Acta Manilana 63 (2015), pp. 79–85
Printed in the Philippines
ISSN: 0065–1370

Solving the class timetabling problem by 0–1 integer programming

Marilyn C. Mabini

Research Center for the Natural and Applied Sciences
University of Santo Tomas, 1015 Manila, Philippines

This paper presents a two-stage approach to solving the timetabling problem for an academic department. The first stage seeks to find the best faculty assignment to courses and sections where these courses are offered. The assignment is based on faculty specializations. The second stage finds an appropriate weekly schedule for the faculty-course-section assignment obtained in the first stage. The problems in both stages are modeled as 0–1 integer linear programming problems. The models are solved using LINGO 14, courtesy of Lindo Systems, Inc., with run times practically negligible.

Keywords: faculty assignment, school timetabling, 0–1 integer linear programming

INTRODUCTION

This paper is concerned with the problem of assigning classes to faculty members and constructing a weekly schedule for the faculty-class assignment scheme. The desired schedule is also known as a school timetable. For our purposes, the term “class” shall be used to refer to a course-section pair, where a section consists of a group of students.

The timetabling problem is combinatorial and NP complete, meaning that if one examines all possible schedules, the time to find the best one would increase significantly as the problem size increases. Since the 1960s, the timetabling problem has been studied and solved using various approaches, including graph coloring, simulation, tabu search, genetic algorithms, and mathematical programming. A good number of studies have appeared in the literature using classical mathematical programming techniques, such as integer programming and goal programming, to solve the timetabling problem.

PROBLEM DESCRIPTION

This study deals with the timetabling problem for an academic department at an engineering school. The department is headed by a Chair, who is responsible for, among other things, drawing up the faculty load and the class schedule for each academic term. Each academic year is comprised of three terms, namely, the first and the second semesters as regular terms, and the summer term. The latter provides a venue for students with failing grades to remove their failures before the next regular term commences.

**To whom correspondence should be addressed*
marilyn.mabini@ust.edu.ph / mcmabini2001@yahoo.com

At present, the faculty-course-section assignment and the class scheduling processes are done manually. The Department Chair determines the number of regular sections to be opened per year level for a given academic term, based on student enrolment forecasts. The forecasts are made by the Chair on the basis of student performance in the preceding term. Each regular section has a known set of courses defined by the official curriculum.

Faculty loading is dictated by the specialization of faculty members, as well as university regulations and provisions of the collective bargaining agreement (CBA) with the Faculty Union. The latter items essentially cover limitations on the load that a faculty member can carry each term. Occasionally, faculty members would make load-related requests, which could be given favorable consideration if they are not prejudicial to university rules and/or CBA provisions.

With the faculty specializations and possible course and section assignments of faculty members known, the Department Chair schedules classes subject to a number of considerations. These considerations include curricular requirements, classroom availability, allowable daily teaching hours, class schedule desirability for faculty members and students, etc. It is understood that no two classes shall be scheduled at the same time on a given day; and at most one class may be taught by one faculty member on a given time slot of any day.

Related literature

The class timetabling problem has been solved using different approaches, including simulated annealing [1]), genetic algorithm [2, 3], graph heuristics [4] and particle swarm optimization [5].

In recent years, significant developments made in mathematical programming optimization software have drawn research interest to the use of mathematical programming approaches in solving the school timetabling problem.

Mirrazavi *et al.* [3] developed a large-scale integer goal programming model for the timetabling problem at one department of a university. The solution approach consisted of two phases; the first of which allocated lectures to rooms, while the second allocated start-times to lectures. Both phases were modeled as integer goal programming (IGP) problems. The first phase was solved using an IGP optimization package, and the second phase was solved using a genetic algorithm.

Ozdemir and Gasimov [6] applied a nonlinear zero-one multi-objective programming formulation for the faculty-course assignment problem. They approached the nonconvex multiobjective problem using a three-step process, consisting of the Analytic Hierarchy Process, scalarization, and the subgradient method.

Daskalaki *et al.* [7] used 0–1 integer programming for the university timetabling problem. The model employed a linear cost function which allowed preferences regarding teaching periods or days or classrooms to be considered.

