Prototype System Development for Examination Proctor Assignment Problem Using Mixed Integer Programing

Takeshi Koide and Kana Iwata, *Member, IAENG*

***Abstract*—In our university, faculties and academic staffs are assigned as proctors to written examinations conducted in regular examination periods. The proctor assignment task for huge amount of examinations is time-consuming and has been conducted manually. In our study, the assignment task is mathematically modeled in a mixed integer programing form and a prototype system is developed to derive an optimal solution for the model. This paper introduces the model and the developed system. Numerical results derived by utilizing practical data are also shown to reveal the effectiveness of the developed system.**

***Index Terms*—mathematical model, mixed integer programing, operations research, optimal assignment**

I. Introduction

Due to the rapid advance in the performance of personal computer and optimization software, optimization techniques have been applied more widely recently to various sorts of practical operations in many industries by using personal computers and optimization software. Suzuki *et al* [1] set up a project team to promote efficiency in several operations in their university by applying techniques in operations research, especially optimization.

This paper discusses a kind of university operation, proctor assignment. In our university, regular examinations are conducted at the end of academic semesters. Written examinations are usually conducted in classrooms and some faculties and academic staffs are assigned to the examinations as proctors in order to prevent students' unjust acts. The proctor assignment task is carried out by staffs in the registrar section with considering some conditions such as schedules of faculties and staffs, required number of proctors for each examination, satisfaction with resulting assignments for faculties and staffs. The assignment task is time-consuming because many courses adopt the written examinations.

The proctor assignment is a part of the operation called examination timetabling, where both examination scheduling and classroom assignment are conducted in addition to the proctor assignment. The timetabling problem is generally so complex that approximate solutions are derived through heuristic algorithms [2, 3] or meta-heuristic approaches such as [4-6]. Such approaches explore adequate solutions within a reasonable time but it is hard to comprehend their mechanisms for users who are unfamiliar with optimization.

From a viewpoint of continuous use of developed systems, linear programing approach is welcomed for general users due to its simplicity to understand reasoning of derived solutions. Dammak *et al* [7] proposed a zero-one linear integer programing model for classroom assignment problem. Onouchi *et al* [8] treated the proctor assignment as a part of timetabling of university regular examinations. They divided the timetabling into two parts. In the first part, examination schedule and classroom assignment are executed, and the proctor assignment is determined in the second step.

This study has been constructed a mathematical model for the examination proctor assignment task in our university as a combinatorial optimization problem in consideration with the model by Onouchi *et al* in their university [8]. This study has also developed a prototype system for the assignment task. The system is based on spreadsheet software and an optimal assignment is derived by solving the optimization problem using external optimization software. This paper explains the proposed optimization problem and the details of the developed system. Some results in numerical experiments using practical data are also represented to show the effectiveness of the developed system. Some difficulties in integration of the proposed system to practical operations and future tasks are also discussed.

II. Examination Proctor Assignment Task

Over two thousands courses are offered in an academic semester at the main campus in our university. Approximately one thousand courses adopt written examinations in the regular examination period. When a course conducts an examination, the faculty in charge of the course is assigned to the examination as a chief proctor. An examination requires a proctor in every forty students. Around seven hundred courses have forty students or fewer and hence the examinations for such courses are operated only by a chief proctor. To the other around three hundred courses, some full-time faculties and/or academic staffs are assigned as assistant proctors. A part-time teacher is only assigned to the examination for his/her courses as a chief proctor and is not assigned to the other examinations as an assistant.

The examination proctor assignment task determines who is assigned to which examinations. Full-time faculties and

Manuscript received July 03, 2014. This work was partly supported by Grant-in-Aid for Scientific Research (B) No. 25285131 and MEXT, Japan.

Takeshi Koide is with Department of Intelligence and Informatics, Konan University, Kobe 658-8501, Japan (corresponding author to provide phone and fax: +81-78-435-2532; e-mail: koide@konan-u.ac.jp).

Kana Iwata is with Tokyo Computer Service Co. Ltd., Kobe 650-0021, Japan (e-mail: si071022@center.konan-u.ac.jp).

academic staffs are referred to as members in the followings. Each day has six periods and the examinations for all courses are assumed to be scheduled in advance. The chief proctors are then automatically determined due to the examination schedule.

