OR: Course Scheduling Research Document

Dung Ha

May 15th 2024

Contents

1 Abstract

For almost two decades, Dickinson College has utilized the same course time slots, which are the allowable days and times that courses can be offered. While the current schedule has served the college well, there is a desire to revisit the schedule to increase the number of possible 75-minute time slots and increase the number of common hours. To achieve this, a sophisticated mathematical model based on linear integer programming was developed. This model, utilizing constraints to enforce existing rules, aims to generate alternative schedules that meet the college's requirements while maximizing the availability of 75-minute classes and common hours. This approach not only saves considerable time and effort but also explores a wider range of possibilities than manual scheduling could achieve.

2 Introduction

2.1 Goal

Use optimization modeling to generate course schedules that maximize the number of 75-minute classes and common hours.Schedules must satisfy rules (constraints) such as:

1. The schedule is broken up into half-hour increments. All sessions start and end on the half-hour. Each day can start no earlier than 8:30 and end no later than 4:30.
2. There are 2 standard types of academic courses: 50-minute classes and 75-minute classes
   (a) The 50-minute classes meet 3 days a week
   (b) The 75-minute classes meet 2 days a week
3. There must be at least 11 total standard classes in a week. (Currently five 50-minute classes and six 75-minute classes).
4. There are three daily morning language courses that end before noon, and one daily afternoon language course that starts after noon. These may overlap with the standard classes.
5. There is a 3-hour lab every day.

2.2 Binary Integer Programming

The standard form of linear programming is:

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

where there are i constraints and j decision variables.

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

We use Gurobi Optimizer, a powerful software used to solve mathematical optimization problems to solve the model. Gurobipy and Jupyter Notebook are used input our course scheduling model into the Gurobi Optimizer. We also use Tkinter, a Python library for GUI, to transform the solutions of our model's solutions into graphical representations.

3 Methods

3.1 Sample Requirements and Objective

There are several general groups of requirements of the course schedule for this model, including, but not limited to:

- 50-minute standard course slots meets three times a week.
- Any two course slots of the same type may not overlap.

The objective is to maximize the number of common hours and 75-minute classes.

3.2 Sample requirements implementation

Decision variables:

- 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) on any day.

- 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.

Index sets: (This approach of index sets is adopted from Havas et al. (2013))

- To efficiently address desired variables, we need to combine the type of the variable with index sets. Sets that start with S help cover the index of periods (start time). Each group of start time indexes helps us limit start times as dependent on the context. Notice that classes are only allowed to start on the half-hour in this model which are 8:00 AM, 8:30 AM, 9:00 AM ... which can be indexed with 0, 1, 2, 3, ...
- Similarly, we have groups that start with C which helps address variables that are related to a set of courses. For example, $x_{d,s,c}$ where $c \in C_1$ tells us that we are only considering meetings for 50-minute course slots 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 start during the morning time, on weekdays, and belongs to a 50-minute standard course time-slot.

Sample conversion of requirements to model constraints:

- every 50-minute standard course slot meets 3 times a week:

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

- 50-minute standard course slots do not overlap:

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

3.3 Code implementation

See code in "Dickinson_Scheduling_Model.ipynb"

4 Model

4.1 Notes

- A meeting is a session that occurs at a specific day at a specific period.
- A course (course slot/course time-slot) is a set of related meeting which offers a time slot for a programmed activity (an academic course, common hours, faculty meetings, labs, seminars)

- A language meeting lasts 50 minutes. Three language course slots must be scheduled in the morning and one must be scheduled in the afternoon. The meetings of these courses can overlap with any courses except for common hours.
- A faculty meeting lasts 1.75 hour. Both faculty meetings and common hour are scheduled when no other meetings is occurring
- A seminar or a lab lasts 2.75 hours. A seminar can only overlap with a language meeting or a lab. A lab can overlap with all courses except for faculty meetings and common hours

