, professors, and courses) are necessary for an adequate modeling of the courses timetabling problems, though they are not sufficient. Didactic and personal goals must also be taken into account. The former are related to the demands for students' learning and the latter are usually dedicated to improve professors' performance.

In fact, these constraints may change through the time [1] and they must be respected in order to maintain the good performance of the institution. Among the didactic goals one must emphasize:

—Each student has a maximum number of classes per day;

—Each class must satisfy its week workload;

Further objectives may also be considered. For the sake of illustration, in order to improve students' learning performance, Costa [4] suggests that disciplines of a same course must be spread as uniformly as possible in different days of the week. In turn, though the elaboration of a timetabling that fully satisfy professors' preferences might be impossible in practice [12], the involvement of weak constraints in this way would certainly allow the promotion of a better performance of the educational system. In this way, to qualitatively ($i.e.$ from personal judgments only) handle courses timetabling might be discouraged.

As follows a genetic algorithm (GA) approach designed to involve organizational, didactic, and personal characteristics of a Brazilian university center is presented. In order to specify the organizational, didactic, and personal goals taken into account in the present paper, Table 1 brings a summary of notations.

2. THE PROPOSED GENETIC ALGORITHM

GA are in fact one of the several available techniques designed for optimization purposes. They are a probabilistic search technique in the sense that they rely on analogies to natural processes by generating, combining, and mutating individuals. With this regard each individual involves a chromosome which is in fact an instance from the state space of a set of (decision) variables.

The evolution is governed according to a given fitness function adjusted to the resulting individuals [3, 11] in such a way that the greater the fitness value of a given individual the greater the chance

Table 1. Summary of notations.

Symbol	Description	Values
n_{pf}	number of professors	14
n_{pe}	number of periods	9
n_{da}	number of class days	7
n_{sc}	number of schedules (time intervals)	2
n_{di}	number of disciplines	53
n_{lb}	number of laboratories	4
n_g^{th}	number of generations of the GA simulation	1 million
$y_{ijk}^{(l)}$	indicative functions related to the disciplines goals	1 or 0
$x_{ijk}^{(l)}$	indicative functions related to the professors' goals	1 or 0
$f_A(y^{(0)})$	the number of disciplines on undesirable days	-
$f_B(x^{0)})$	the number of professors on undesirable days	-
$f_C(y^{(1)})$	the number of disciplines on unavailable days	-
$f_D(x^{(1)})$	the number of professors on unavailable days	-
$f_E(y^{(2)})$	the number of disciplines allocations in the laboratories	-
$f_F(x^{(2)})$	the number of allocations of the professor	-
$fitness(\cdot)$	the value of the objective function related to a given chromosome	-

of it is selected for the next generations. Specifically, genetic operators like selection, crossover, and mutation are enveloped by GA for operating on the individuals [3, 11]. As follows such operators are presented in the context of the present paper.

2.1 The real timetabling problem

The problem addressed in this work is composed by $n_{pf} = 14$ professors of the Computer Science Program of a given Brazilian university center. The curriculum grid of the program involves $n_{pe} = 9$ periods and the classes occur on $n_{da} = 7$ days: from Mondays to Saturdays. Further, every class day is divided in $n_{sc} = 2$ time intervals (schedules) and the institution offers $n_{lb} = 4$ computer laboratories (LABIN), where 20 out of the $n_{di} = 53$ available disciplines should be distributed through the LABIN everyday. All notation employed here is described in the Table 1 .

2.2 The chromosomes

In the present work, a chromosome is a given timetabling for the aforementioned Computer Science Program. Specifically, it is a matrix where each row is a period at first level and a time interval (schedule) at second level, whilst each column is a week day. Table 2 illustrates this structure in such a way that each cell bin must be fulfilled by the double (Discipline and Professor).

2.3 Selection

A roulette wheel method, similar to the one described in James [6], was adopted for selecting chromosomes. So the greater the value of the fitness the greater the probability of the respective individual being selected for genetic manipulation. At the end of this step, the population of individuals to be crossed and mutated is obtained.

Table 2. Chromosome structure.

Period	Monday	Tuesday	...	Saturday
1	schedule 1	schedule 1	...	schedule 1
1	schedule 2	schedule 2	...	schedule 2
2	schedule 1	schedule 1	...	schedule 1
2	schedule 2	schedule 2	...	schedule 2
...	...	...	...	...
9	schedule 1	schedule 1	...	schedule 1
9	schedule 2	schedule 2	...	schedule 2

2.4 Crossover

Given the set of selected individuals, the crossover operation is dedicated to combine their chromosomes. In this way, under a 80% crossover probability each individual is crossed with other one. Then the crossover between two chromosomes, say $parent_a$ and $parent_b$, is performed as follows in order to generate two new chromosomes, say $child_{ab}$ and $child_{ba}$: a period, say i, is selected at random (from 1 to 9). Then, $child_{ab}$ equals $parent_a$ but at period i $child_{ab}$ equals $parent_b$. The composition of $child_{ba}$ follows the same reasoning.

