The Making of Department Schedules

An effort to optimize preferences

The Question

Can we use Python to create a schedule for the Math department that allows us to create an optimized schedule based on professors' preferences?

A table with all the preferences of all the teachers for each time and each class. The goal is to then optimize the preferences.

Instructor	Math 109	Math 246	Math 201	Math 215	Math
Deb	5	0	0	0	
Jessica	0	5	0	0	
MarieClaire	0	5	0	0	
Warren	4	3	1	5	
Aziz	0	5	0	0	
Sam	0	5	0	0	

Prof_name	0	1	2	3	
Deb	5	5	5	5	
Jessica	5	5	5	5	
MarieClaire	0	5	5	5	
Warren	5	5	5	5	
Aziz	0	0	0	0	
Sam	0	0	0	0	

			Preference
Course	Class_time_number	Instructor	
DS 140	0	Deb	0
	1	Deb	0
	2	Deb	0
	3	Deb	0
	4	Deb	0
...	...	...	...
Math 443	80	Chloe	0
	81	Chloe	0
	82	Chloe	0
	83	Chloe	0
	84	Chloe	0

			assigned	Preference
Course	Class_time_number	Instructor		
DS 140	0	Deb	assigned_DS 140_0_Deb	0
	1	Deb	assigned_DS 140_1_Deb	0
	2	Deb	assigned_DS 140_2_Deb	0
	3	Deb	assigned_DS 140_3_Deb	0
	4	Deb	assigned_DS 140_4_Deb	0
...	...	...	...	...
Math 443	80	Chloe	assigned_Math 443_80_Chloe	0
	81	Chloe	assigned_Math 443_81_Chloe	0
	82	Chloe	assigned_Math 443_82_Chloe	0
	83	Chloe	assigned_Math 443_83_Chloe	0
	84	Chloe	assigned_Math 443_84_Chloe	0

```
schedule_model.maximize((total.Preference*total.assigned).sum())
```

Creating a Model

Using linear algebra, we tell the program what equation we want solved and how we want it solved.

Constraints

- If a class section happens at a time, then it must happen at that time throughout the week or not happen at that time at all.
- A professor can only teach one class at a time
- To increase variability in section offerings if a class has less than 8 sections, it should be offered at different times
- Professors should teach between their minimum and maximum credits
- Classes should be offered between their minimum and maximum number of sections

