OR: Course Slots Scheduling Model

Dung Ha

Feb 7th 2024

1 Motivation

A course (time-slot) schedule is the arrangement of time slots during the week where instructors can schedule their course. A linear programming model is a mathematical formulation of a problem into linear relationships. The solution of those relationships is the solution to the model and provides great analytical value to solving the original problem. This linear programming model is being devised to help create the best possible course schedule for the Dickinson College.

This model aims to solve the following problem: Dickinson College's current course schedule only has two 1.5hr common hours (time where there is no class) during the week. Is there a way to have more common hours and 75 minute classes while keeping similar requirements?

This model aims to automatically create schedules that has similar requirements as the current Dickinson College's course scheduling, while optimizing the schedules to have as many 75 minute standard classes, and 1.5hr common hour as possible.

2 Method

2.1 Linear Programming

The standard form of linear programming is:

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} w_j x_j \\ \text{subject to} \quad & \sum_{j:e_i \in S_j} a_{ij} x_j \leq b_i, i = 1, ..., n \\ & \qquad\qquad\qquad j = 1, ..., m \end{aligned}$$

where there are i constraints and j decision variables.

2.2 Binary Integer Programming

We can put a constraint on the decision variables to have only value 0 or 1 and create a Binary Integer Programming model:

$$\begin{aligned}
\text{minimize} \quad & \sum_{j=1}^{m} w_j x_j \\
\text{subject to} \quad & \sum_{j:e_i \in S_j} a_{ij} x_j \leq b_i, \qquad i = 1, ..., n \\
& x_j \in \{0, 1\}, \quad j = 1, ..., m
\end{aligned}$$

where there are i constraints and j decision variables.

This model is very useful in creating a situation where some thing may happen or not. For example if a course can start at a certain time on Monday.

2.3 Gurobi Optimizer

We will use Gurobi Optimizer, a powerful software used to solve mathematical optimization problem to solve the model. We will also use Gurobipy and Jypiter Notebook to input our course scheduling model into the Gurobi Optimizer.

2.4 Tkinter

We will also use Tkinter, a python library for GUI to transfer the solutions to our model into graphical representations.

3 Modeling

3.1 Requirements

There are several general groups of requirement of the course schedule for this model:

- The schedule includes standard courses that lasts for 50 minutes or 75 minutes, three hour labs, language classes, three hours seminar, 1h common hour, and 1.5 hours common hour.
- The number of courses of each type is limited
- All of theses classes are called courses in this model. Each course starts at the same time during a day and meet a specific number of times during the week.
- Courses may or may not start at certain times
- A course may meet in consecutive days or not. But it cannot meet more than once a day

- There are courses that can overlap with each other and others that must not
- Classes starts on half-hour in this model (8:00 AM, 8:30 AM, 9:00 AM, ..., 15:30 PM).
- Everyday, courses can start only as early as 8:30 AM, except for at most two days where it can start at 8:00 AM The specific requirements are to be implemented as constraint in the model (see model below). The objective is maximizing the number of common hours and 75 minute classes.
- The detailed requirements can be seen from the model

3.2 Requirements implementation

- Each variable $x_{d,s,c}$ represents whether a meeting of a specific course s will start on day d of the week at period s.
- Each variable $w_{s,c}$ represents whether a meeting of course c is scheduled on period s. It is used to make sure all meeting of the same course time slot starts at the same time (period) in any day.
- Each variable e_d represents whether meetings can start at 8:00 AM (the earliest possible start time) on day d.
- Each variable a_c represents whether course c is scheduled. We use this variable for consistency between the availability of the course and the appropriate scheduling of the meeting of that course.
- To efficiently address desired variables, we need to combine the type of the variable with index sets. Sets that starts with S help cover the index of periods (start time). Each group of start time indexes help us limit start times as dependent on the context. Notice that classes are only allowed to start on half-hour in this model which are 8:00 AM, 8:30 AM, 9:00 AM ... which can be indexed with 1,2,3, ...
- Similarly, we have groups that starts with C which helps addressing variable that is related to a set of courses. For example, x_d, s, c where $c \in C_3$ tells us that we are only consider meetings for 3-hour lab sessions.
- Lastly, sets that starts with D, index the specific days in consideration
- So for example, x_d, s, c where $d \in D_1, s \in S_1, c \in C_1$ is an expression that considers the availability of each meeting that can only starts during the morning time, on weekdays, and belongs to a 50-minute standard course time-slot.

