Homework Assignment 7 – [30 points]

STAT430 Mathematical Optimization – Fall 2025

Questions:

Case Studies 1 and 2 in this pdf. You will need to code parts of these case studies in the attached Jupyter notebooks as well.

Video Question:

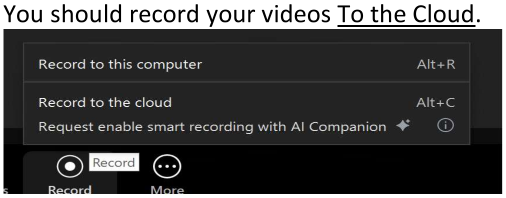
- Explain **Stage 1 (including code).**
- Pretend you are a TA for this class and record a 3-4 minute video explaining what you did here [to a student who hasn't taken this class] for this **particular question number** in this homework assignment.
- Share your screen, showing your answers.

IMPORTANT Video Element of ALL Homework Assignments:

- In order to receive points for each video submission, you need to do **ALL** of the following.
    - Have your camera on.
    - Show your FULL screen in Zoom (not just a particular application).
    - We should be able to hear the audio. Make sure to turn your mic on.
    - You should give a good faith attempt to answer the prompt.
    - Your video meet the minimum time requirement.
    - It should not sound like you are just reading off a script.
    - It's ok if your video recording is not the most eloquent. What's important is that you are putting together YOUR authentic thoughts on your particular understanding of the assignment and the lecture content.

How to Submit Videos:

- You should record your videos in your UIUC Zoom client.
- You should record your videos To the Cloud.



- You can find your recording link at https://illinois.zoom.us/recording/.
- Click on the corresponding video and Copy shareable link to paste the link to the video prompt in Canvas.

Problem	Points
Case Study 1: Conference Scheduling	
1.1	1
1.2	8
2.1	1.5
2.2	1.5
2.3	3
3.1	3
3.2	1
3.3	3
3.4	1.5
Case Study 2: Sudoku	
1	4.5
Video Question	2

<u>Case Study 1</u>: Conference Scheduling (same as in Assignment 6)

Problem Statement

You are responsible for scheduling talks at a one-day academic conference that features a lineup of 30 invited speakers, including 5 keynote speakers. The conference consists of 10 time slots spread across 3 parallel tracks (rooms 1, 2, 3). Each time slot on each track must be filled, and every invited speaker must present exactly once.

Time Slot	Time
1	9am
2	10am
3	11am
4	12pm
5	1pm
6	2pm
7	3pm
8	4pm
9	5pm
10	6pm

Stipulations

Your task is to assign speakers to rooms and time slots in a way that adheres to the following constraints:

1. <u>Keynote Speakers</u>: There are 5 keynote speakers, and no two can present at the same time across any rooms.
2. <u>Room Capacity</u>: Each room must host exactly one talk per time slot (no empty slots).
3. <u>Availability</u>: Speakers can only present in time slots where they are available (see ***unavailable*** time slots below).
4. <u>Sequencing Agreements</u>: Some keynote speakers have agreements to present immediately after another speaker, in the same room.
5. <u>Every Speaker Presents</u>: Each speaker must present exactly once.

Data

The table below lists the following for each invited speaker

a. The time slots they are **<u>unavailable</u>**
b. The time slot that would <u>prefer</u> the most.
c. Whether they are a <u>keynote speaker</u> or <u>regular speaker</u>.
d. Whether or not they have a <u>sequencing agreement</u> and if so, which speaker they must present exactly after in the same room.

Goal

Ideally we would like for as many presenters as possible to have their <u>preferred</u> time slot

Speaker	Tier	Unavailable_Time_Slots	Must Present After the Following Speaker	Time Slot Preference
1	Keynote	[4, 5, 6, 7, 8, 9, 10]	6	3
2	Keynote	[1, 5, 6, 7, 8, 9, 10]		4
3	Keynote	[1, 2, 3, 4, 8, 9, 10]	10	6
4	Keynote	[1, 2, 3, 4, 5, 9, 10]		7
5	Keynote	[4, 5, 6, 7, 8, 9, 10]		4
6	Regular	[3, 4, 5, 6, 7, 8, 9, 10]		2
7	Regular	[1, 2, 5, 6, 7, 8, 9, 10]		3