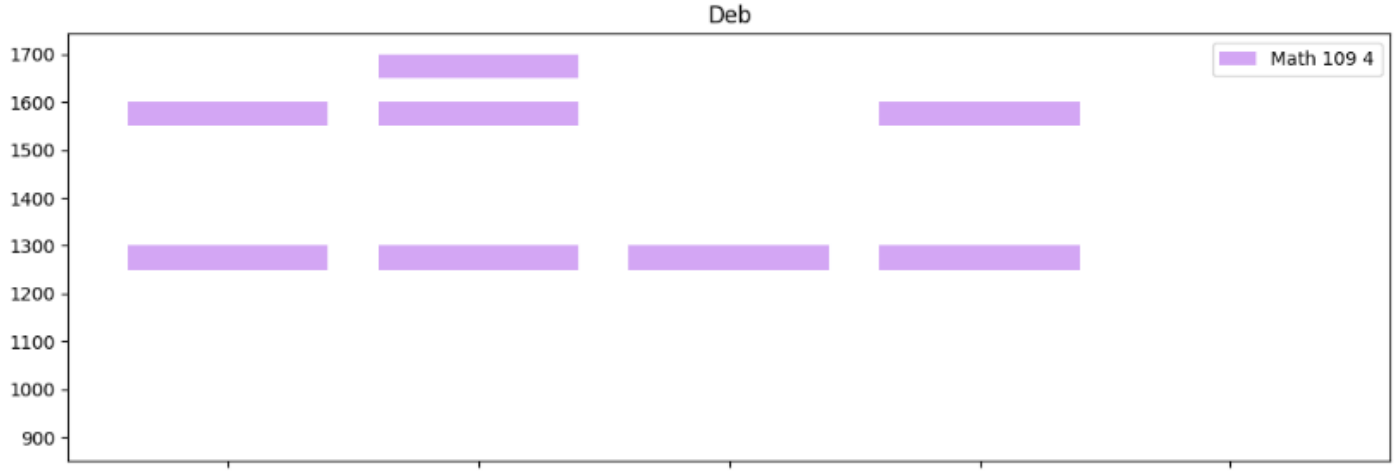
Teaching one class at a time

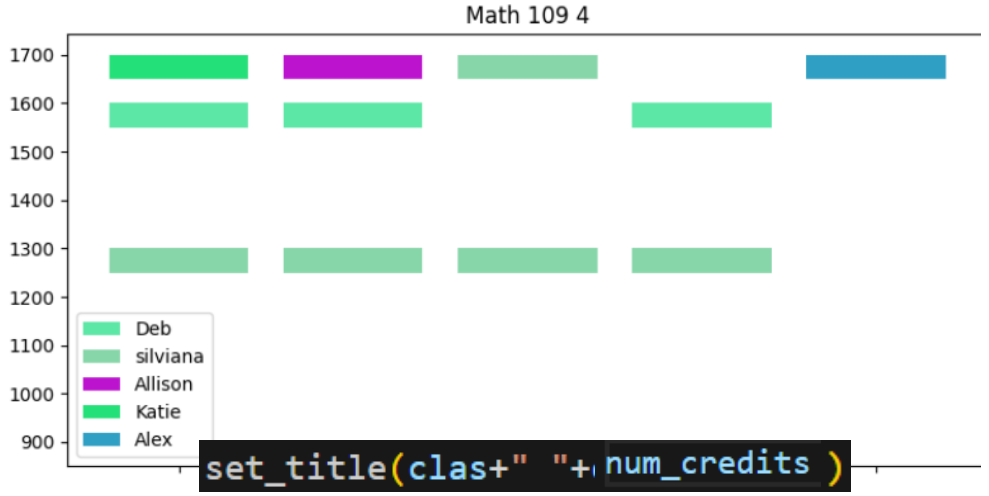
	Instructor	start	end	assigned
0	Deb	08:00:00	08:50:00	assigned_DS 140_0_Deb
85	Deb	08:00:00	08:50:00	assigned_DS 740_0_Deb
170	Deb	08:00:00	08:50:00	assigned_Math 109_0_Deb
255	Deb	08:00:00	08:50:00	assigned_Math 201_0_Deb

