Applied Mathematical Sciences, Vol. 9, 2015, no. 58, 2871 - 2880
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2015.52134

Exam Invigilators Assignment Problem: A Goal Programming Approach

Farida Hanum, Mira A. Romliyah, Toni Bakhtiar[1]

Division of Operations Research, Department of Mathematics
Bogor Agricultural University, Dramaga, Bogor 16680, Indonesia

Abstract

In this paper we considered a capacitated scheduling problem of exam invigilators. A goal programming framework was undertaken in formulating the problem, where a number of constraints related type of exam, type of invigilator, time availability, and other preferences were classified into primary and secondary constraints. The model was then applied to a simple case of exam invigilators assignment at Department of Mathematics, Bogor Agricultural University.

Mathematics Subject Classification: 90B35, 90B50, 90C29

Keywords: education timetabling, invigilator scheduling, goal programming

1 Education Timetabling

Scheduling problems deal with the issue of coupling one or several given resources to activities such as production planning, personnel planning [5], product configuration, and transportation over a certain time period, subject to specific constraints in such a way as to satisfy as nearly as possible a set of desirable objectives. Education related timetabling problem as special subclass

[1]Corresponding author

of scheduling problem involves assigning a sequence of events (normally lectures or examinations) to specific but limited number of rooms, encompassing teachers, administrative staffs and students, in a predetermined period of time [6]. It is required that the desired schedule satisfies a set of hard constraints as well as soft constraints of various types. The schedule is usually aimed to minimize student and teacher's workload.

Education timetabling can loosely be divided into school timetabling [2], course timetabling [10], and examination timetabling [8], which differ from each other by the type of events, the kind of institution involved, and the type and the relative influence of constraints [6]. In general, education timetabling problems are large and highly constrained problem, more precisely, kinds of complex combinatorial problem consisting of NP-complete subproblems [3]. However, they are subject to similar types of constraints such as overlapping, resource availability, and room capacity. Considerable interest has been devoted to research on automated timetabling during the decades and continues to attract the attention of researchers due to its characteristics of additional constraints required and better solution desired by end users making increase of the complexity of the problem. Moreover, several applications and techniques have been developed and employed with good success. It is discussed in [9] the mathematical modeling of examination scheduling problem and its parameter estimation by using the Analytic Network Process (ANP) approach. A so-called constructive heuristic was proposed by [8] in solving timetabling problem of certain university with good quality solutions, which are superior to the existing ones. A particle swarm based hyper-heuristic approach was applied by [1] in the similar problem. Benders' partitioning method was applied by [10]. It is also well-known that the general timetabling problem can easily be modeled within the graph coloring framework as employed by [11]. An exam scheduling software system that performs efficient, accurate and robust solution searching to solve large and complex exam scheduling problems is presented by [4].

This current paper presents an invigilator-exam assignment, a problem which receives less attention than the examination timetabling problem. We mean by the problem assigning invigilators to examination rooms in which an examination timetable is already in place, usually prepared by scheduling management unit. We consider a very specific small problem arising at the Department of Mathematics, Bogor Agricultural University. Even though the problem is small-scaled, some unique constraints are introduced. For a real world university level of invigilator scheduling problem, readers may follow the work by [7], where additional constraints were included by taking into account the comments made by the invigilators and a constructive algorithm was also presented in obtaining better solutions. While the ANP approach was used in [9] to prioritize the objectives of the invigilator-exam assignment problem, we here adopt a goal programming framework to incorporate some preferences into the model.

2 Invigilator-Exam Assignment Model

Usually, the invigilator-exam assignment problem is prepared manually at the Department of Mathematics, where the invigilator's time availability is the only concern in designing the timetable. In most cases, this manual system fails to adhere some hard constraints imposed by the scheduler. To improve the scheduling process we develop an assignment model in term of non-preemptive goal programming framework. The following assumptions are considered in the model:

1. The examination timetable is already in our hands and the examination rooms are available with sufficient capacity.

2. The examination involves undergraduate and graduate programs.

3. Examinations can be conducted on Monday till Saturday and lasted for two hours. Five time slots are available: $t_1 : 08.00 - 10.00$, $t_2 : 09.00 - 11.00$, $t_3 : 10.30 - 12.30$, $t_4 : 13.00 - 15.00$, and $t_5 : 13.30 - 15.30$. Time slots t_1, t_3, and t_5 are devoted for undergraduate exams, while time slots t_2 and t_4 are allocated for graduate exams.