2.5 Mutation

The mutation operator is designed to mimic the possibility of new individuals bringing characteristics that disregard from genetic inheritance, something possible though rare in practice. Here each new chromosome, say $child$, is exposed to mutation at random with a 10% probability. Specifically, the mutation operation is given by exchanging and it is summarized as follows: a period, say i, is selected at random (from 1 to 9) as well as two days, say day_a and day_b. Then one discipline offered in day_a and other in day_b of $child$ in period i are exchanged.

2.6 Stop condition

Two stop criteria are considered for the simulation: When the n_g^{th} generation is achieved or when every constraint is satisfied. This latter condition is reflected in the best fitness value.

2.7 Fitness

In summary, the fitness (objective) funtion adopted in the present paper involves organizational, didactic, and personal goals and constraints in such a way that the greater the fitness value the better the individual is. The constraints taken into account with this regard are grouped in three types: undesirable, unavailable, and the inviolable ones.

—Undesirable constraints: They are considered weak in the sense that they can be violated. For example, the professors' preferences for a given day;

—Unavailable constraints: They are considered moderate in the sense that they can be violated if and only if there is a competing constraint of the same nature or even inviolable. In other terms, an unavailable constraint might be violated if a viable timetabling is prevented otherwise;

—Inviolable constraints: They are considered strong in the sense that they can never be violated. As illustration, the allocation of the same professor to different classrooms at the same time is inviolable as well as the number of classes in the laboratories at a given schedule being bounded by the number of laboratories.

In order to formally define the fitness function and the constraints, let $y^{(l)}_{ijk}$ and $x^{(l)}_{ijk}$ be indicative functions, respectively related to the disciplines and professors such as:

$$y^{(0)}_{ijk}\begin{cases}1, & \text{If the discipline } y \text{ is allocated in the period } i\text{, at desirable day } j \text{ and schedule } k;\\ 0, & \text{Otherwise.}\end{cases}$$

$$y^{(1)}_{ijk}\begin{cases}1, & \text{If the discipline } y \text{ is allocated in the period } i\text{, at undesirable day } j \text{ and schedule } k;\\ 0, & \text{Otherwise.}\end{cases}$$

$$y^{(2)}_{ijk}\begin{cases}1, & \text{If the discipline } y \text{ is allocated at a laboratory in the period } i\text{, at day } j\text{, schedule } k;\\ 0, & \text{Otherwise.}\end{cases}$$

$$y^{(3)}_{ijk}\begin{cases}1, & \text{If the discipline } y \text{ is allocated in the period } i\text{, at unavailable day } j \text{ and schedule } k;\\ 0, & \text{Otherwise.}\end{cases}$$

In turn, regarding $x^{(l)}_{ijk}$:

$$x^{(0)}_{ijk}\begin{cases}1, & \text{If the professor } x \text{ is allocated in the period } i\text{, at undesirable day (to him) } j \text{ and schedule } k;\\ 0, & \text{Otherwise.}\end{cases}$$

$$x^{(1)}_{ijk}\begin{cases}1, & \text{If the professor } x \text{ is allocated in the period } i\text{, at unavailable day (to him) } j \text{ and schedule } k;\\ 0, & \text{Otherwise.}\end{cases}$$

$$x^{(2)}_{ijk}\begin{cases}1, & \text{If the professor } x \text{ is allocated in the period } i\text{, at day } j \text{ and schedule } k;\\ 0, & \text{Otherwise.}\end{cases}$$

Regarding the undesirable, unavailable, and inviolable restrictions,

2.7.1 Undesirable constraints. A discipline y of the period i should not be allocated in an undesirable day j and shedule k. This restriction is mathematically formulated as follows:

$$f_A(y^{(1)}) = \sum_{i=1}^{n_{pe}}\sum_{j=1}^{n_{da}}\sum_{k=1}^{n_{sc}} y^{(1)}_{ijk} \leq 0, \qquad (1)$$

where
y belongs to the set of disciplines (indexed from 1 to n_{di});
$i \equiv$ index of the period (i = 1, 2,..., n_{pe});
$j \equiv$ index of day of the week (j = 1, 2,..., n_{da});
$k \equiv$ index of the schedule (k = 1, n_{sc}).
Taking professors' preferences into account, the professor x should not be allocated at schedule k of the period i in the undesirable day j:

$$f_B(x^{(0)}) = \sum_{i=1}^{n_{pe}}\sum_{j=1}^{n_{da}}\sum_{k=1}^{n_{sc}} x^{(0)}_{ijk} \leq 0, \qquad (2)$$

where x belongs to the set of professors (indexed from 1 to n_{pf}).