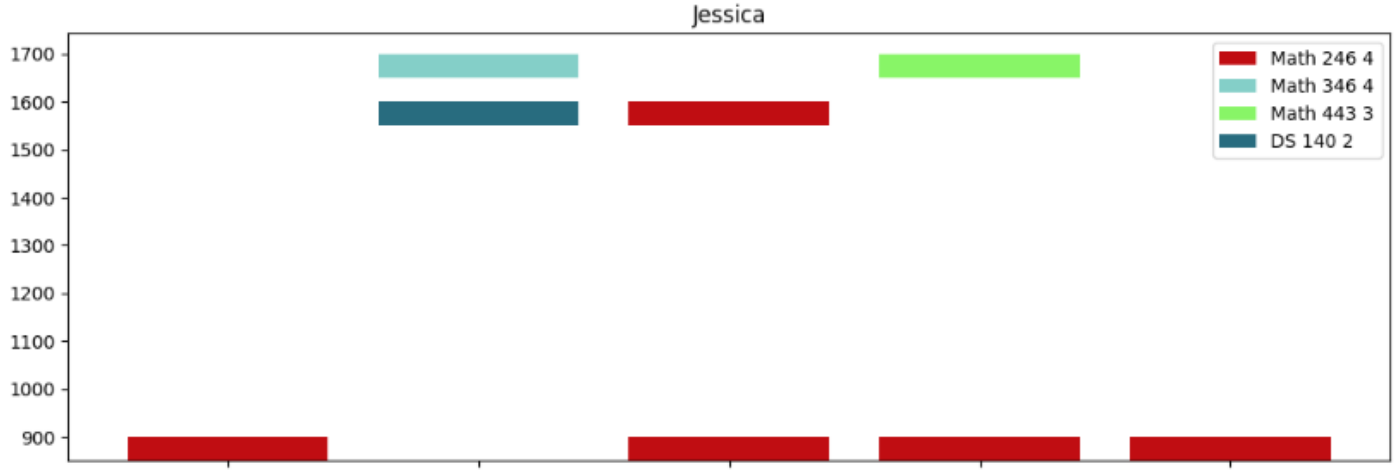
e_1 X_{a_1} s_2 X_{a_2}

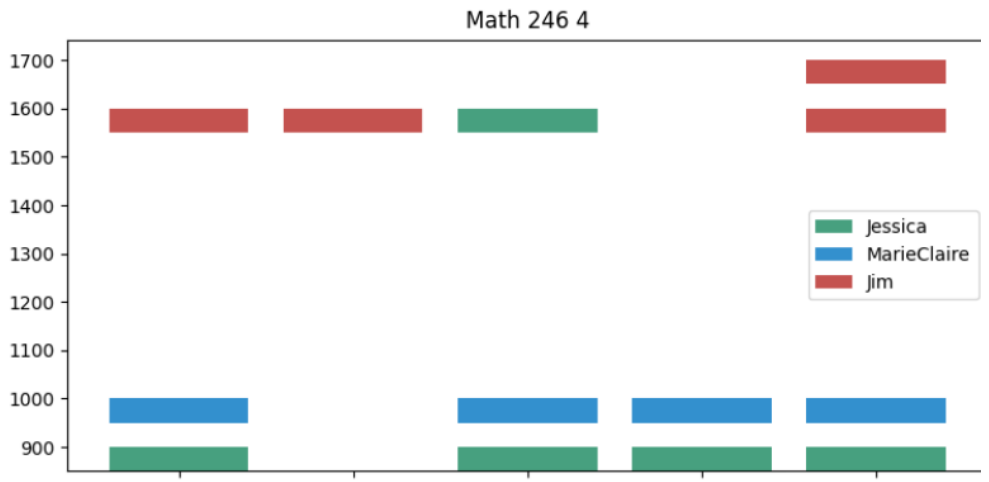
if $s_2 < e_1$,
then
$X_{a_2} + X_{a_1} \leq 1$

```
schedule_model.add_constraint(shift_2.assigned+shift.assigned<=1)
```