4. There are three types of invigilator, namely administration staff, teaching assistant, and senior (undergraduate) student. The first priority to be assigned is staff, followed by assistant and student. It is assumed that teaching assistants and senior students can always invigilate anytime. A teaching assistant will only supervises an exam of his/her own subject.

5. The model focuses on and most of its requirements deal with administration staffs assignment. Regarding the senior student, what we need is only the required number of students to supervise.

To state our model we introduce a number of sets, which defined based on our special needs. We define by $\mathbb{E}$ set of all exams, where $\mathbb{E} = \{1, \ldots, n\}$ with n is the number of exams to be scheduled, by $\mathbb{E}_u$ set all undergraduate exams ($\mathbb{E}_u \subseteq \mathbb{E}$), by $\mathbb{E}_g$ set of all graduate exams ($\mathbb{E}_g \subseteq \mathbb{E}$ and $\mathbb{E}_u \cup \mathbb{E}_g = \mathbb{E}$), by $\mathbb{E}_e$ set of all exams conducted on 08.00-10.00 ($\mathbb{E}_e \subseteq \mathbb{E}$), by $\mathbb{E}_o$ set of all exams that conducted in the same day with either the same time slot or overlapping time slots. The followings are overlapping time slots: t_1 and t_2, t_2 and t_3, or t_4 and t_5. Next we define by $\mathbb{E}_s$ set of all exams conducted on Saturday ($\mathbb{E}_s \subseteq \mathbb{E}$), by $\mathbb{E}_d$ set of all exams conducted on day d, where $d \in \mathbb{D} := \{1, \ldots, T\}$, where

T is the length of exam period ($\mathbb{E}_d \subseteq \mathbb{E}$), and by $\mathbb{S}$ set of all administration staffs, where $\mathbb{S} = \{S_1, \ldots, S_m\}$ with m is the number of staffs.

We define following binary decision variables:

$$x_{ij} = \begin{cases} 1 & ; \text{ exam } i \text{ is invigilated by administration staff } j \\ 0 & ; \text{ otherwise.} \end{cases} \tag{1}$$

Our model considers a number of hard constraints, which generally obey the common rules or follow the rules required by the Department of Mathematics.

1. No administration staff can invigilate more than one exam concurrently, i.e., clash-free requirement:

$$\sum_{i \in \mathbb{E}_o} x_{ij} \leq 1, \quad \text{for all } j \in \mathbb{S}. \tag{2}$$

2. The number of required invigilators to supervise an exam depends on the number of participants. It is required that one invigilator supervises at most 25 participants, otherwise we need more than one invigilators:

$$P_i = \lceil M_i/25 \rceil, \quad \text{for all } i \in \mathbb{E}, \tag{3}$$

   where P_i is the number of invigilators required for exam i and M_i is the number of participants for exam i.

3. There is one and only one administration staff supervises an undergraduate exam:

$$\sum_{j \in \mathbb{S}} x_{ij} = 1, \quad \text{for all } i \in \mathbb{E}_u. \tag{4}$$

4. In addition to administration staff type invigilator, a number of senior students might be needed to supervise undergraduate exams:

$$N_i = P_i - A_i - 1, \quad \text{for all } i \in \mathbb{E}_u, \tag{5}$$

   where N_i and A_i are, respectively, the numbers of senior students and teaching assistants required for exam i.