2.7.2 Unavailable constraints. A discipline y of the period i should not be allocate in an unavailable day j:

$$f_C(y^{(3)}) = \sum_{i=1}^{n_{pe}}\sum_{j=1}^{n_{da}}\sum_{k=1}^{n_{sc}} y^{(3)}_{ijk} \leq 0. \qquad (3)$$

Then, a professor x may not be allocate at one schedule k of the period i in the day j classified as unavailable for him:

$$f_D(x^{(1)}) = \sum_{i=1}^{n_{pe}}\sum_{j=1}^{n_{da}}\sum_{k=1}^{n_{sc}} x^{(1)}_{ijk} \leq 0. \qquad (4)$$

2.7.3 Inviolable constraints:. At schedule k the total sum of allocations of disciplines that make use of laboratories is constrained by the number of laboratories of the institution:

$$f_E(y^{(2)}) = \sum_{i=1}^{n_{pe}}\sum_{j=1}^{n_{da}}\sum_{k=1}^{n_{sc}} \left(y^{(2)}_{ijk} - n_{lb}\right) \leq 0. \qquad (5)$$

On the other hand, a professor may not assume more than a classroom per schedule per day. Therefore the professor x may not be allocated in two different periods i, in the same day j and same schedule k:

$$f_F(x^{(2)}) = \sum_{i=1}^{n_{pe}}\sum_{j=1}^{n_{da}}\sum_{k=1}^{n_{sc}} (x^{(2)}_{ijk} - 1) \leq 0, \qquad (6)$$

Thus, in order to more severely penalize unavailable constraints in relation to undesirable ones, the resulting fitness function is given by

$$\begin{aligned} &fitness(individual) = \\ &\left[\sum_{y=1}^{n_{di}} f_A(y^{(0)}) - \sum_{y=1}^{n_{di}} f_A(y^{(1)}) - \sum_{x=1}^{n_{pf}} f_B(x^{(0)})\right] \times 500 \\ &- \left[\sum_{y=1}^{n_{di}} f_C(y^{(3)}) + \sum_{x=1}^{n_{pf}} f_F(x^{(2)})\right] \times 1000 \\ &- \left[\sum_{x=1}^{n_{pf}} f_D(x^{(1)}) + \sum_{y=1}^{n_{di}} f_E(y^{(2)})\right] \times 3000 \qquad (7)\end{aligned}$$

3. RESULTS

The resulting GA approach has been implemented in Java language and it has been ran in a personal computer with Windows 7 Basic operating system, with an Intel Pentium Dual-Core T4300 processor with 2.10 GHz and 4GB of RAM. The mutation and crossover rates were respectively 10% and 80% whilst the population was composed by 1500 individuals and the maximum number of generations equaled 1 million.
Four components of the fitness function have been accompanied through simulation (the negative values considered in Equantion (7) were suppressed). The first one calculates the aptitude with the shock of the professors' schedule (a), Equation (6); the second one calculates the professors' preferences (b), Equations (2) and (4); the third computes the restrictions of the disciplines (c), Equations (1), (3), and (5), and the last function (d) is the overall fitness value from functions a, b, and c, Equation (7).
In Fig. (1) it is shown the results of the tests accomplished with 10 thousand and 50 thousand generations. As expected the greater the number of generations the better the fitness is. Further, Function (a)

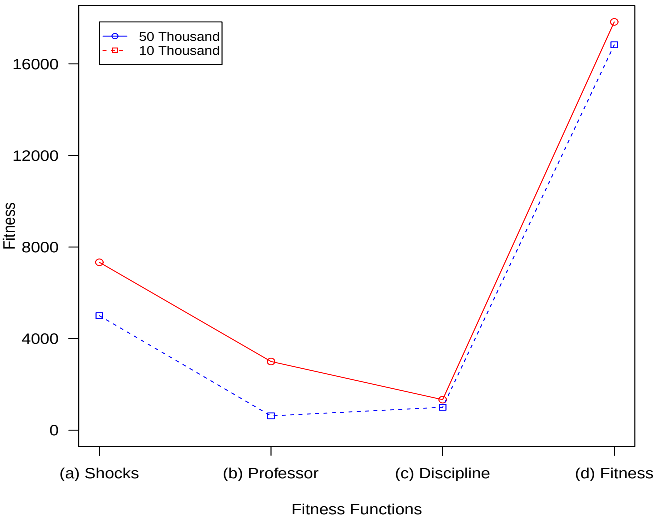

Fig. 1. The performance of the best GA individual per function: (a) Number of schedule shocks; (b) Number of violations against professors' preferences; (c) Number of violations against disciplines demands; (d) The overall $fitness$ function.