4.2 Sets

- Start time (periods): $S = \{0, 1, ..., 15\}$ (8:00, 8:30, 9:00, 9:30, ..., 15:30)
- 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 $S_4 = \{8, 9, 10, 11\}$
- 75-minute common hour periods $S_5 = \{0, 1, 2, ..., 13, 14\}$
- Faculty meeting periods $S_6 = \{0, 1, 2, ..., 13\}$
- Middle periods: $S_7 = \{5, 6, 7, ..., 13\}$
- Courses: $C = \{0, 1, 2, ..., 23\}$ (All courses)
- Course1: $C_1 = \{0, 1, ..., 5\}$ (50-minute courses)
- Course2: $C_2 = \{6, 7, ..., 14\}$ (75-minute courses)
- Course3: $C_3 = \{15\}$ (2.75h lab)
- Course4: $C_4 = \{16\}$ (75-minute common hour)
- Course5: $C_5 = \{17, 18, 19\}$ (morning language courses)
- Course6: $C_6 = \{20\}$ (afternoon language course)
- Course7: $C_7 = \{21\}$ (50-minute common hour)
- Course8: $C_8 = \{22\}$ (faculty meeting)
- Course9: $C_9 = \{23\}$ (seminar)
- Course10: $C_{10} = \{0, 1, 2\}$ (non-overlapping 50-minute course slots)
- Course11: $C_{11} = \{3, 4, 5\}$ (overlapping 50-minute course slots)

- Course12: $C_{12} = \{6, 7, 8\}$ (non-overlapping 50-minute course slots)
- Course13: $C_{13} = \{9, 10, ..., 14\}$ (overlapping 50-minute course slots)
- Day: $D = \{0, 1, ..., 5\}$ (Monday - Saturday)
- Day: $D_1 = \{0, 1, ..., 4\}$ (Monday - Friday)

4.3 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 on day d
- a_c: Course c is scheduled

4.4 Objective function

maximize

$$z = 20 \sum_{d\in D}\sum_{s\in S}\sum_{c_4\in C_4}(x_{d,s,c_4}) + 60\sum_{d\in D}\sum_{s\in S_7}\sum_{c_4\in C_4}(x_{d,s,c_4}) + 3\sum_{d\in D}\sum_{s\in S}\sum_{c_7\in C_7}(x_{d,s,c_7}) + 15\sum_{d\in D}\sum_{s\in S_7}\sum_{c_7\in C_7}(x_{d,s,c_7})$$

$$+130\sum_{d\in D}\sum_{s\in S}\sum_{c_2\in C_2}(x_{d,s,c_2}) + 10\sum_{d\in D}\sum_{s\in S}\sum_{c_1\in C_1}(x_{d,s,c_1}) + 20\sum_{d\in D_1}\sum_{s\in\{7,8,9\}}\sum_{c_3\in C_3}(x_{d,s,c_3})$$

$$-100\sum_{d\in D_1}\sum_{s\in\{0,1\}}\sum_{c\in C}(x_{d,s,c})$$

(Add as many 75-minute course slots, common hours, and 50-minute course slots as possible, and prioritize adding the courses in that order. Prioritize early lab start times. Heavily penalize start times before 9 AM).

4.5 Constraints groups

(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 50-minute 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 75-minute 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.
$$\sum_{d\in D_1}\sum_{s\in S/S_6}\sum_{c\in C_8} x_{d,p,c} = 0$$

(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.
$$\sum_{d\in D_1}\sum_{c\in C_2, C_4} x_{d,15,c} = 0$$

(no period 15 is valid for 75-minute courses)

9.
$$\sum_{c\in C_5,}\sum_{d\in D_1}\sum_{s\in S_2} x_{d,s,c} = 0$$

(No afternoon meetings for morning language courses)

10.
$$\sum_{c\in C_6, C_3,}\sum_{d\in D_1}\sum_{s\in S_1} x_{d,s,c} = 0$$

(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)

(III) Fixed meeting times

12.

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

(each 50-minute 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 75-minute 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) Standard courses' sets size and number of 75-minute common hours

15.