5. Neither teaching assistant nor senior student allowed for supervising graduate exams:

$$\sum_{j \in \mathbb{S}} x_{ij} = P_i, \quad \text{for all } i \in \mathbb{E}_g. \tag{6}$$

6. Each staff invigilates exams no more than k times a day:

$$\sum_{i \in \mathbb{E}_d} x_{ij} \leq k, \quad \text{for all } d \in \mathbb{D} \text{ and for all } j \in \mathbb{S}. \tag{7}$$

7. Two administration staffs, let say S_k and S_ℓ, due to their duties, are not allowed to supervise exam either in the same or overlapping time slots:

$$\sum_{i\in\mathbb{E}_o}\sum_{j\in\{S_k,S_\ell\}} x_{ij} \leq 1, \quad \{S_k, S_\ell\} \subset \mathbb{S}. \tag{8}$$

8. Due to acceptable reasons, a staff, let say $S_k \in \mathbb{S}$, cannot supervise exam on 08.00 and another staff, let say $S_\ell \in \mathbb{S}$, is unable to supervise exam on Saturday:

$$x_{iS_k} = 0, \quad \text{for all } i \in \mathbb{E}_e, \tag{9}$$
$$x_{iS_\ell} = 0, \quad \text{for all } i \in \mathbb{E}_s. \tag{10}$$

In addition to hard constraints, we also take into account a set of soft constraints which reflect the preference of staffs in undertaking exam supervisory. In this work we consider two fairness aspects and formulate them into goal functions.

1. Each staff has almost the same number of invigilating tasks in one scheduling period:

$$\sum_{i\in\mathbb{E}} x_{ij} + \alpha_j^- - \alpha_j^+ = \frac{1}{m}\sum_{i\in\mathbb{E}}\left(P_i - A_i - N_i\right), \quad \text{for all } j \in \mathbb{S}. \tag{11}$$

2. Each staff has almost the same number of invigilating tasks on 08.00-10.00 time slot:

$$\sum_{i\in\mathbb{E}_e} x_{ij} + \beta_j^- - \beta_j^+ = \frac{1}{q}\sum_{i\in\mathbb{E}_e}\left(P_i - A_i - N_i\right), \quad \text{for all } j \in \mathbb{S}, \tag{12}$$

where q is the number of staff which are available on 08.00-10.00 time slot.

As objective function we consider to minimize all deviations around the goals given in (11) and (12):

$$\min\ Z := \sum_{j\in\mathbb{S}}\left(a_{1j}\alpha_j^- + a_{2j}\alpha_j^+ + b_{1j}\beta_j^- + b_{1j}\beta_j^+\right), \tag{13}$$

where a_{1j}, a_{2j}, b_{1j}, and b_{2j} are weights that indicate the relative importance among goals.

3 Model Implementation

In this section we present the implementation of the model into a simple case of invigilator-exam assignment problem in Department of Mathematics, Bogor Agricultural University. In odd semester, the Department organizes 44 exams, which consist of 38 undergraduate exams and 6 graduate exams and will be supervised by 6 administration staffs. The exam timetable in term of time and room is made available by the University Scheduling Office, see Table 1. The period of exam is two weeks, which is equivalent to 10 exam days.

From Table 1 we can define following sets:

$$\begin{aligned}
\mathbb{E} &= \{1, \ldots, 44\}, \\
\mathbb{E}_g &= \{2, 8, 12, 28, 31, 40\}, \\
\mathbb{E}_u &= \mathbb{E} - \mathbb{E}_g, \\
\mathbb{E}_e &= \{1, 6, 7, 15, 16, 17, 18, 25, 26, 27, 32, 33, 41, 42, 43, 44\}, \\
\mathbb{E}_o &= \{\{1, 2\}, \{2, 3, 4\}, \{6, 7, 8\}, \{8, 9, 10, 11\}, \ldots, \{41, 42, 43, 44\}\}, \\
\mathbb{E}_s &= \{24\}, \\
\mathbb{E}_1 &= \{1, 2, 3, 4, 5\}, \\
&\vdots \\
\mathbb{E}_{10} &= \{41, 42, 43, 44\}.
\end{aligned}$$

We also find in the same table that M_i and A_i are, respectively, given by columns 7 and 8. P_i and N_i will be determined by the model. The following value of parameters are also applied: $n = 44$, $m = 6$, $T = 10$, $q = 5$, $k = 2$, $a_{1j} = a_{2j} = 1$, and $b_{1j} = b_{2j} = 2$. We assume that two administration staffs S_1 and S_2 are not allowed to supervise exam either in the same or overlapping time slots since their administrative tasks are similar, thus at least one of them should always present at his/her desk. Additionally, S_5 cannot supervise exam on 08.00 everyday and on Saturday.