The assignment task for assistant proctors considers the following conditions. First, a member should be assigned to the examinations curried out on his/her available dates. The availability for around two fifty members are ascertained before the assignment task starts. A member is assigned to around five examinations on average in a semester including the assignments as a chief proctor. The number of assignments for a member in a day is also restricted up to three whereas there are six periods in a day because the proctor task is punishing work for most members. The assignments in two consecutive periods in a day should be avoided from the same reason. At the same time, two assignments between which there are three or four periods should be avoided for members' conveniences.

III. Mathematical Modeling

In this section, the examination proctor assignment task is modeled as a combinatorial optimization problem. The sets, constants, and variables in the problem are defined in the followings. Here, the examination slot is referred to as a period with consecutive index inducted by both a date and a period. Concretely, the slot $6k + t$ means the t-th period in the k-th date for $t = 1, 2, \ldots, 6$ and k is positive integer.

A. Nomenclature

Sets:

- I index set for members to be assigned
- J index set for courses
- K index set for days in the examination period
- T index set for examination slots

Constants:

- u_i upper limit of assignments for member i
- r_i required number of assignments for member i
- n_j number of members to be assigned to the examination for course j
- a_{it} 1 if member i can be assigned to an examination conducted in slot t, 0 otherwise
- b_{jt} 1 if the examination for course j is conducted in slot t, 0 otherwise
- c_{ij} 1 if member i teaches course j, namely member i is the chief proctor in the examination for course j, 0 otherwise

Decision Variables:

- x_{ij} 1 if member i is assigned to the examination for course j, 0 otherwise
- y_{it} 1 if member i is assigned to an examination conducted in slot t, 0 otherwise
- z_{ih} penalty imposed when member i violates the h-th soft constraint

B. Formulation

By using the definitions in the previous subsection, the examination proctor assignment procedure is formulated as an optimization problem, named Problem EPA, as follows:

<u>Problem EPA</u>:

$$\text{Minimize} \quad \sum_{i\in I}\sum_{\substack{h=1\\h\neq 2}}^{8} z_{ih} \tag{1}$$

subject to

$$y_{it} \le a_{it}, \quad i\in I, t\in T \tag{2}$$

$$x_{ij} \ge c_{ij}, \quad i\in I, j\in J \tag{3}$$

$$\sum_{i\in I} x_{ij} = n_j, \quad j\in J \tag{4}$$

$$\sum_{j\in J} b_{jt}\, x_{ij} = y_{it}, \quad i\in I, t\in T \tag{5}$$

$$r_i \le \sum_{t\in T} y_{it} \le u_i, \quad i\in I \tag{6}$$

$$y_{i\{6k+h\}} + y_{i\{6k+h+1\}} \le 1 + z_{ih}, \quad i\in I, k\in K, h = 1,2,\ldots,5 \tag{7}$$

$$y_{i\{6k+1\}} - \sum_{l=2}^{4} y_{i\{6k+l\}} + y_{i\{6k+5\}} \le 1 + z_{i6}, \quad i\in I, k\in K \tag{8}$$

$$y_{i\{6k+2\}} - \sum_{l=3}^{5} y_{i\{6k+l\}} + y_{i\{6k+6\}} \le 1 + z_{i7}, \quad i\in I, k\in K \tag{9}$$

$$y_{i\{6k+1\}} - \sum_{l=2}^{5} y_{i\{6k+l\}} + y_{i\{6k+6\}} \le 1 + z_{i8}, \quad i\in I, k\in K \tag{10}$$

$$x_{ij} \in \{0,1\}, \quad i\in I, j\in J \tag{11}$$

$$y_{it} \in \{0,1\}, \quad i\in I, t\in T \tag{12}$$

$$z_{ih} \ge 0, \quad i\in I, h = 1,2,\ldots,8 \tag{13}$$

The objective function (1) in Problem EPA minimizes the sum of penalties imposed by unsatisfied soft constraints shown as Inequalities (7) through (10). The meaning of the exception of $h = 2$ are mentioned later.