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

(At least 11 standard courses will be scheduled)

16.

$$\sum_{c \in C_1} a_c \geq 3$$

(At least 3 50-minute standard courses will be scheduled)

17.

$$\sum_{c \in C_10} a_c \geq 2$$

(At least 2 50-minute non-overlapping (with other standard courses) standard courses will be scheduled)

18.

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

(At least 5 75-minute standard courses will be scheduled)

19.

$$\sum_{c \in C_12} a_c \geq 2$$

(At least 2 75-minute non-overlapping (with other standard courses) standard courses will be scheduled)

20.

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

(At least 2 75-minute common hours will be scheduled)

(V) Early starts

21.

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

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

22.

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

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

(VI) Day spacing for each course

23.

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

(no more than two meetings will be scheduled in three consecutive weekdays for every 50-minute course slot)

24.

$$\sum_{s \in S_5} (x_{d,s,c} + x_{(d+1)mod|D_1|,s,c}) \leq 1, d \in D_1, c \in C_2$$

(no meetings will be scheduled in consecutive weekdays for each 75-minute course slot)

25.

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

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

(VII) Collision (overlapping) of courses' meetings

26. (constraint 4.5 in the model)

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

(50-minute standard course slots do not collide)

27. (constraint 5.5 in the model)

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

(75-minute standard course and 75-minute common hour slots do not collide)

28.

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

(50-minute common hour slots do not collide)

29.

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

(language courses cannot collide)

30.

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

(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.)

31.

$$(x_{d,s-1,c_2} + x_{d,s-2,c_2} + x_{d,s,c_2} + x_{d,s+1,c_2})^{(*)} + Mx_{d,s,c_1} = M, c_1 \in C_{10}, c_2 \in C_2, s \in S, d \in D_1$$

(*: $s-1, s-2, s+1$ only exist if they belong to S)
(no overlapping allowed for 50-minute non-overlapping standard course slots and 75-minute standard course slots)

32.

$$(x_{d,s-1,c_2} + x_{d,s-2,c_2} + x_{d,s,c_2} + x_{d,s+1,c_2})^{(*)} + Mx_{d,s,c_1} = M, c_1 \in C_1, c_2 \in C_{12}, s \in S, d \in D_1$$

(*: $s-1, s-2, s+1$ only exist if they belong to S)
(no overlapping allowed for 75-minute non-overlapping standard course slots and 50-minute standard course slots)

33.

$$(x_{d,s-2,c_2} + x_{d,s+1,c_2})^{(*)} + Mx_{d,s,c_1} = M, c_1 \in C_{11}, c_2 \in C_{13}, s \in S, d \in D_1$$

(*: $s-1, s-2, s+1$ only exist if they belong to S)
(if a standard 75-minute course slot meeting overlaps with a standard 50-minute course slot meeting, either they start at the same time or end at the same time)

(VIII) Arrangement for special courses

34.

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

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

$$+\sum_{c\in C/C_4}(x_{d,s,c}+x_{d,s+1,c}+x_{d,s+2,c})+Mx_{d,s,c_4}\le M, d\in D_1, c_4\in C_4, s\in S_5$$

(*: $s-1, s-2, s-3, s-4, s-5$ only exist if they belong to S)
(course slot arrangement for 75-minute common hours)

35.

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

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

$$+\sum_{c\in C/C_7}(x_{d,s,c}+x_{d,s+1,c})+Mx_{d,s,c_7}\le M, d\in D_1, c_7\in C_7, s\in S$$

(*: $s-1, s-2, s-3, s-4, s-5$ only exist if they belong to S)
(course slot arrangement for 50-minute common hours)

36.

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

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

$$+\sum_{c\in C/C_8}(x_{d,s,c}+x_{d,s+1,c}+x_{d,s+2,c}+x_{d,s+3,c})+Mx_{d,s,c_8}\le M, d\in D_1, c_8\in C_8, s\in S_6$$

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

37.