Table 2 presents the invigilator timetables, which constitutes the main output of the model. We can see that, to organize all exams, we need a number of 30 teaching assistants and 36 senior students. The needs for each exam are provided in column 6 for teaching assistant (A_i) and in column 7 for senior student (N_i). The total number of invigilators required for each exam (P_i), including the staff, is given in the last column.

From the view point of goals accomplishment, it can be seen in the left-hand pane of Table 3 that all exam supervision tasks can be equally distributed to staffs, where each staff has to supervise exams 8 times during the period. In particular, staff S_3 has a higher number of 08.00 tasks, i.e., task at time slot t_1, than other staffs by one. In this case, S_5 was intentionally left restrained for doing this task as indicated by (10). How far did they deviate from the

Table 1: The exam timetable

No	Day	Day Name	Time	Course	Program	Participant	Assistant
1	1	Monday	t_1	MAT223	Undergraduate	79	2
2	1	Monday	t_2	MAT511	Graduate	35	
3	1	Monday	t_3	MAT453-1	Undergraduate	46	1
4	1	Monday	t_3	MAT453-2	Undergraduate	37	1
5	1	Monday	t_5	MAT354	Undergraduate	74	
6	2	Wednesday	t_1	MAT212-1	Undergraduate	45	1
7	2	Wednesday	t_1	MAT212-2	Undergraduate	44	1
8	2	Wednesday	t_2	MAT633	Graduate	4	
9	2	Wednesday	t_3	MAT331-1	Undergraduate	32	1
10	2	Wednesday	t_3	MAT331-2	Undergraduate	31	1
11	2	Wednesday	t_3	MAT331-3	Undergraduate	31	
12	2	Wednesday	t_4	MAT521	Graduate	35	
13	2	Wednesday	t_5	MAT442-1	Undergraduate	42	
14	2	Wednesday	t_5	MAT442-2	Undergraduate	62	
15	3	Thursday	t_1	MAT217-1	Undergraduate	71	
16	3	Thursday	t_1	MAT217-2	Undergraduate	70	
17	3	Thursday	t_1	MAT341-1	Undergraduate	50	
18	3	Thursday	t_1	MAT341-2	Undergraduate	49	
19	3	Thursday	t_3	MAT421	Undergraduate	84	1
20	3	Thursday	t_5	MAT221-1	Undergraduate	94	
21	3	Thursday	t_5	MAT221-1	Undergraduate	117	
22	4	Friday	t_2	MAT314-1	Undergraduate	37	
23	4	Friday	t_2	MAT314-2	Undergraduate	37	
24	5	Saturday	t_5	STK211	Undergraduate	79	2
25	6	Monday	t_1	MAT219-1	Undergraduate	85	
26	6	Monday	t_1	MAT219-2	Undergraduate	86	
27	6	Monday	t_1	MAT219-3	Undergraduate	43	
28	6	Monday	t_2	MAT641	Graduate	4	
29	6	Monday	t_3	MAT311	Undergraduate	23	
30	6	Monday	t_5	MAT431	Undergraduate	34	1
31	7	Tuesday	t_2	MAT513	Graduate	35	
32	8	Wednesday	t_1	MAT252-1	Undergraduate	40	1
33	8	Wednesday	t_1	MAT252-2	Undergraduate	39	1
34	8	Wednesday	t_3	MAT451-1	Undergraduate	41	1
35	8	Wednesday	t_3	MAT451-2	Undergraduate	40	1
36	8	Wednesday	t_5	MAT321-1	Undergraduate	69	2
37	8	Wednesday	t_5	MAT321-2	Undergraduate	49	1
38	8	Wednesday	t_5	MAT321-3	Undergraduate	48	1
39	8	Wednesday	t_5	MAT321-4	Undergraduate	80	2
40	9	Thursday	t_2	MAT551	Graduate	35	
41	10	Friday	t_1	MAT211-1	Undergraduate	63	2
42	10	Friday	t_1	MAT211-2	Undergraduate	60	2
43	10	Friday	t_1	MAT211-3	Undergraduate	82	2
44	10	Friday	t_1	MAT211-4	Undergraduate	85	2
							30

corresponding goal? It is confirmed by the output of the model that $\sum_j \alpha_j^- = \sum_j \alpha_j^+ = 0$ and $\sum_j \beta_j^- = \sum_j \beta_j^+ = 0.8$, and thus $Z^* = 3.2$.