allows one to search for solutions where there is no schedule shock. It is thus notable that this function is more intriguing than the other ones ($i.e.$, (b) and (c)). In turn, Function (b) is dedicated to optimize professors' preferences and it operates similarly to Function (a). Regarding Function (d), the overall fitness value, it is evident from the figure that the solution obtained with 50 thousand generations is very similar to that involving 10 thousand generations.

To achieve the best solution, the algorithm has spent about two hours of computation time. The quality of the solution provided by the proposed algorithm was evaluated by means of the parameters: quantity of shocks of schedules, number of violations against professors' preferences, disciplines, and the rate of laboratories usage per day.

Regarding the best solution found in the simulation, Table 3 exhibits that 7% (1 out of 14) of the professors would not have personal preferences satisfied, only. In turn, 18% (10 out of 53) of the disciplines would not have its didactic constraints satisfied. In turn, the found solution did violate no schedule constraint. Specifically, every discipline which requires laboratories (a total of 20) had been satisfied and the constraint regarding the number of laboratories per schedule per day had always been satisfied. Such a solution seemed unachievable in a qualitative timetabling elaboration.

Table 3. Summary of the satisfied and unsatisfied demands from professors and disciplines.

Restriction	**Unavailable**	**Undesirable**	**Desirable**
The professors' preferences	0%	7%	93%
Disciplines demands	0%	18%	82%

4. CONCLUSIONS

This work has addressed the courses timetabling problem of a Brazilian university center by means of a genetic algorithm method. The mathematical program problem taken into account had been an effort to consider not only organizational constraints but also didactic and personal goals.

The resulting algorithm has been implemented in Java language and has presented useful solutions, though at high computational time and memory demands. Anyway, it seems that the qualitative elaboration (from personal judgments only) of a solution similar to the proposed one would be an unlikely event.

In order to improve and allow the application of the resulting algorithm in more complex educational systems, ongoing works regarding the research for alternative heuristics and languages are currently in advance.

Acknowledgment

The authors would like to thank CAPES (Coordination for the Improvement of Higher Level Personnel), FACEPE (Science and Technology Foundation of the State of Pernambuco) and UFRPE (Federal Rural University of Pernambuco) for the support.

5. REFERENCES

[1] J. V. D. Broek, C. Hurkens, and G. Woeginger. Timetabling problems at the tu eindhoven. *European Journal of Operational Research*, 196(3):877–885, 2009.

[2] E.K. Burke, A.J. Eckersley, B. McCollum, S. Petrovic, and R. Qu. Hybrid variable neighbourhood approaches to university exam timetabling. *European Journal of Operational Research*, 206(1):46 – 53, 2010.

[3] David A. Coley. *An introduction to genetic algorithms for scientists and engineers*. World Scientific, 1999.

[4] Daniel Costa. A tabu search algorithm for computing an operational timetable. *European Journal of Operational Research*, 76(1):98 – 110, 1994.

[5] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5(4):691 – 703, 1976.

[6] Tabitha James, Mark Vroblefski, and Quinton Nottingham. A hybrid grouping genetic algorithm for the registration area planning problem. *Computer Communications*, 30(10):2180 – 2190, 2007.

[7] Zhipeng L and Jin-Kao Hao. Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200(1):235 – 244, 2010.

[8] Nashat Mansour, Vatche Isahakian, and Iman Ghalayini. Scatter search technique for exam timetabling. *Applied Intelligence*, 34(2):299–310, 2011.

[9] Danial Qaurooni and Mohammad-R. Akbarzadeh-T. Course timetabling using evolutionary operators. *Applied Soft Computing*, 13(5):2504 – 2514, 2013.

[10] Rong Qu, Edmund K. Burke, and Barry McCollum. Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems. *European Journal of Operational Research*, 198(2):392 – 404, 2009.

[11] Colin R Reeves and Jonathan E Rowe. *Genetic algorithms-principles and perspectives: a guide to GA theory*, volume 20. Kluwer Academic Pub, 2002.

[12] Haroldo G. Santos, Luiz S. Ochi, and Marcone J.F. Souza. A tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem. *J. Exp. Algorithmics*, 10, December 2005.