4 Model

4.1 Sets

- Start time (periods): $S = \{0, 1, ..., 15\}$ (8:00 AM, 8:30 AM, 9:00 AM, 9:30 AM, ..., 15:30 PM)
- Morning periods $S_1 = \{0, 1, 2, ..., 7\}$
- Afternoon periods $S_2 = \{8, 9, ..., 15\}$
- Lab periods $S_3 = \{8, 9, 10, 11\}$
- Seminar periods $S_4 = \{8, 9, 10, 11\}$
- 1.5hr Common hour periods $S_5 = \{0, 1, 2, ..., 13, 14\}$
- Faculty meeting periods $S_6 = \{0, 1, 2, ..., 13\}$
- Middle periods: $S_7 = \{5, 6, 7, ..., 14\}$
- Courses: $C = \{0, 1, 2, ..., 24\}$ (All courses)
- Course1: $C_1 = \{0, 1, ..., 6\}$ (1h courses)
- Course2: $C_2 = \{5, 6, ..., 15\}$ (1.5h courses)
- Course3: $C_3 = \{16\}$ (3h lab)
- Course4: $C_4 = \{17\}$ (1.5h common hour)
- Course5: $C_5 = \{18, 19, 20\}$ (morning language courses)
- Course6: $C_6 = \{21\}$ (afternoon language course)
- Course7: $C_7 = \{22\}$ (1h common hour)
- Course8: $C_8 = \{23\}$ (faculty meeting)
- Course9: $C_9 = \{24\}$ (seminar)
- Day: $D = \{0, 1, ..., 6\}$ (Monday - Saturday)
- Day: $D_1 = \{0, 1, ..., 5\}$ (Monday - Friday)

4.2 Variables

- $x_{d,s,c}$: a course slot c is scheduled to start at period s, day d
- $w_{s,c}$: a course slot c is scheduled to start at period s
- e_d: Courses are allowed to start at 8:00 AM on day d
- a_c: Course c is scheduled

4.3 Objective function

maximize

$$z = 20 \sum_{d \in D} \sum_{s \in S} \sum_{c_4 \in C_4} (x_{d,s,c_4}) + 60 \sum_{d \in D} \sum_{s \in S_7} \sum_{c_4 \in C_4} (x_{d,s,c_4}) + 3 \sum_{d \in D} \sum_{s \in S} \sum_{c_7 \in C_7} (x_{d,s,c_7}) + 8 \sum_{d \in D} \sum_{s \in S_7} \sum_{c_7 \in C_7} (x_{d,s,c_7})$$

$$+30 \sum_{d \in D} \sum_{s \in S} \sum_{c_2 \in C_2} (x_{d,s,c_2})$$

(add as many common hrs as possible, prioritize 1.5hr common hour, add as many 75 minutes class as possible)

4.4 Constraints

(I) Number of meetings in specific courses

1. $$\sum_{d \in D} \sum_{s \in S} x_{d,s,c} = 3a_c, c \in C_1$$

   (every 1hr standard course slot meets 3 times a week)

2. $$\sum_{d \in D} \sum_{s \in S} x_{d,s,c} = 2a_c, c \in C_2$$

   (every 1.5hr standard course slot meets twice a week)

3. $$\sum_{d \in D} \sum_{s \in S} x_{d,s,c} = 5, c \in C_3, C_5, C_6$$

   (every 5 meetings course slot meets five times a week)

4. $$\sum_{d \in D_1} \sum_{s \in S_4} \sum_{c \in C_9} x_{d,s,c} = 1$$

   (one seminar meeting per week)

(II) Invalid course start times

5. $$\sum_{s \in S} \sum_{c \in C} x_{6,s,c} = 0$$

   (No classes on Saturday)

6. $$\sum_{d \in D_1} \sum_{s \in S/S_5} \sum_{c \in C_8} x_{d,p,c} = 0$$

   (A faculty meeting can only be scheduled in Faculty Meeting hours)

7.