4 Concluding Remark

In this paper we have presented an invigilator-exam assignment model which formulated in the form of non-preemptive goal programming framework. This simple model has been successfully applied in invigilators assignment at the De-

Table 2: The invigilator timetable

No	Day	Time	Participant	Invigilator			
				Staff	**Assistant**	**Student**	**Total**
1	1	t_1	79	S_2	2	1	4
2	1	t_2	35	S_3, S_5			2
3	1	t_3	46	S_2	1		2
4	1	t_3	37	S_4	1		2
5	1	t_5	74	S_1		2	3
6	2	t_1	45	S_2	1		2
7	2	t_1	44	S_3	1		2
8	2	t_2	4	S_4			1
9	2	t_3	32	S_5	1		2
10	2	t_3	31	S_3	1		2
11	2	t_3	31	S_1		1	2
12	2	t_4	35	S_4, S_5			2
13	2	t_5	42	S_6		1	2
14	2	t_5	62	S_1		2	3
15	3	t_1	71	S_4		2	3
16	3	t_1	70	S_6		2	3
17	3	t_1	50	S_2		2	3
18	3	t_1	49	S_3		1	2
19	3	t_3	84	S_3	1	2	4
20	3	t_5	94	S_6		3	4
21	3	t_5	117	S_1		4	5
22	4	t_2	37	S_6		1	2
23	4	t_2	37	S_2		1	2
24	5	t_5	79	S_6	2	1	4
25	6	t_1	85	S_4		3	4
26	6	t_1	86	S_1		3	4
27	6	t_1	43	S_3		1	2
28	6	t_2	4	S_5			1
29	6	t_3	23	S_2			1
30	6	t_5	34	S_1	1		2
31	7	t_2	35	S_4, S_5			2
32	8	t_1	40	S_6	1		2
33	8	t_1	39	S_1	1		2
34	8	t_3	41	S_2	1		2
35	8	t_3	40	S_5	1		2
36	8	t_5	69	S_2	2		3
37	8	t_5	49	S_6	1		2
38	8	t_5	48	S_5	1		2
39	8	t_5	80	S_3	2	1	4
40	9	t_2	35	S_4, S_5			2
41	10	t_1	63	S_3	2		3
42	10	t_1	60	S_6	2		3
43	10	t_1	82	S_4	2	1	4
44	10	t_1	85	S_1	2	1	4
					30	**36**	

partment of Mathematics, Bogor Agricultural University. It is demonstrated that the model can solve the scheduling problem efficiently. Beyond the common hard requirements, the model offers more fairness by taking into account some preferences related to the equity of the number of supervision tasks and those in the first time slot, which normally undesirable.

Developing such kind of model enables us to consider different scenarios of scheduling, for instance, due to the shortage of human resources. Suppose that a staff, let say S_6, is no longer available for supervising an exam. Then we must

Table 3: The goal accomplishment with six and five staffs

	Goal with 6 Staffs		Goal with 5 Staffs	
Staff	**Total Tasks**	**Total t_1 Tasks**	**Total Tasks**	**Total t_1 Tasks**
S_1	8	3	10	4
S_2	8	3	9	4
S_3	8	4	9	4
S_4	8	3	10	4
S_5	8	×	10	×
S_6	8	3	×	×
	48	**16**	**48**	**16**

reschedule the assignment by only possessing five staffs. This problem can be undertaken either by setting $m = 5$ hence $\mathbb{S} = \{S_1, \ldots, S_5\}$, or by adding extra constraints $x_{iS_6} = 0$, for all $i \in \mathbb{E}$, both in addition to $q = 4$. Quick inspection by running the model confirmed that this problem was unfeasible. Thus it is imperative to modify the model such as by relaxing a hard constraint. We may, for instance, relax the requirement that S_1 and S_2 must not supervise an exam in the same or overlapping time slots. It can be accomplished by converting (8) into a goal constraint:

$$\sum_{i\in\mathbb{E}_o} \sum_{j\in\{S_1,S_2\}} \left(x_{ij} + \gamma_j^- - \gamma_j^+\right) = 1. \tag{14}$$