8	Regular	[1, 2, 3, 4, 7, 8, 9, 10]		6
9	Regular	[1, 2, 3, 4, 5, 6, 9, 10]		7
10	Regular	[1, 2, 3, 7, 8, 9, 10]		9
11	Regular	[4, 5, 6, 7, 8, 9, 10]		1
12	Regular	[1, 5, 6, 7, 8, 9, 10]		2
13	Regular	[1, 2, 3, 4, 8, 9, 10]		5
14	Regular	[1, 2, 3, 4, 5, 9, 10]		7
15	Regular	[1, 2, 3, 4, 5, 6, 7]		8
16	Regular	[5, 6, 7, 8, 9, 10]		2
17	Regular	[1, 6, 7, 8, 9, 10]		3
18	Regular	[1, 2, 3, 4, 5, 9, 10]		6
19	Regular	[4, 5, 6, 7, 8, 9, 10]		1
20	Regular	[1, 2, 6, 7, 8, 9, 10]		5
21	Regular	[1, 2, 3, 4, 5, 9, 10]		7
22	Regular	[1, 2, 3, 4, 5, 6, 7, 8]		9
23	Regular	[3, 4, 5, 6, 7, 8, 9, 10]		2
24	Regular	[1, 5, 6, 7, 8, 9, 10]		4
25	Regular	[1, 2, 3, 4, 9, 10]		6
26	Regular	[1, 2, 3, 4, 5, 10]		8
27	Regular	[1, 5, 6, 7, 8, 9, 10]		2
28	Regular	[1, 2, 3, 7, 8, 9, 10]		4
29	Regular	[1, 2, 3, 4, 5, 6, 10]		7
30	Regular	[1, 2, 3, 4, 5, 6, 7, 8]		9

Stage 1: Checking Initial Feasibility

1.1. Model Formulation

Above is the same problem statement that we saw in assignment #6 (except now we've listed the UNAVAILABLE times for each speaker). We're skeptical that there exists even ONE feasible schedule that meets ALL of our constraints stipulated above. Formulate a model that first checks whether there exists at least one solution that meets all of our model constraints.

1.2. Coding

Now, use PuLP to run this model and check if at least one feasible solution exists for this model.

Stage 2: Relaxation - Not All Bands Have to Present

2.1. Model Relaxation and Newly Discovered Goal that Takes Precedence

In Stage 1 above, you should have found that this model is infeasible. Unfortunately, it may be the case that:

- not every speaker may get to present at the conference
- and thus not every room and timeslot may have a speaker presenting.

Constraint Relaxations

So first let's try *relaxing* two of our original types of constraints in the following way.

	Original Constraint	**Relaxed Constraint**
5 (number of speaker presentations)	Each speaker must present in exactly one room and timeslot.	Each speaker must present in at most one room and timeslot.
2 (room capacity)	Each room and time slot must have exactly one speaker presenting.	Each room and time slot must have at most one speaker presenting.

Newly discovered goal that takes higher precedence

In addition, given that we're assuming that not all speakers might get to present, we've discovered a *new* goal that is more important than our originally stated goal of trying to maximize the room/timeslot PREFERENCE of each speaker. Rather, it is of the most importance that we try to maximize the total number of distinct speakers who get to present in the conference overall.

Formulate this new model with these relaxed constraints, using an objective function that maximizes the total number of speakers who get to present overall.

2.2. Coding

Now, use PuLP to try to find an optimal solution to this Stage 2 model.

2.3. Examining the Schedule and Revising the Model as Needed

- How many of the 30 speakers got to present in the optimal schedule that we found in 2.2? Which speakers(s) were not able to present in the optimal schedule?
- Does this new schedule end up with one presenter in a "sequence pair" getting the opportunity to present, whereas the other speaker in the pair does not? If so, this is not a situation that we want to allow. If this happened, you should add (or modify) constraints in this new model to ensure that if at least one speaker in a sequence pair is assigned to present, then the other presenter in the pair must also present (in the right timeslot/room sequence).
- If you had to modify your constraints and re-solve your model, how many speakers got to present now? Which speakers (if any) were not allowed to present?
- How many of the 30 (rooms, timeslots) were empty in the optimal schedule that we found in 2.2? Which (room, timeslots) were empty?

<u>Stage 3</u>: Some Bands Get to Present More than Once

<u>Two Important Goals</u>

Having empty rooms for a given timeslot during the conference is highly unideal to the conference organizer. So the organizer has now discovered two goals that she would like to satisfy *in this order*.