Inequality (2) implies the constraint for availability of members; member i can be assigned to a course examination conducted in his/her available slots. Inequality (3) represents the constraint with respect to chief proctors; member i who teaches course j must be assigned to its examination as a chief proctor. Equation (4) defines the required proctor size for course j. The decision variable y_{it} is determined the value of another type of decision variable x_{ij} by using Equation (5). Thanks to Equation (5) and Condition (12), a member is never assigned to more than one class at the same slot. The upper limit and the required number of assignments for member i are represented by Inequalities (6). Inequalities (7) through (10) are soft constraints, whereas conditions (2) through (5) are normal constraints. Inequality (7) prohibits assignments of member i to two consecutive slots in a certain date if possible unless the slots are in the second period and

ISBN: 978-988-19253-7-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

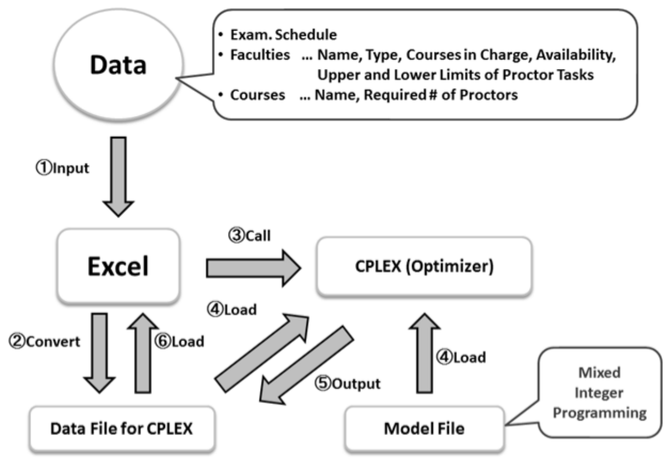

Fig. 1. System flow diagram of the developed system

the third period because there is lunch break between the periods. When member i is assigned to more than one slots in a certain date, more than two blank slots are also prohibited between his/her assigned slots if possible, which is modeled by Inequalities (8) through (10). Conditions (11) through (13) show constraints for all the design variables in Problem EPA. As a result, Problem EPA is modeled in an integer programming form.

IV. Developed System and Implementation

The developed system is coded by VBA (Visual Basic for Application) based on Microsoft Excel, spreadsheet software. The required data for Problem EPA are inputted into spreadsheets and the problem is then solved by CPLEX [9], commercial optimization software, and finally the resulting assignment is displayed in a spreadsheet to a user. Figure 1 illustrates the system flow diagram for the system.

The developed system requires fundamental constants in the first step, such as the number of members to be assigned $|I|$, number of courses $|J|$, and number of days $|K|$. The number of examination slots $|T|$ is given as $6|K|$ because there are six periods a day. In the second step, the list of members is required. The user of the system inputs the name, lower limit r_i and upper limits u_i of assignments, and the attribute of members, where the attribute is either full-time faculty, part-time faculty, or academic staff. In the third step, dates for the $|K|$ days in the examination period are inputted and then $|T|$ examination slots are automatically created. In the next step, the information associated with examinations is required, such as the name of courses, the number of required proctors n_j, the scheduled slot, the faculty in charge. Finally, the availability of members is inputted to the system. Based on the request of the charged staff in registrar section, the availability is designed to be inputted in dates because the availability in slots is six times more complex than that in dates.

The developed system converts some inputted data to the values of the constants in Problem EPA, such as a_{it}, b_{jt}, and c_{ij}. Some inputted data and converted data are written down to a data file for CPLEX. The mathematical model for Problem EAP is represented in a model file in advance. CPLEX loads the date file, the model file, and a setting file for an execution of optimization. The resulting optimum solution by CPLEX is outputted to the data file. The system loads the optimal solution and displays it in a designed layout on some spreadsheets. One layout is designed for members to clarify which courses a member is assigned to. Another layout is for courses to represents their chief proctor and assistant proctors.

TABLE I
Utilized Practical Data

	Date Set I	Data Set II
semester	2013 Spring	2013 Fall
number of members $\|I\|$	28	869
number of courses $\|J\|$	43	998
number of days $\|K\|$	9	10

TABLE II
Parameter Setting and Results

	Date Set I	Data Set II
lower limit of assignments r_i	1	1
upper limit of assignments u_i	3	9
the value of objective function	3	163
computational time for optimization	0.3 sec.	0.75 sec.

V. Numerical Result and Discussion

The developed system has been executed using practical data in order to estimate its performance. The summary of the utilized practical data is shown in Table 1. Data set I and II are just for our department and for all departments in our university, respectively. The value of a_{it}, that is, member's availability is estimated by the academic schedule. For instance, if a faculty teaches courses provided in Tuesday and Friday, he/she can be assigned to examination conducted in Tuesday and Friday in the examination period.