$$\sum_{d\in D_1}\sum_{s\in S/S_4}\sum_{c\in C_9} x_{d,p,c} = 0$$

(A seminar meeting can only be scheduled in Seminar hours)

8.

$$\sum_{d\in D}\sum_{c\in C_2,C_4} x_{d,15,c} = 0$$

(no period 15 is valid for 1.5h courses)

9.

$$\sum_{c\in C_5,}\sum_{d\in D}\sum_{s\in S_2} x_{d,s,c} = 0$$

(No afternoon meetings for morning language courses)

10.

$$\sum_{c\in C_6,C_3,}\sum_{d\in D}\sum_{s\in S_1} x_{d,s,c} = 0$$

(No morning meetings for afternoon language courses, and labs)

11.

$$\sum_{c\in C_3,}\sum_{d\in D_1}\sum_{s\in S/S_3} x_{d,s,c} = 0$$

(No morning meetings or late afternoon meetings for labs)

(V) Fixed meeting times

12.

$$\sum_{d\in D}(x_{d,s,c}) - 3w_{s,c} = 0, c \in C_1, s \in S$$

(each 1hr course slot will be scheduled at a fixed period (start time))

13.

$$\sum_{d\in D}(x_{d,s,c}) - 2w_{s,c} = 0, c \in C_2, s \in S$$

(each 1.5hr course slot will be scheduled at a fixed period (start time))

14.

$$\sum_{d\in D}(x_{d,s,c}) - 5w_{s,c} = 0, c \in C_5, C_6, s \in S$$

(each language course slot will be scheduled at a fixed period (start time))

(IV) Day spacing for each course

15.

$$\sum_{s\in S}(x_{d,s,c}+x_{(d+1)mod|D_1|,s,c}+x_{(d+2)mod|D_1|,s,c})\leq 2, d\in D_1, c\in C_1$$

(no more than two meetings will be scheduled in three consecutive weekdays for every 1hr course slot)

16.

$$\sum_{s\in S}(x_{d,s,c}+x_{(d+1)mod|D_1|,s,c})\leq 1, d\in D_1, c\in C_2$$

(no meetings will be scheduled in consecutive weekdays for each 1.5hr course slot)

17.

$$\sum_{s\in S}x_{d,s,c}\leq 1, c\in C_3, C_5, C_6, d\in D_1$$

(at most 1 meeting per day for every 5 meetings course slot)

(VI) Collision (overlapping) of courses' meetings

18. (constraint 4.5 in the model)

$$\sum_{c\in C_1}(x_{d,s,c}+x_{d,s+1,c})\leq 1, d\in D_1, s\in S/\{15\}$$

(1h standard course slots do not collide)

19. (constraint 5.5 in the model)

$$\sum_{c\in C_2,C_4}(x_{d,s,c}+x_{d,s+1,c}+x_{d,s+2,c})\leq 1, d\in D_1, s\in S/\{14,15\}$$

(1.5hr standard course and 1.5 common hour slots do not collide)

20.

$$\sum_{c\in C_7}(x_{d,s,c}+x_{d,s+1,c})\leq 1, d\in D_1, s\in S/\{15\}$$

(1hr common hour slots do not collide)

21.

$$\sum_{c\in C_5,C_6}x_{d,s,c}+x_{d,s+1,c}\leq 1, s\in S_1$$

(language courses cannot collide)

22.

$$\sum_{c\in C_5,C_3} x_{d,s,c} + x_{d,s+1,c} \leq 1, s \in S_1$$

(morning language courses cannot collide with labs. Works at the moment because labs are only scheduled for the afternoon. In other words with the current model a lab always starts after morning language classes.)