The objective function (13) is then adjusted to

$$\min\ \hat{Z} := Z + \sum_{j\in\{S_1,S_2\}} \left(c_{1j}\gamma_j^- + c_{2j}\gamma_j^+\right). \tag{15}$$

Under the choice of weights $c_{1j} = c_{2j} = 2$, the right-hand pane of Table 3 shows that the model can now allocated tasks quite well and further exploration reveals that S_1 and S_2 supervise together only in three occasions, one in overlapping and two in the same time slots, but in different exams.

References

[1] A. Ahmed, M. Ali, A. Sajid, and A. H. S. Bukhari, Particle swarm based hyper-heuristic for tackling real world examinations scheduling problem, *Australian Journal of Basic and Applied Sciences*, **5(10)** (2011), 1406–1413. ISSN: 1991-8178.

[2] T. Birbas, S. Daskalaki, and E. Housos, School timetabling for quality student and teacher schedules, *Journal of Scheduling*, **12** (2009), 177–197. http://dx.doi.org/10.1007/s10951-008-0088-2

[3] M.W. Carter and G. Laporte, Recent developments in practical course timetabling, in: Selected Papers from the Second International Conference on the Practice and Theory of Automated Timetabling II, *Lecture Notes in Computer Science*, **1408** (1998), 3–19. http://dx.doi.org/10.1007/bfb0055878

[4] Z. Chunbao and T. Nu, An intelligent, interactive & efficient exam scheduling system (IIEESS v1.0), *Proceeding of the Practice and Theory of Automated Timetabling (PATAT), Norway*, (2012), 437–450. ISBN: 978-82-14-05298-5.

[5] A. T. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier, Staff scheduling and rostering: a review of applications, methods, and models, *European Journal of Operational Research*, **153** (2004), 3–27. http://dx.doi.org/10.1016/s0377-2217(03)00095-x

[6] L. D. Gaspero, Local search techniques for scheduling problems: algorithms and software tools, *Ph.D. Thesis, Universita degli Studi di Udine*, 2003. ISBN: 88-8420-151-9.

[7] M. N. M. Kahar and G. Kendall, Universiti Malaysia Pahang examination timetabling problem: scheduling invigilators, *Journal of the Operational Research Society*, **65** (2014), 214–226. http://dx.doi.org/10.1057/jors.2012.6

[8] M.N.M. Kahar and G. Kendall, The examination timetabling problem at Universiti Malaysia Pahang: comparison of a constructive heuristic with an existing software solution, *European Journal of Operational Research*, **207(2)** (2010), 557–565. http://dx.doi.org/10.1016/j.ejor.2010.04.011

[9] M. Sagir and Z. K. Ozturk, Exam scheduling: Mathematical modeling and parameter estimation with the analytic network process approach, *Mathematical and Computer Modelling*, **52** (2010), 930–941. http://dx.doi.org/10.1016/j.mcm.2010.05.029

[10] S.C. Sarin, Y. Wang, and A. Varadarajan, A university timetabling problem and its solution using Benders' partitioning—a case study, *Journal of Scheduling*, **13** (2010), 131–141. http://dx.doi.org/10.1007/s10951-009-0157-1

[11] M. A. Selemani, E. Mujuni, and A. Mushi, An examination scheduling algorithm using graph colouring—the case of Sokoine University of Agriculture, *International Journal of Computer Engineering & Applications*, **2(1/3)** (2013), 116–127. ISBN: 2321-3469.

Received: March 2, 2015; Published: April 12, 2015