ISSTEC
The 3rd International Seminar on Science and Technology 2021

20515

# Mathematical Model and Simulation for Timetabling in Mathematics Department of Bandung Institute of Technology

**Riswansyah I.a*, Frietz N. A.a, Naira H. N. B.a, Justisia Q. R. A. S.a , and M. Ridwan R.N.a****

*Bandung Institute of Technology, Ganesa Street No. 10, Bandung, Indonesia, *riswansyah08@gmail.com*

**Abstract.** A process to give teaching assignments for the teachers or lecturers to the available days and times by meeting certain goals is called timetabling. The lecturer's availability, number of courses, type of courses, number of course credits, classroom availability, course start time, and course end time are the factors that affect the learning schedule that makes timetabling complicated especially in university. Therefore, by using a mathematical optimization model, timetabling process becomes easier. Although many mathematical models have been made to do timetabling, there is no general model that can use in every university. In this paper, Binary Integer Programming is used to build the learning schedule in the Mathematics Study Program of Bandung Institute of Technology (ITB). The compulsory courses do not overlap, the courses with the same year are evenly distributed, the lecturer's schedule being unable to teach, the lecturer's schedule shouldn't be teaching, and the lecturer's schedule should be teaching become the consideration in this paper. Then, based on the given data, there are three outputs, that is the schedule of the lecturers, the schedule based on the course year, and the schedule based on the day. The schedule that has been made based on the model is quite good and can be implemented in the daily life.

**Keywords:** Timetabling, Learning Schedule, Binary Integer Programming

## INTRODUCTION

According to Oxford Dictionaries, timetabling is the act of arranging for something to take place at a particular time. Almost all institutions and organizations use timetabling to manage their internal work system, one of the examples is timetabling or class scheduling in university. In general, many aspects need to be considered if we do timetables in university, like the number of students, courses, and classroom, up to the availability of lecturers, and these things depend on the university structure and policy [10]. However, there are various kinds of obstacles that will limit the completion process, such as conflicting class schedules, too busy class schedules, and many more. A conflicting and too busy class schedule will make the course will not be feasible to take.

In Mathematics ITB, the assignment of courses taught by a lecturer has been made by the head office. We have given the data and we will focus on scheduling the day and time of courses by considering some constraints that we mention above. Martin (1999) said that Binary Integer Programming is a mathematical optimization model that has variable values 1 or 0 that generally represent yes (1) and no (0) [3]. Many cases in real life can be modeled by Binary Integer Programming, like the problem of funding a project or not, opening a warehouse or not, even the timetabling problem. Timetabling or scheduling in Mathematics ITB also can be approached by Binary Integer Programming (0-1 Integer Programming). Therefore, we will build the mathematics model based on Binary Integer Programming to process the data and find the optimal solution, so that an optimal class schedule is formed for lecturers and students.

## RESULTS AND DISCUSSION

### The Timetable of Students based on Year

| | Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|---|
| 7.00 - 8.00 | MA1202_01 | MA1203_01 | 0 | MA1201_02,MA1201_02 | 0 |
| 8.00 - 9.00 | MA1202_01 | MA1203_01 | 0 | MA1201_02,MA1201_02 | 0 |
| 9.00 - 10.00 | MA1201_08,MA1201_01, MA1201_07,MA1201_11 | MA1201_03,MA1201_10, MA1201_06,MA1201_09 | MA1201_01,MA1201_04, MA1201_02,MA1201_02, MA1202_01,MA1201_09 | MA1201_03,MA1201_10, MA1201_04,MA1201_11, MA1201_05 | MA1203_01,MA1201_08, MA1201_07,MA1201_05, MA1201_06 |
| 10.00 - 11.00 | MA1201_08,MA1201_01, MA1201_07,MA1201_11 | MA1201_03,MA1201_10, MA1201_06,MA1201_09 | MA1201_01,MA1201_04, MA1201_02,MA1201_02, MA1201_09 | MA1201_03,MA1201_10, MA1201_04,MA1201_11, MA1201_05 | MA1201_08,MA1201_07, MA1201_05,MA1201_06 |
| 11.00 - 12.00 | 0 | 0 | 0 | 0 | 0 |
| 12.00 - 13.00 | 0 | 0 | 0 | 0 | 0 |
| 13.00 - 14.00 | 0 | 0 | 0 | 0 | 0 |
| 14.00 - 15.00 | 0 | 0 | 0 | 0 | 0 |
| 15.00 - 16.00 | 0 | 0 | 0 | 0 | 0 |
| 16.00 - 17.00 | 0 | 0 | 0 | 0 | 0 |
| 17.00 - 18.00 | 0 | 0 | 0 | 0 | 0 |

Figure above is the timetable for the first-year students with the point of view based on the course year. We can see that the number of every course with the same major, same year, and different code is a maximum of three for each session, this is because of the fifth hard constraint. We also can see that the courses with the same code, but different classes have the same schedule, this is because of the sixth hard constraint. If we take a look at the first-year courses, they are held under 11.00 AM, these things happen because of the tenth hard constraint. This kind of output is very suitable for the students to arrange their schedule.