23.

$$(x_{d,s-1,c_2}+x_{d,s-2,c_2}+x_{d,s,c_2}+x_{d,s+1,c_2})^{(*)}+Mx_{d,s,c_1} = M, c_1 \in C_1, c_2 \in C_2, s \in S, d \in D_1$$

(*: $s-1, s-2, s+1$ only exist if they belong to S)
(no overlapping allowed for 1hr standard course slots and 1.5hr standard course slots)

(III) Arrangement for special courses

24.

$$\sum_{c\in C_1,C_5,C_6,C_7} x_{d,s-1,c}+\sum_{c\in C_2}(x_{d,s-1,c}+x_{d,s-2,c})+\sum_{c\in C_8}(x_{d,s-1,c}+x_{d,s-2,c}+x_{d,s-3,c})$$
$$+\sum_{c\in C_3,C_9}(x_{d,s-1,c}+x_{d,s-2,c}+x_{d,s-3,c}+x_{d,s-4,c}+x_{d,s-5,c})$$
$$+\sum_{c\in C/C_4}(x_{d,s,c}+x_{d,s+1,c}+x_{d,s+2,c})+Mx_{d,s,c_4} = M, d \in D, c_4 \in C_4, s \in S_5$$

(*: $s-1, s-2, s-3, s-4, s-5$ only exist if they belong to S)
(course slots arrangement for 1.5hr common hours)

25.

$$\sum_{c\in C_1,C_5,C_6} x_{d,s-1,c}+\sum_{c\in C_2,C_4}(x_{d,s-1,c}+x_{d,s-2,c})+\sum_{c\in C_8}(x_{d,s-1,c}+x_{d,s-2,c}+x_{d,s-3,c})$$
$$+\sum_{c\in C_3,C_9}(x_{d,s-1,c}+x_{d,s-2,c}+x_{d,s-3,c}+x_{d,s-4,c}+x_{d,s-5,c})$$
$$+\sum_{c\in C/C_7}(x_{d,s,c}+x_{d,s+1,c})+Mx_{d,s,c_7} = M, d \in D, c_7 \in C_7, s \in S$$

(*: $s-1, s-2, s-3, s-4, s-5$ only exist if they belong to S)
(course slots arrangement for 1hr common hours)

26.

$$\sum_{c\in C_1,C_5,C_6,C_7} x_{d,s-1,c}+\sum_{c\in C_2,C_4}(x_{d,s-1,c}+x_{d,s-2,c})$$

$$+\sum_{c\in C_3,C_9}(x_{d,s-1,c}+x_{d,s-2,c}+x_{d,s-3,c}+x_{d,s-4,c}+x_{d,s-5,c})$$

$$+\sum_{c\in C/C_8}(x_{d,s,c}+x_{d,s+1,c}+x_{d,s+2,c}+x_{d,s+3,c})+Mx_{d,s,c_8}=M, d\in D, c_8\in C_8, s\in S_6$$

(*: $s-1, s-2, s-3, s-4, s-5$ only exist if they belong to S)
(course slots arrangement for 2.0hr faculty meeting)

27.

$$\sum_{c\in C_1,C_5,C_6,C_7} x_{d,s-1,c}+\sum_{c\in C_2,C_4}(x_{d,s-1,c}+x_{d,s-2,c})+\sum_{c\in C_8}(x_{d,s-1,c}+x_{d,s-2,c}+x_{d,s-3,c})$$

$$+\sum_{c\in C/C_9}(x_{d,s,c}+x_{d,s+1,c}+x_{d,s+2,c}+x_{d,s+3,c}+x_{d,s+4,c}+x_{d,s+5,c})+Mx_{d,s,c_9}=M, d\in D, c_9\in C_9, s\in S_4$$

(class arrangement for seminar)
(*: $s-1, s-2, s-3, s-4, s-5$ only exist if they belong to S)

(VII) Standard courses size and 1.5hr common hours size

28.

$$\sum_{c\in C_1,C_2} a_c \geq 11$$

(At least 11 standard courses will be scheduled)

29.

$$\sum_{c\in C_1} a_c \geq 4$$

(At least 4 1hr standard courses will be scheduled)

30.

$$\sum_{c\in C_2} a_c \geq 5$$

(At least 5 1.5hr standard courses will be scheduled)

31.

$$\sum_{d\in D_1}\sum_{s\in S_5}\sum_{c\in C_4} x_{d,s,c} \geq 2$$

(At least 2 1.5hr common hours will be scheduled)

(VIII) Early starts

32.

$$\sum_{d\in D_1} e_d \leq 2$$

(At most 2 days are allowed where courses can start early)

33.
$$\sum_{c \in C} x_{d,0,c} + Me_d \leq M, d \in D_1$$

(No courses are allowed to start early when the day is not eligible for an early start)

5 Result

Not only has the model been able to find more common hours than that of our original course time-slot schedule (only two), it has helped us see that we can increase the length of some common hours by 30 minutes.