Ismayilova *et al.* [8] proposed a multi-objective 0–1 linear programming model considering both the administration's and instructors' preferences for the faculty-course-time slot assignment problem. The Analytic Hierarchy Process (AHP) and Analytic Network Process (ANP) were applied to weigh the different and conflicting objectives. The weights obtained were used in the scalarization process, and the resulting scalarized problems were solved using an optimization package.

Al Yakoob and Sherali [9] also used integer programming to solve the problem of assigning faculty members to classes at the Department of Mathematics and Computer Science in Kuwait University. Initially assuming the time-slots for classes to be given, an integer programming model was developed for the resulting problem with the objective of minimizing the dissastisfaction of faculty members. Subsequently, the model was modified to allow the time-slots for the classes to be changed, subject to restrictions related to facility utilization, and providing a limit to the number of changes. Gender-based modeling considerations were introduced in the solution.

Sorensen and Dahms [10] solved a practical case of high school timetabling using a two-stage decomposition of an integer programming (IP) model. The model was split into two separate problems (forming Stage I and Stage II) with significantly fewer variables, solved sequentially. Computational results showed a considerably better performance for the two-stage model over the original IP model.

Approach

We approach the present problem in two stages, both utilizing 0–1 integer programming. In the first stage, we seek the assignment of faculty members to courses and sections, mainly on the basis of faculty specializations. The optimal assignment obtained from this stage is then passed on to the second stage for the determination of the desired class schedule. Stage 2 finds the day and the time slot for each faculty-course-section assignment generated in stage 1, which accommodates timetabling requirements for the department.

Model formulation

The 0–1 integer programming model for the first stage is as follows:

$$\text{Minimize } Z = \sum_{i\in I} \sum_{j\in J} c_{ij} \sum_{k\in K} w_{ijk} \qquad (1.01)$$

subject to:

$$\sum_{j\in J} \sum_{k\in K} a_j w_{ijk} <= U_i, \text{ for all i} \qquad (1.02)$$

$$\sum_{j\in J} \sum_{k\in K} a_j w_{ijk} >= L_i, \text{ for all i} \qquad (1.03)$$

$$\sum_{i\in I} \sum_{k\in K} w_{ijk} = n_j, \text{ for all j} \qquad (1.04)$$

$$w_{ijk} = \begin{cases} 1, \text{ if faculty member i is assigned} \\ \quad \text{to course j with section k} \\ 0, \text{ otherwise} \end{cases} \qquad (1.05)$$

where,

c_{ij} = "cost" of assigning faculty member i to course j; this cost is associated with faculty specialization (see description below)
U_i = upper limit on the number of units that may be assigned to each faculty member
L_i = lower limit on the number of units that may be assigned to each faculty member
a_j = number of credit units for course j
n_j = number of sections taking course j

The objective function (1.01) strives to minimize the total cost of assigning the faculty members to teach the course offerings in the sections opened in the school term under consideration. The assignment cost is a value used to reflect the course specializations of each faculty member, so that when course j is within the teaching specialization of faculty i, c_{ij}, is assigned a zero value. The constraint set (1.02) ensures that the load of each faculty member does not exceed the upper limit prescribed by the university. The constraint set (1.03) is applied to uphold the minimum load of each faculty member. The constraints in (1.04) take care that there are as many faculty assignments to each course as needed by the number of sections opened.

In the second stage, the faculty-course-section assignments obtained from the first stage are scheduled. The mathematical model, presented below, endeavors to minimize the total scheduling cost (2.01). Here, a linear cost function, c_m, is used which increases as class time progresses each day. Constraint sets (2.02) and (2.03), respectively, are applied to ensure that the upper and lower limits on each faculty member's teaching load are complied with. Constraints (2.04) take care that at most one section k, has a class on day l ("l" is lower case "L") and time slot m. Meanwhile, the constraints in (2.05) are applied so that each faculty member, i, has at most one class to teach at time slot m on day l. The required number of weekly hours each course j must be taught is provided for by the set of constraints (2.06). The application of the constraints in (2.06) and (2.07) is based directly on the results of the first stage model. For courses which must be taught for two or more hours in a day, the class hours must be consecutive. Adapted from Daskalaki *et al.* [7], the constraints in (2.08) are used to provide for this requirement.