### The Timetable of Lecturers

Figure beside shows some results of the simulation for build the timetable based on the lecturer's point of view. The L22 lecturer does not teach any course at 07.00 AM until 08.00 AM on Monday and 02.00 PM until 03.00 PM (14.00–15.00) on Friday respectively. These things show the function of the ninth hard constraint. Then, every lecturer teaches only one course for each session each day, this is because of the third hard constraint. We also notice that the first and second hard constraints are satisfied by the results of the simulation. This kind of output is very suitable for the lecturers to arrange their schedule.

L22 Lecturer Shcedule :

| | Monday | Tuesday | Wednesday | Thursday | Friday |
|---|---|---|---|---|---|
| 7.00 - 8.00 | 0 | 0 | MA5232_01 | 0 | 0 |
| 8.00 - 9.00 | 0 | 0 | MA5232_01 | 0 | 0 |
| 9.00 - 10.00 | MA1201_01 | 0 | MA1201_01 | 0 | 0 |
| 10.00 - 11.00 | MA1201_01 | 0 | MA1201_01 | 0 | 0 |
| 11.00 - 12.00 | 0 | 0 | 0 | 0 | 0 |
| 12.00 - 13.00 | 0 | 0 | 0 | 0 | 0 |
| 13.00 - 14.00 | 0 | 0 | 0 | 0 | 0 |
| 14.00 - 15.00 | 0 | 0 | 0 | 0 | 0 |
| 15.00 - 16.00 | 0 | MA5232_01 | 0 | 0 | 0 |
| 16.00 - 17.00 | 0 | 0 | 0 | 0 | 0 |
| 17.00 - 18.00 | 0 | 0 | 0 | 0 | 0 |

## CONCLUSION

After we have all the mathematical models of hard constraints and soft constraints, we try to test this model by using the actual data. Therefore we collect the lecturer assignment data to teach a course from the administrative officer. Then we run the simulation for solving this optimization problem by using Python Programming Language to find the optimal solution. After that, we build the learning schedule based on the optimal solution that we got. There are three types of output, that is the schedule of the lecturers, the schedule based on the course year, and the schedule based on the day.

If we look at the learning schedule that was built from the simulation, we notice that there is no violation of hard constraints, and the violation of soft constraints is minimum. We can say that this learning schedule is optimized. If we compare with the real schedule in the Mathematics Department, this schedule has some similarities. Therefore, this schedule can be applied in the real world.

## METHODOLOGY

**Objective Function**

$$Minimize \sum_{l=1}^{N_l}\sum_{k=1}^{N_k}\sum_{d=1}^{N_d}\sum_{s=1}^{N_s} c_{l,k,d,s}\cdot x_{l,k,d,s}, \forall l \in L, \forall k \in K, \forall d \in D, \forall s \in S$$

Description:

$$x_{l,k,d,s} = 1 \rightarrow \text{The lecturer 'l' teaches the course 'k' on day 'd' in session 's'}$$

$$x_{l,k,d,s} = 0 \rightarrow \text{The lecturer 'l' doesn't teach the course 'k' on day 'd' in session 's'}$$

$$c_{l,k,d,s} \text{ is the cost of objective function}$$

**Constraints**

- **Hard Constraints**

1. Each lecturer teaches according to the assigned courses and the number of credits given

$$\forall l \in L, \forall k \in \text{course assignment of lecturer } l$$

$$\sum_{d=1}^{N_d}\sum_{s=1}^{N_s} x_{l,k,d,s} = \text{the credit number of lecturer } l\text{'s course}$$

2. Each course lasts a maximum of two consecutive sessions for each day

- The constraint for a maximum of two sessions

$$\forall l \in L, \forall k \in K, \forall d \in D$$

$$\sum_{s=1}^{N_s} x_{l,k,d,s} \le 2$$

- The constraint for the consecutive course with even credit

$$\forall l \in L, \forall k \in K_{even}, \forall d \in D, \forall s \in [1,3,5,7,9]$$

$$x_{l,k,d,s+1} - x_{l,k,d,s} = 0$$

- The constraint for the consecutive course with odd credit

$$\forall l \in L, \forall k \in K_{odd}, \forall d \in D, \forall s \in [1,4,7,10]$$

$$x_{l,k,d,s+1} - x_{l,k,d,s} = 0$$

$$\forall l \in L, \forall k \in K_{odd}, \forall d \in D, \forall s \in [3,6,9]$$