6 Discussion

This model not only save time and effort compared to manually create a schedule that adhere to the same set of requirements/constraint, but also help us approach the scheduling problem in incremental steps. We can revert to less requirements or add requirements on the go.

7 Sample Schedules

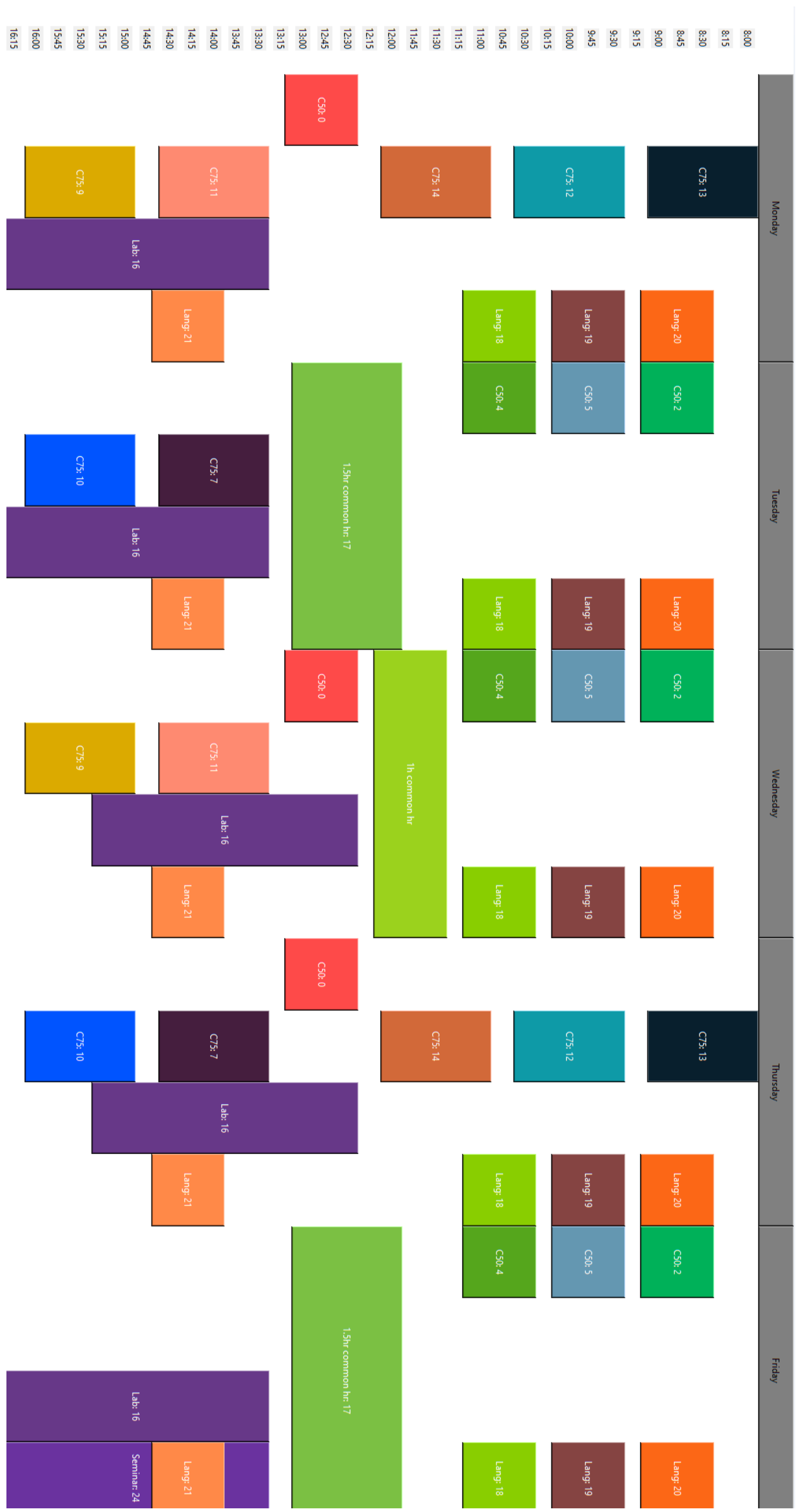

Figure 1: Sample Schedule 1

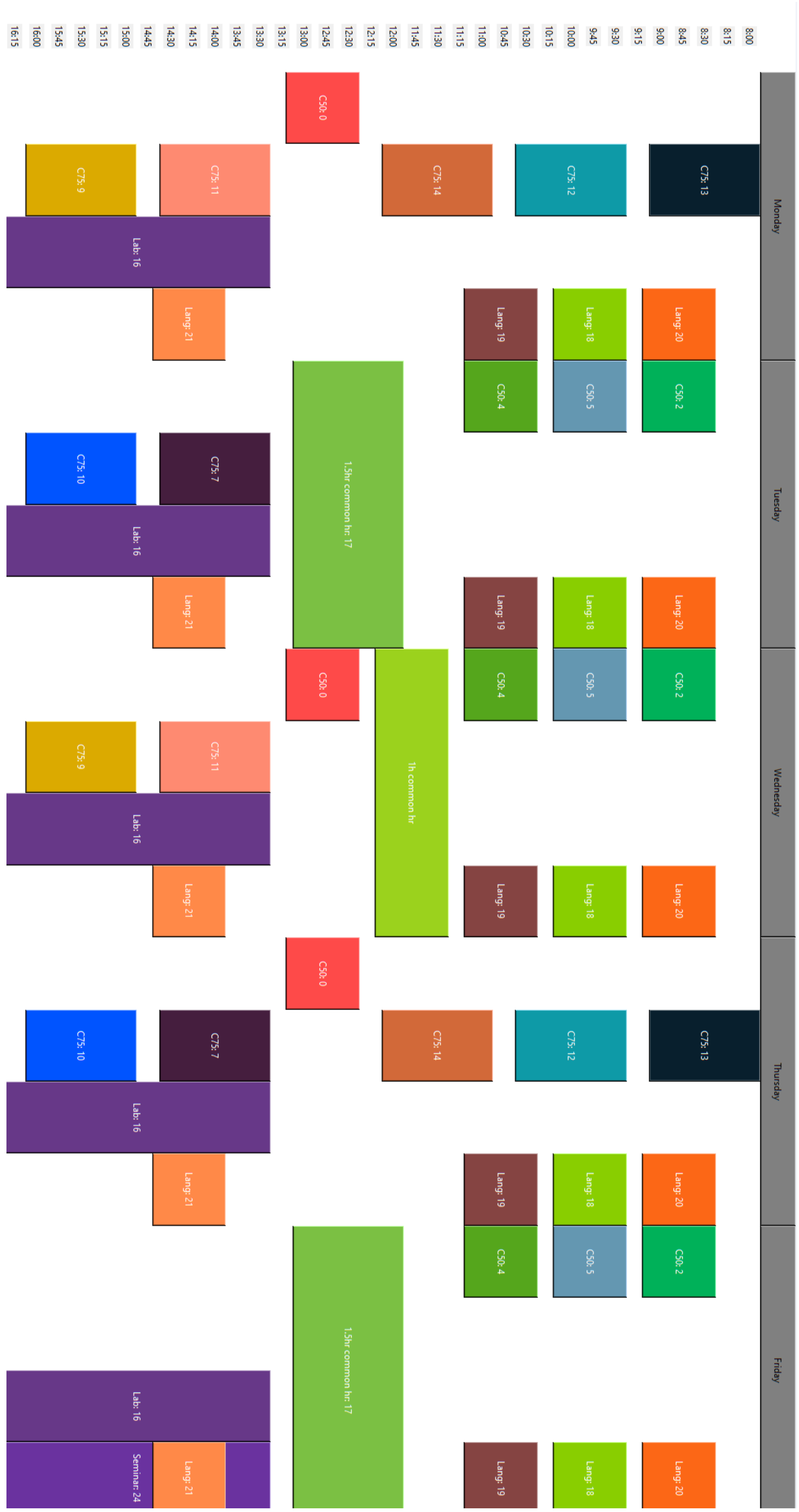

Figure 2: Sample Schedule 2