Meanwhile, constraints (2.09) limit each faculty member to teach only up to a certain number of hours each day. Finally, the constraints in (2.10) ensure that the number of classes that can take place simultaneously each day may not exceed the number of rooms available to the department.

$$\text{Minimize } Z = \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{l\in L}\sum_{m\in M} c_m x_{ijklm} \quad (2.01)$$

subject to:

$$\sum_{j\in J}\sum_{k\in K}\sum_{l\in L}\sum_{m\in M} x_{ijklm} <= U_i, \text{ for all i} \quad (2.02)$$

$$\sum_{j\in J}\sum_{k\in K}\sum_{l\in L}\sum_{m\in M} x_{ijklm} >= L_i, \text{ for all i} \quad (2.03)$$

$$\sum_{i\in I}\sum_{j\in J} x_{ijklm} <= 1, \text{ for all k,l,m} \quad (2.04)$$

$$\sum_{j\in J}\sum_{k\in K} x_{ijklm} <= 1, \text{ for all i,l,m} \quad (2.05)$$

$$\sum_{l\in L}\sum_{m\in M} x_{ijklm} = H_j, \text{ for all i,j,k; } w_{ijk} = 1 \quad (2.06)$$

$$\sum_{l\in L}\sum_{m\in M} x_{ijklm} = 0, \text{ for all i,j,k; } w_{ijk} = 0 \quad (2.07)$$

$$\sum_{m\in M} x_{ijklm} - \sum_{n\in N} n\cdot y_{jkln} = 0, \text{ for all i, j, k; } w_{ijk} = 1 \quad (2.08)$$

$$\sum_{j\in J}\sum_{k\in K}\sum_{m\in M} x_{ijklm} <= HD, \text{ for all i,l} \quad (2.09)$$

$$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K} x_{ijklm} <= R, \text{ for all l,m} \quad (2.10)$$

$$x_{ijklm} = \begin{cases} 1, \text{ if faculty member i is assigned} \\ \quad \text{to handle course j with section k,} \\ \quad \text{on day l and time slot m} \\ 0, \text{ otherwise} \end{cases} \quad (2.11)$$

$$y_{jkln} = \begin{cases} 1, \text{ if course j requiring N (with n<=N)} \\ \quad \text{consecutive hours, is scheduled} \\ \quad \text{on day l for section k} \\ 0, \text{ otherwise} \end{cases} \quad (2.12)$$

where,

c_m = function relating the cost to the time slot
U_i = upper limit on the teaching load of faculty member i
L_i = lower limit on the teaching load of faculty i
N = number of consecutive hours a course should be taught within a given day
H_j = number of weekly hours course j should be taught
HD = limit on the number of daily working hours of each faculty member
R = number of lecture rooms available to the department each time of the day

Solving the models

We solved the 0–1 integer linear programming models for the first and second stages of the problem using LINGO 14 [11] on a Veriton PC with processor Intel (I) Core™ i5-2300 CPU @ 2.80GHz. Run times were negligible at 0.07 second for the first stage model and 1.21 seconds for the stage 2 model.

Data

Data from the subject school in the first semester, AY 2012–2013, were used to illustrate how the models work. The study focused on the professional courses of the Industrial Engineering Department; these are the courses which, for class and faculty scheduling purposes, are within the direct control of the Department Chair. The relevant data are found in Table 1 and Table 2.

Table 1. Course and section assignments of faculty members

Course	Faculty Member											
	1	2	3	4	5	6	7	8	9	10	11	12
1											3-A,B.C	
2				3-A,B,C								
3										3-A,B,C		
4									4-A,B			
5										4-A,B		
6								4-A,B				
7		4-B						4-A				
8				4-B					4-A			
9		4-A			4-B							
10								4-A,B				
11							5-A,B					
12										5-A,B		
13							5-A,B					
14						5-A,B						
15												5-A,B
16	5-B		5-A									
17												5-A,B
18		5-A,B										
19										5-A,B		

Table 3. Results from stage 1 model: course and section assignments of faculty members

Course	Faculty Member											
	1	2	3	4	5	6	7	8	9	10	11	12
1											3-A,B.C	
2				3-A,B,C								
3										3-A,B,C		
4									4-A,B			
5										4-A,B		
6								4-A,B				
7		4-A,B										
8				4-A					4-B			
9					4-A,B							
10								4-A,B				
11							5-A,B					
12										5-A,B		
13							5-A,B					
14						5-A,B						
15												5-A,B
16	5-A		5-B									
17												5-A,B
18		5-A,B										
19										5-A,B		