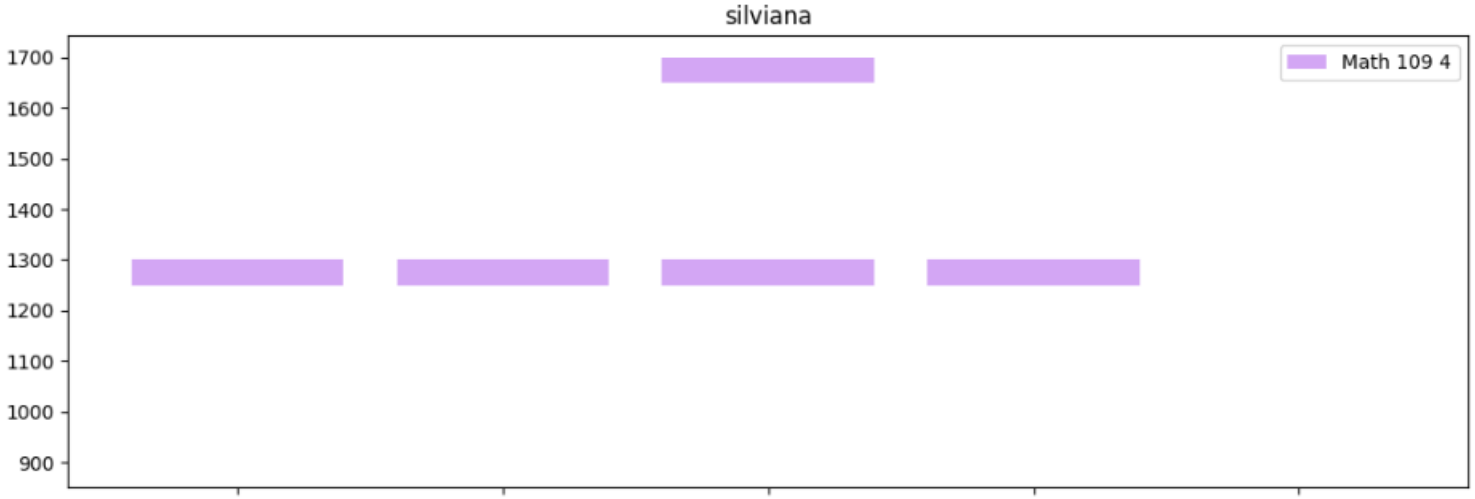


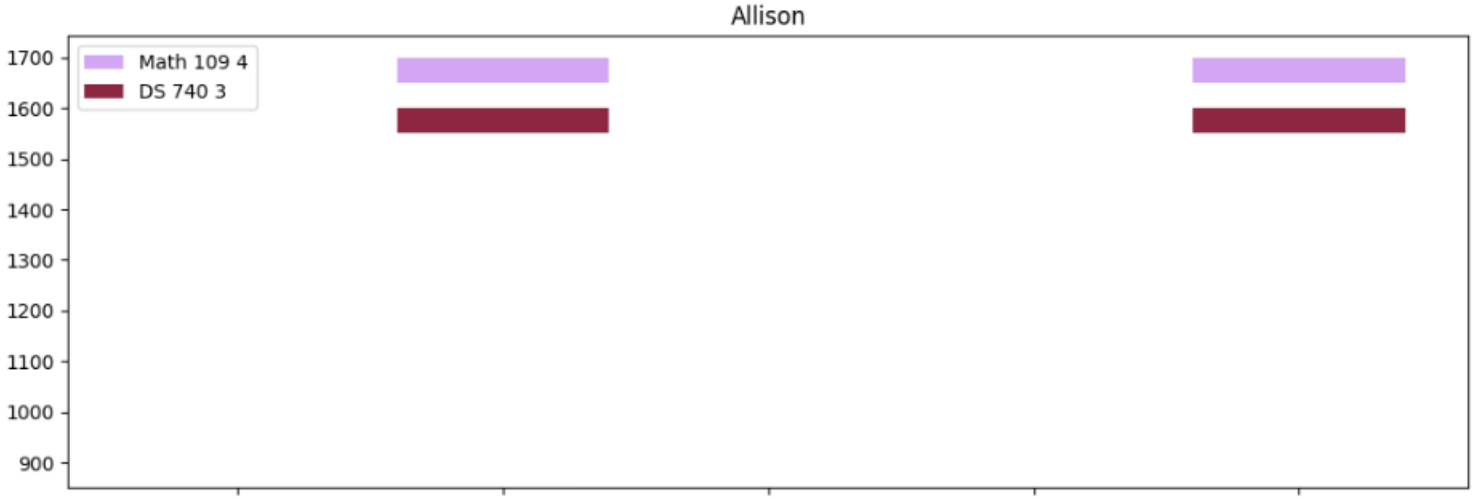


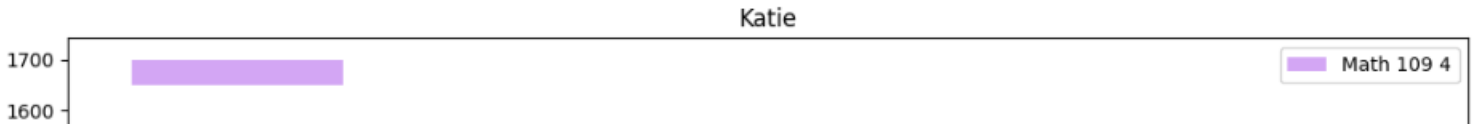
Visualizing the Results

```
, bottom=teacher_class_schedule.End,
```

Creating a visual allows us to see the results in a way that is much easier to understand and follow.








```
for clas in classes:
    class_colors[clas]=np.random.rand(3,)
```

```
label=clas+" "+num_credits
```

Future directions

- assignment across week error
- double check all constraints are working
- readability of code
- improve visualization of schedule

Questions?