The system was first estimated by using data set I as a small data. Lower limit r_i and upper limit u_i of assignments were set to the common value shown in Table 2. The value of objective function (1) was three, which means that a faculty teaches three courses in the third period through the fifth period sequentially and another faculty offers two courses in two consecutive periods. In other words, all soft constraints are satisfied except for the unavoidable conditions. The resulting assignment was demonstrated to the staffs in the registrar section and they recommended us to estimate the system continuously by using large quantities of data, data set II.

The optimization software CPLEX derived the optimal assignment for data set II in 0.75 seconds, which means that the system derived an optimal solution in a short time even for the assignment targeted at all departments. The value of the objective function was 163 and virtually all avoidable soft constraints were satisfied as in data set I. The system, however, required several minutes to execute the other procedures, especially writing the values of constants down to the data file. Other detailed discussion on computational results is in progress and it will be shown in the conference presentation.

Some implemental difficulties have been found through

ISBN: 978-988-19253-7-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the experiments. Especially, improvement on system usability should be essential to input required data. All names of members and courses are necessary to execute the assignment. It is anticipated to introduce data conversion functions derived from existing database in an enterprise system. As for the availability of members, default values should be created from the regular academic timetable, that is, the days when a faculty teaches courses are set to be available by default.

Some future tasks have been revealed by the implementation of the system using practical data sets. All of avoidable soft constraints are satisfied under the current environment. Hence, the objective function enables to incorporate different factors such as departments members belong to. In fact, some faculties show a preference for the proctors for their department's courses than those for the courses in the other departments. While the staffs in the registrar section have no afford to take such a factor into account now, the proposed system enables to incorporate the factor into the mathematical model.

The proctor assignment task is a part of the timetabling for regular examinations. Ahead of the proctor assignment, the staffs in the registrar section have to determine an examination schedule, an assignment of classrooms to examinations, partitioning of students according to the number of classes for courses with large number of students. These tasks including the proctor assignment affect mutually and their tasks should be accomplished concurrently to improve total efficiency.

VI. Conclusion

This paper introduced a prototype spreadsheet-based system for an examination proctor assignment task. The task is modeled mathematically as a mixed integer programing problem which is preferable for general users to comprehend system mechanism. The model considers availability of assigned members as well as load balance for assigned members charged by the assignment. Resulting assignments outputted by the developed system are desirable for practical datasets.

In the future, the system should be upgraded to be convenient for system users and to tackle more various requests. The system, moreover, should be extended to deal with the whole task on the regular examination timetabling, namely examination scheduling as well as classroom assignment. The availability of members naturally is dependent on examination scheduling and global optimization for the whole task leads to be complicated. In order to shorten total computational time for the optimization, the target problem should be divided into some sub-problems, which will be discussed in the forthcoming paper.

References

[1] A. Suzuki, K. Sawaki, and T. Hasegawa, "An OR/MS approach to managing Nanzan Gakuen (Nanzan Educational Complex): from the strategic to the daily operational level," Interfaces, vol. 36, no. 1, pp. 43-54, 2006.

[2] Z. N. Azimi, "Hybrid heuristics for examination timetabling problem," Applied Mathematics and Computation, vol.163, no.2, pp. 705-733, 2005.

[3] N. Pillay and W. Banzhaf, "A study of heuristic combinations for hyper-heuristic systems for uncapacitated examination timetabling problem," European Journal of Operational Research, vol. 197, no. 2, pp. 482-491, 2009.

[4] M. A. Ahandani, M. T. V. Baghmisheh, M. A. B. Zadeh, and S. Ghaemi, "Hybrid particle swarm optimization transplanted into a hyper-heuristic structure for solving examination timetabling problem," Swarm and Evolutionary Computation, vol. 7, pp. 21-34, 2012.

[5] S. Abdullah and M. Alzaqebah, "A hybrid self-adaptive bees algorithm for examination timetabling problems," Applied Soft Computing, vol. 13, no. 8, pp. 3608-3620, 2013.

[6] S. Petrovic, Y. Yang, and M. Dror, "Case-based selection of initialization heuristics for metaheuristic examination timetabling," Expert Systems with Application, vol. 33, no. 3, pp. 772-785, 2007.

[7] A. Dammak, A. Elloumi, and H. Kamoun, "Classroom assignment for exam timetabling," Advances in Engineering Software, vol. 37, no. 10, pp. 659-666, 2006.

[8] Y. Onouchi, T. Uchigaito, and M. Sasaki, "Timetabling university regular examination," in *Abstracts Book* (in Japanese), ORSJ Spring Study Meetings, pp. 66-67, 2013.

[9] IBM CPLEX Optimizer, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.