Table 2. Credit units and number of sections per course

Course	Year Level	Units	No. of Sections
1	3	2	3
2	3	3	3
3	3	3	3
4	4	4	2
5	4	3	2
6	4	1	2
7	4	1.5	2
8	4	3	2
9	4	3	2
10	4	2	2
11	5	2	2
12	5	1.5	2
13	5	3	2
14	5	3	2
15	5	1.5	2
16	5	3	2
17	5	2	2
18	5	1	2
19	5	3	2

Results

The results are presented in Table 3 for the first stage model, and in Table 4(A-G), for the second stage model.

The faculty-course-section assignment in Table 3 very closely matches the actual assignment shown in Table 1. The three differences in the figures of Table 3 *vis-à-vis* those of Table 1 all refer to shifts in section assignment of certain faculty-course pairs. Nonetheless, the objective of assigning faculty members according to their specializations remained upheld.

For the stage 2 model, which finds a weekly timetable for the faculty load generated in the first stage, our results show classes scheduled in the early time slots, as much as possible (see Table 4[A-G]). This is a consequence of the fact that the cost function used in the objective function favors scheduling classes early on during the day, which is the general preference of faculty members. Late afternoon and evening classes, scheduled out of need or as requested by some faculty members who teach on a part time basis, have been made possible through the inclusion of appropriate constraints.

The schedules in Table 4(A-G) are sparse, the reason being that we included only the Department's core and professional courses in the calculations. The gaps in the schedules provide available time slots for the non-core/ professional courses serviced by other departments.

Conclusion

We have presented a two-stage approach to solve the class timetabling problem for an academic department. The first stage consisted in finding a desired faculty assignment to courses opened in various sections, with the assignment based on the course specializations

Table 4(A-G). Results From Stage 2 Model

The entries are course numbers consistent with those used in Table 1 to Table 3.

A: 3rd Year Section A
Partial Class Schedule

	Mon	Tue	Wed	Thu	Fri	Sat
7–8A.M.				1	3	
8–9A.M.		1		2		
9–10A.M.		2	3			
10–11A.M.	3					
11–12NN.			2			
12–1P.M.						
1–2P.M.						
2–3P.M.						
3–4P.M.						
4–5P.M.						
5–6P.M.						
6–7P.M.						
7–8P.M.						
8–9P.M.						

B: 3rd Year Section B
Partial Class Schedule

	Mon	Tue	Wed	Thu	Fri	Sat
7–8A.M.						
8–9A.M.	3				2	
9–10A.M.				1		
10–11A.M.			2		3	
11–12NN.	1	3		2		
12–1P.M.						
1–2P.M.						
2–3P.M.						
3–4P.M.						
4–5P.M.						
5–6P.M.						
6–7P.M.						
7–8P.M.						
8–9P.M.						

C: 3rd Year Section C
Partial Class Schedule

	Mon	Tue	Wed	Thu	Fri	Sat
7–8A.M.	2	1			2	
8–9A.M.	1		3			
9–10A.M.					3	
10–11A.M.		2		3		
11–12NN.						
12–1P.M.						
1–2P.M.						
2–3P.M.						
3–4P.M.						
4–5P.M.						
5–6P.M.						
6–7P.M.						
7–8P.M.						
8–9P.M.						

D: 4th Year Section A
Partial Class Schedule

	Mon	Tue	Wed	Thu	Fri	Sat
7–8A.M.		10	10			
8–9A.M.						
9–10A.M.		6				
10–11A.M.					8	
11–12NN.					5	
12–1P.M.	8	8	5	5		
1–2P.M.						
2–3P.M.						
3–4P.M.						
4–5P.M.				4		
5–6P.M.				4		
6–7P.M.	9	4	4			
7–8P.M.	9					
8–9P.M.	9					

E: 4th Year Section B
Partial Class Schedule

	Mon	Tue	Wed	Thu	Fri	Sat
7–8A.M.	6					
8–9A.M.			10	5		
9–10A.M.	3					
10–11A.M.	10					
11–12NN.			5			
12–1P.M.						
1–2P.M.						
2–3P.M.						
3–4P.M.						
4–5P.M.		8	4		4	
5–6P.M.		8	4		4	
6–7P.M.		9			8	
7–8P.M.		9				
8–9P.M.		9				

F: 5th Year Section A
Partial Class Schedule

	Mon	Tue	Wed	Thu	Fri	Sat
7–8A.M.			11			16
8–9A.M.					13	16
9–10A.M.	13		18			16
10–11A.M.			17	13		14
11–12NN.		17				14
12–1P.M.				11		14
1–2P.M.						
2–3P.M.						
3–4P.M.						
4–5P.M.						
5–6P.M.						
6–7P.M.						
7–8P.M.						
8–9P.M.						