$$c_{l,k,d,s\_new} = c_{l,k,d,s\_old} \cdot 3$$

3. Each lecturer can only teach one course in each session and each day

$$\forall l \in L, \forall d \in D, \forall s \in S$$

$$\sum_{k=1}^{N_k} x_{l,k,d,s} \le 1$$

4. There is only one compulsory course in each session and each day

$$\forall d \in D, \forall s \in S$$

$$\sum_{l=1}^{N_l}\sum_{k \in K_{compulsory}} x_{l,k,d,s} \le 1$$

5. Maximal course with the same year in the same session each day is 3

$$\forall d \in D, \forall s \in S$$

$$\sum_{l=1}^{N_l}\sum_{k \in K_{same\ year}} x_{l,k,d,s} \le 3$$

6. The courses that have the same code, but different classes have the same schedule

$$\forall k \in K_{same\ code}, \forall d \in D, \forall s \in S$$

$$\sum_{l=1}^{N_l} x_{l,k,d,s} - \sum_{l=1}^{N_l} x_{l,k+1,d,s} = 0$$

7. The courses that have the same code, but different classes have the same schedule

$$\forall k \in K_{same\ code}, \forall d \in D, \forall s \in S$$

$$\sum_{l=1}^{N_l} x_{l,k,d,s} - \sum_{l=1}^{N_l} x_{l,k+1,d,s} = 0$$

8. The maximal course in each session is $M_{ks}$ for everyday

$$\forall d \in D, \forall s \in S$$

$$\sum_{l=1}^{N_l}\sum_{k=1}^{N_k} x_{l,k,d,s} \le M_{ks}$$

9. The course that teaches by 2 lecturers or more have the same schedule

$$\forall l \in L, \forall k \in K_{teaches\ by\ more\ than\ 1\ lecuter}, \forall d \in D, \forall s \in S_{odd}$$

$$x_{l,k,d,s+1} - x_{l,k,d,s} = 0$$

10. The lecturer can enter their unavailability schedule

$$\forall l \in L_{unavailable}, \forall k \in K, \forall d \in D_{unavailable}, \forall s \in S_{unavailable}$$

$$x_{l,k,d,s} = 0$$

11. There are several courses that have been scheduled at certain times and days

$$\forall l \in L_{scheduled\ teach\ k}, \forall k \in K_{scheduled}, \forall d \in D_{scheduled}, \forall s \in S_{scheduled}$$

$$x_{l,k,d,s} = 1$$

- **Soft Constraints**

1. The class is tried not to be held at some predetermined time like break time

$$\forall l \in L, \forall k \in K, \forall d \in D_{predetermined}, \forall s \in S_{predetermined}$$

$$c_{l,k,d,s\_new} = c_{l,k,d,s\_old} \cdot 100$$

2. The lecturer can suggest their schedule where they should teach

$$\forall l \in L_{suggest}, \forall k \in K, \forall d \in D_{should\ teach}, \forall s \in S_{should\ teach}$$

$$c_{l,k,d,s\_new} = c_{l,k,d,s\_old} \cdot 0.1$$

3. The lecturer can suggest their schedule where they should not teach

$$\forall l \in L_{suggest}, \forall k \in K, \forall d \in D_{shouldn't\ teach}, \forall s \in S_{shouldn't\ teach}$$

$$c_{l,k,d,s\_new} = c_{l,k,d,s\_old} \cdot 10$$

## REFERENCES

1. Bazaraa, Mokhtar S., John J. Jarvis, and Hanif D. Sherali. (2010). Linear Programming and Network Flows (4th Ed). New Jersey: John Wiley & Sons, Inc.
2. Luenberger, David G. and Yinyu Ye. (2008). Linear and Nonlinear Programming (3rd Ed). New York: Springer Science + Business Media, LLC.
3. Martin, Richard Kipp. (1999). Large Scale Linear and Integer Optimization: A Unified Approach. New York: Springer Science + Business Media, LLC.
4. Kellerer, Hans, Ulrich Pferschy, and David Pisinger. (2004). Knapsack Problems. New York: Springer-Verlag Berlin Heidelberg.
5. Nakasuwan, Janewit, Piyawadee Srithip, and Somrote Komolavanij. (1999). Class Scheduling Optimization. Thammasat International Journal of Science and Technology, 4(2), 88-98.
6. Perzina, Radom´ır. (2007). Solving the University Timetabling Problem with Optimized Enrollment of Students by a Self-adaptive Genetic Algorithm. LNCS, 3867, 248-263.
7. Murray, Keith, Tom´aˇs M¨uller, and Hana Rudov´a. (2007). Modeling and Solution of a Complex University Course Timtabling Problem. LNCS, 3867, 189-209.
8. Haan, Peter de, Ronald Landman, Gerhard Post, and Henri Ruizenaar. (2007). A Case Study for Timetabling in a Dutch Secondary School. LNCS, 3867, 267-279.
9. Carter, Michael W., and Gilbert Laporte. (1998). Recent Developments in Practical Course Timetabling. LNCS, 1408, 3-19.
10. Daskalaki, S. and T. Birbas. (2003). Efficient Solutions for a University Timetabling Problem through Integer Programming. European Journal of Operational Research, 160(2005), 106-120.
11. Benli, Omer S. and A. Reha Botsali. (2004). An Optimization-Based Decision Support System for a University Timetabling Problem: An Integrated Constraint and Binary Integer Programming Approach. Computer and Industrial Engineering, 1-29.