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

$$+\sum_{c\in C/C_9}(x_{d,s,c}+x_{d,s+1,c}+x_{d,s+2,c}+x_{d,s+3,c}+x_{d,s+4,c}+x_{d,s+5,c})+Mx_{d,s,c_9}\le M, d\in D_1, c_9\in C_9, s\in S_4$$

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

5 Result

The schedule above satisfies all the requirements and has two more 75-minute course slots and one more 75-minute common hour compared to the current Dickinson College course schedule.

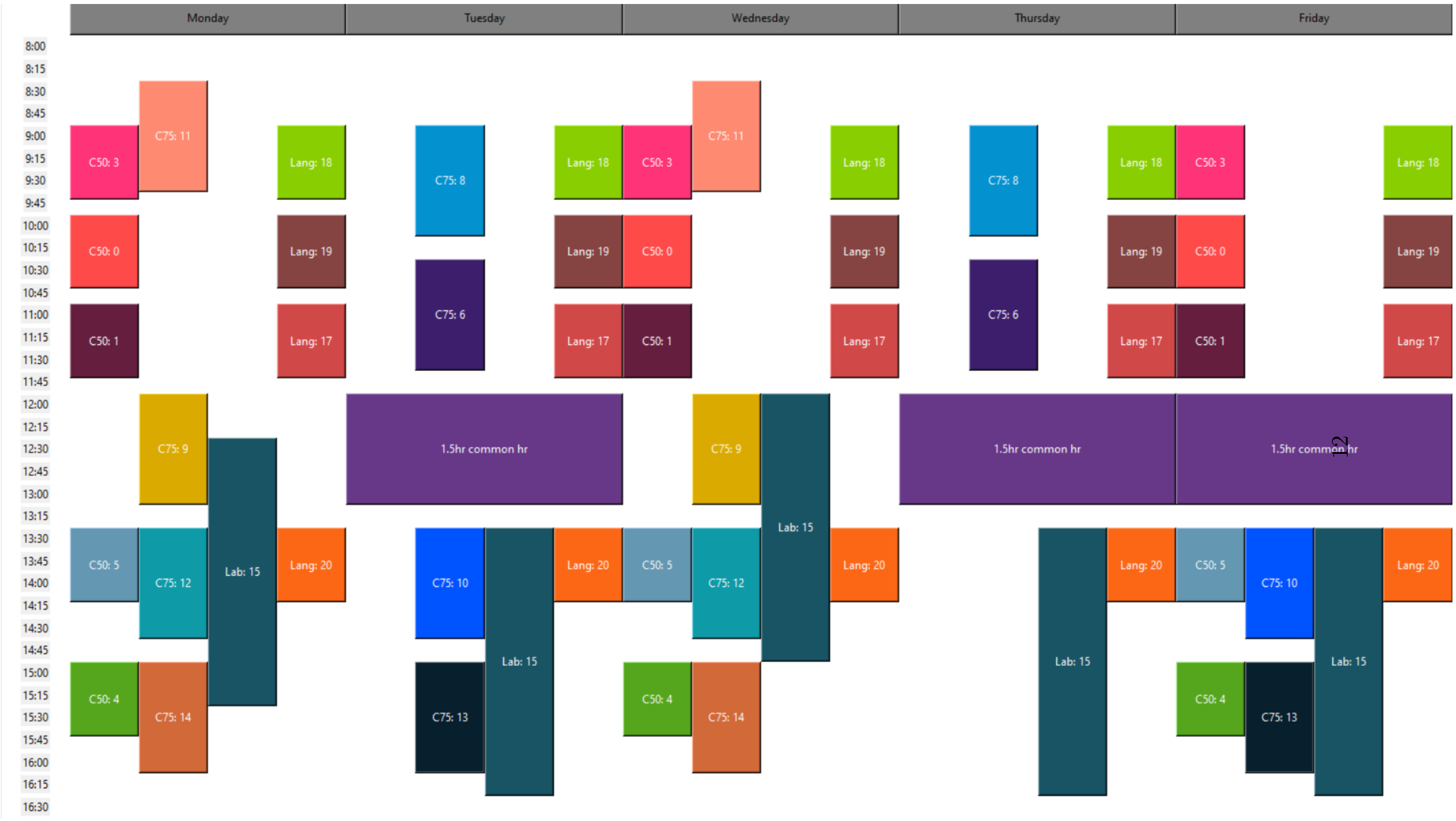
6 Graphical Representation of Example Generated Schedule