- <u>Goal 1</u>: Maximize the total number of speakers that get to present <u>exactly once</u>.
- <u>Goal 2</u>: And then, if there are any empty rooms/timeslots (which we've already seen there will be), allow for some speakers to present multiple rooms/timeslots to fill them.

<u>Modifications to the Stage 2 BILP</u>

There are most likely many ways to try to pursue these two goals (with the order of importance that we expressed), but the following is the approach that we will explore. We will do this by making three modifications to our Stage 2 BILP.

1. <u>Number of Rooms per Room/Time Slot</u>
   Let's now try enforcing our same constraint from stage 1 again in stage 3 for the room capacity. That is, let's now force each room and time slot to have exactly one speaker.

	Stage 1	**Stage 2**	**Stage 3**
2 (room capacity)	Each room and time slot must have <u>exactly one speaker</u> presenting.	Each room and time slot must have <u>at most one</u> speaker presenting.	Each room and time slot must have <u>exactly one speaker</u> presenting

2. <u>Number of Rooms/Times Slots per Speaker Constraint</u>

   Rather than stipulate that each speaker must present in <u>at most one</u> room/timeslot, let's now ALLOW for speakers to present in more than one room/timeslot. However, it is also ok if they present in 0 or 1 rooms/time slots as well.

	Stage 1	**Stage 2**	**Stage 3**
5 (number of speaker presentations))	Each speaker must present in <u>exactly one</u> room and timeslot.	Each speaker must present in <u>at most one</u> room and timeslot.	Speakers are <u>ALLOWED</u> to present in <u>more than one room</u>/timeslot. But it is also ok if they present in 0 or 1 room/timeslot.

3. <u>New Decision Variables and New Objective Function</u>

   However, by now allowing some speakers to present more than once at the conference, if we're not careful we may end up with an optimal solution in which a many speakers may not even get to present once, when it was possible for them to. *(For instance, maybe just 3 speakers end up presenting in all rooms/times).*

   We'd still like to maximize the total number of speakers that get to <u>present at least once</u> at the conference, but make sure that there are no speakers presenting in multiple time/slots and taking away an opportunity for a speaker to present even once.

So what we'd like to do is the following.

a) Create a new set of decision variables

$$y_s = \begin{cases} 1 \quad \textit{if speaker s presents in } \textbf{at least one} \textit{ room/timeslot} \\ 0 \textit{ otherwise (ie. speaker s presents in } 0 \textit{ room/timeslots)} \end{cases}$$

b) Maximize $\sum_{s=1}^{30} y_s$

3.1. Decision Variable Defining Constraints

For each speaker s come up with a constraint that *forces* our s to be defined as above, given what your given solution values are$(ex\text{: } x_{srt})$.

	Original Constraint (Stage 1)	**Relaxed Constraint (Stage 2)**	**Changed Constraint (Stage 3)**
5 (number of speaker performances)	Each speaker must perform on <u>exactly one</u> room and timeslot.	Each speaker must perform on <u>at most one</u> room and timeslot.	The constraint that defines your y_b decision variables.

<u>Hint</u>: A minor modification of the Big-M technique we learned in class will help you do this.

<u>Hint</u>: This new constraint that you come up with (for each speaker s) will REPLACE the corresponding constraint of type (5) that you had in Stage 1 and 2.

<u>Tip</u>: If you absolutely get stuck on this question, I can give you a small penalty point hint for how to do this (if you ask for it), so you can proceed with the rest of the analysis.

3.2. Full Model Formulation

Formulate your Stage 3 model altogether by:

- Modifying your objective function
- Modifying your type 2 constraints (back to what we had in stage 1)
- Modifying your type 5 constraints (which also force the definition of our new y_b variables).

3.3. Coding

Now formulate this model in PuLP and try to find an optimal solution to this model.

3.4. Examining the Schedule

- Were we able to find a feasible solution with our Stage 3 model?
- How many speakers got to present at the conference? Which speakers were left out?
- How many rooms/timeslots were filled in this schedule?
- Did any speakers present more than one room/timeslot? If so, which speakers?

Case Study 2:

Formulate your Sudoku problem from Assignment 6 in PuLP to find a solution to the puzzle. The Sudoku_Cell_Data.csv file may help.

Same as in Assignment 6.

5		9				4		
7		8	3		4	9		
6		1				7	3	
4	6	2	5					
3	8	5	7	2	1	6	4	9
1	9	7	4		8	2		
2			1					4
		3		4			8	7
	7			5	3			6