G: 5th Year Section B
Partial Class Schedule

	Mon	Tue	Wed	Thu	Fri	Sat
7–8A.M.				11		14
8–9A.M.		13				14
9–10A.M.				17	13	14
10–11A.M.		17				16
11–12NN.	18			13	11	16
12–1P.M.						16
1–2P.M.						
2–3P.M.						
3–4P.M.						
4–5P.M.						
5–6P.M.						
6–7P.M.						
7–8P.M.						
8–9P.M.						

of faculty members. Using data in the first semester, AY 2012–2013, from the subject academic department, our first stage model generated results which generally replicated the actual faculty assignment.

The second stage model determined the weekly schedule for the faculty-course-section assignment found in the first stage, the objective being that of scheduling classes in the earliest possible time slots. In the model, we used a cost function in the objective function which favored scheduling classes early on during the day, in compliance with the general preference of faculty members. The timetables generated also included late afternoon and evening classes, scheduled out of a special need or based on requests of faculty members who teach on a part time basis. Such special arrangements had been accommodated in the model by introducing appropriate constraints.

Computing time for each of our two 0–1 integer programming models was practically negligible, at approximately 0.07 s for the first stage model and 1.21 s for the second stage model.

Acknowledgments

The researcher gratefully acknowledges the Research Center for the Natural and Applied Sciences, University of Santo Tomas (UST), for the research support provided for this project in AY 2013–2014, in terms of research load, space and other facilities. She, likewise, extends her appreciation and thanks to the UST Faculty of Engineering for the research time allowed her.

References

[1] Abramson D. Constructing school timetables using simulated annealing: sequential and parallel algorithms. *Manage. Sci.* **1991**; 37:98–113.

[2] Yen-Zen W. Using genetic algorithm methods to solve course scheduling problems. *Expert Syst. Appl.* **2003**; 25:39–50.

[3] Mirrazavi SK, Mardle SJ, & Tamiz M. A two-phase multiple objective approach to university timetabling utilising optimisation and evolutionary solution methodologies. *J. Oper. Res. Soc.* **2003**; 54:1155–1166.

[4] Burke EK, McCollum B, Meisels A, Petrovic S, & Qu R. A graph-based hyper-heuristic for educational timetabling problems. *Eur. J. Oper. Res.* **2007**; 176:177–192.

[5] Der-Fang S. A hybrid particle swarm optimization for a university course scheduling problem with flexible preferences. *Expert Syst. Appl.* **2011**; 38:235–248.

[6] Ozdemir MS & Gasimov RN. The analytic hierarchy process and multiobjective 0–1 faculty course assignment. *Eur. J. Oper. Res.* **2004**; 157:398–408.

[7] Daskalaki S, Birbas T, & Housos E. An integer programming formulation for a case study in university timetabling. *Eur. J. Oper. Res.* **2004**; 153:117–135.

[8] Ismayilova MA, Saðir M, & Gasimov RN. A multiobjective faculty-course-time slot assignment problem with preferences. *Math. Comput. Model.* **2007**; 46:1017–1029.

[9] Al-Yakoob SM & Sherali HD. Mathematical programming models and algorithms for a class-faculty assignment problem. *Eur. J. Oper. Res.* **2006**; 173:488–507.

[10] Sorensen D & Dahms FHW. A Two-stage decomposition of high school timetabling applied to cases in Denmark. *Comput. Oper. Res.* **2014**; 43:36–49.

[11] Lindo Systems Inc., Lingo the Modeling Language and Optimizer. (Chicago, Illinois: Lindo Systems Inc., **2013**) http://www.tstat.it/software/lindo-lingo/LINGO14.PDF