Guide for reading the schedule:

- For every day:
  - The first sub-column has the (standard) 50-minute course meetings
  - The second sub-column has the (standard) 75-minute course meetings
  - The third sub-column has the labs
  - The fourth sub-column has the language course meetings and potentially a seminar meeting

Generated statistics:

- There are 5 50-minutes standard courses
- There are 8 75-minutes standard courses
- There are 3 1.5hr common hrs
- There are 0 1h common hrs


Monday
Tuesday
Wednesday
Thursday
Friday
8:00
8:15
8:30
8:45
9:00
9:15
9:30
9:45
10:00
10:15
10:30
10:45
11:00
11:15
11:30
11:45
12:00
12:15
12:30
12:45
13:00
13:15
13:30
13:45
14:00
14:15
14:30
14:45
15:00
15:15
15:30
15:45
16:00
16:15
16:30
C50: 3
C75: 11
Lang: 18
C50: 0
Lang: 19
C50: 1
Lang: 17
C75: 9
Lab: 15
C50: 5
C75: 12
Lang: 20
C50: 4
C75: 14
C75: 8
C75: 6
Lang: 18
Lang: 19
Lang: 17
1.5hr common hr
C75: 10
Lab: 15
Lang: 20
C75: 13
C50: 3
C75: 11
Lang: 18
C50: 0
Lang: 19
C50: 1
Lang: 17
C75: 9
Lab: 15
C50: 5
C75: 12
Lang: 20
C50: 4
C75: 14
C75: 8
C75: 6
Lang: 18
Lang: 19
Lang: 17
1.5hr common hr
Lab: 15
Lang: 20
C50: 3
C50: 0
C50: 1
Lang: 18
Lang: 19
Lang: 17
1.5hr common hr
C50: 5
C75: 10
Lab: 15
Lang: 20
C50: 4
C75: 13

7 Discussion

The new schedule was generated after running the code for under 2 minutes. The configuration of the PC used for executing the code is:

- CPU model: AMD Ryzen 7 PRO 7840U w/ Radeon 780M Graphics, instruction set [SSE2—AVX—AVX2—AVX512]
- Thread count: 8 physical cores, 16 logical processors, using up to 16 threads

The schedule above differs from the current Dickinson's course time-slot schedule in that the model only requires three 75-minute standard course slots and two 50-minute course slots to not overlap with standard course slots of another type. The rest of the standard course slots can overlap with the following pattern: if a 50-minute standard course slot's meeting overlaps with a 75-minute course's meeting, either both meetings start at the same time, or end at the same time.

This presents a new challenge in course scheduling because, for standard course slots, not only can courses conflict if they are scheduled in the same course slot, but they can also conflict with other courses that are scheduled in an overlapping standard course slot.

8 Conclusion

In conclusion, the generation of course time-slot schedules with a linear programming model is an efficient and mathematically optimal way to create a schedule that obeys a specific set of rules as long as those rules can be converted to constraints in the model. Our model has helped to give a quick visualization of Dickinson's course time-slot schedule if a few requirements are altered. In fact, with our model, changes in requirements can be implemented by tweaking the model instead of building it anew. This research has given a perspective into the automation of creating course time-slot schedules based on a set of requirements which has the potential of speeding up the development process of course time-slot schedules at Dickinson and at other colleges as well.

9 Reference

1. J. Havas, A. Olsson, J. Persson, and M.S. Schierscher. Modeling and optimization of university timetabling - a case study in integer programming. Göteborgs universitets publikationer, 2013.