Variability in section meetings

	Instructor	start	end	assigned	Course	shift	Day	Goal_number_max
255	Deb	08:00:00	08:50:00	assigned_Math 201_0_Deb	Math 201	(08:00:00, 08:50:00, Mon)	Mon	4
1020	Jessica	08:00:00	08:50:00	assigned_Math 201_0_Jessica	Math 201	(08:00:00, 08:50:00, Mon)	Mon	4
1785	MarieClaire	08:00:00	08:50:00	assigned_Math 201_0_MarieClaire	Math 201	(08:00:00, 08:50:00, Mon)	Mon	4
2550	Warren	08:00:00	08:50:00	assigned_Math 201_0_Warren	Math 201	(08:00:00, 08:50:00, Mon)	Mon	4
3315	Aziz	08:00:00	08:50:00	assigned_Math 201_0_Aziz	Math 201	(08:00:00, 08:50:00, Mon)	Mon	4

If $s_2 < e_1$ and $G_{max} < 8$,
then
$$X_{a_2} + X_{a_1} \leq 1$$

Professors and Classes credit max and min

Instructor	num_credits_max	num_credits_min
Deb	8	8

Instructor	credits_per_meeting	assigned
Deb	1.0	assigned_DS 140_0_Deb
Deb	1.0	assigned_DS 140_1_Deb
Deb	1.0	assigned_DS 140_2_Deb
Deb	1.0	assigned_DS 140_3_Deb
Deb	1.0	assigned_DS 140_4_Deb
...	...	...
Deb	2.0	assigned_Math 201_80_Deb
Deb	2.0	assigned_Math 201_81_Deb
Deb	2.0	assigned_Math 201_82_Deb
Deb	2.0	assigned_Math 201_83_Deb
Deb	2.0	assigned_Math 201_84_Deb

$$C_{min} \leq \sum X_a * cpm \leq C_{max}$$

Course	Goal_number_max	Goal_number_min
Math 109	17	13

Course	credits_per_meeting	assigned
Math 109	1.0	assigned_Math 109_0_Deb
Math 109	1.0	assigned_Math 109_1_Deb
Math 109	1.0	assigned_Math 109_2_Deb
Math 109	1.0	assigned_Math 109_3_Deb
Math 109	1.0	assigned_Math 109_4_Deb
...	...	...
Math 109	2.0	assigned_Math 109_80_Chloe
Math 109	2.0	assigned_Math 109_81_Chloe
Math 109	2.0	assigned_Math 109_82_Chloe
Math 109	2.0	assigned_Math 109_83_Chloe
Math 109	2.0	assigned_Math 109_84_Chloe

Assignment across week

	Course	start	hours_of_class	Instructor	assigned	meeting_times_in_a_week	Day
170	Math 109	08:00:00	1	Deb	assigned_Math 109_0_Deb	4	Mon
179	Math 109	08:00:00	1	Deb	assigned_Math 109_9_Deb	4	Tues
188	Math 109	08:00:00	1	Deb	assigned_Math 109_18_Deb	4	Wed
197	Math 109	08:00:00	1	Deb	assigned_Math 109_27_Deb	4	Thurs
206	Math 109	08:00:00	1	Deb	assigned_Math 109_36_Deb	4	Fri

$$\sum X_a = meeting\ times\ in\ a\ week$$

or

$$\sum X_a = 0$$

```
schedule_model.add_constraint((np.sum(df_acrswk_tms.assigned) == meeting_times_week)|(np.sum(df_acrswk_tms["assigned"]) == 0))
```