Crash Report

Summary: The University Course Assignment System optimization code displays inconsistent behavior in assigning course loads to X3 professors. While most instances correctly allocate 1.5 course loads, occasionally, only 1 course load is assigned.

Issue Description: The main concern revolves around the stochastic behavior observed during the optimization process, leading to varying outcomes for X3 professors. My limited expertise in using OR-Tools poses a challenge in addressing this inconsistency effectively.

Impact: The unpredictable assignment of course loads directly impacts the reliability of optimization results, particularly for X3 professors. This inconsistency jeopardizes the intended fulfillment of category-based constraints, potentially resulting in suboptimal course assignments.

Steps to Reproduce:

- Utilize the code with a dataset containing X3 professors and their preferences.
- Execute the optimization process.
- Observe the course assignments for X3 professors, noting the variability between 1 and 1.5 course loads.

Expected Behavior: The code should consistently assign 1.5 course loads to X3 professors, ensuring a stable and reliable optimization outcome.

Additional Information: While the code struggles to consistently meet the condition that X3 professors should teach 1.5 courses, it successfully adheres to the conditions that professors must be assigned courses only if they are in their preferences. Additionally, X1 and X2 category professors are correctly assigned courses according to their specified course load formats.

Consistent Input Data:

Courses	X1_Aiden	X2_Shar	X3_Arvind
FD1	10	8	10
FD2	9	10	8
FD3	8	9	9

Consistent Output Data:

Course	Professor
FD1	X1_Aiden
FD1	X3_Arvind
FD2	X2_Shar
FD2	X3_Arvind
FD3	X2_Shar
FD3	X3_Arvind

Inconsistent Input Data:

Courses	X1_Aiden	X1_Bella	X1_Caleb	X1_Delilah	X1_Ethan	X3_Mohar	X3_Abdul
FDE2	0	1	10	0	10	5	0
FDE5	0	0	0	5	0	0	2
FDC10	0	0	0	0	3	9	7
HDC2	0	4	0	6	0	4	6
HDC1	4	8	0	0	1	0	0
FDC2	0	0	2	0	0	0	0
HDE4	7	0	8	1	4	0	0
FDC11	0	0	0	0	8	8	5
FDC7	5	2	9	4	0	1	3
HDE1	8	9	3	0	0	2	1
HDC3	0	0	0	10	0	0	8
FDC4	1	0	6	0	0	0	0
FDC3	2	0	0	0	5	3	0
FDE4	0	7	0	3	0	0	0
FDC1	0	6	5	0	0	0	0
FDC6	9	0	0	0	0	0	0
HDE2	0	0	0	2	0	7	0
HDC4	0	10	0	0	0	10	0
HDE3	3	0	0	8	9	0	10
FDE3	0	3	0	0	0	6	0
FDC9	6	0	1	0	0	0	4
FDC8	0	0	0	0	0	0	0
FDE1	0	5	4	9	2	0	0
FDC5	10	0	0	0	6	0	0
FDE6	0	0	7	7	7	0	9

Inconsistent Output Data:

Course	Professor
FDE2	X1_Jasmine
FDE2	X3_Uriel
FDE5	X2_Lily
FDE5	X2_Penelope
FDC10	X3_Donald
FDC10	X3_Mohar
HDC2	X1_Gabriel
HDC2	X3_Yosef
HDC1	X2_Penelope
HDC1	X2_Rose
FDC2	X2_Kaden
FDC2	X3_Zari
HDE4	X2_Kaden
HDE4	X3_Tessa
FDC11	X3_Violet
FDC11	X3_William
FDC7	X1_Caleb
FDC7	X3_Tessa
HDE1	X1_Bella
HDE1	X3_Xena
HDC3	X3_Violet
HDC3	X3_Donald
FDC4	X2_Quentin
FDC4	X2_Rose
FDC3	X2_Lily
FDC3	X2_Mason
FDE4	X2_Oliver
FDE4	X3_Daniel
FDC1	X2_Nora
FDC1	X3_Uriel
FDC6	X1_Isaac
FDC6	X2_Mason
HDE2	X3_William
HDE2	X3_Zari
HDC4	X2_Oliver
HDC4	X3_Mohar
HDE3	X1_Ethan
HDE3	X3_Abdul
FDE3	X3_Sebastian
FDE3	X3_Yosef
FDC9	X1_Fiona
FDC9	X1_Hannah
FDC8	X3_Sebastian
FDC8	X3_Xena
FDE1	X1_Delilah
FDE1	X2_Nora
FDC5	X1_Aiden
FDC5	X3_Daniel
FDE6	X2_Quentin
FDE6	X3_Abdul

Observation: In this case, the code exhibits inconsistent behavior by occasionally assigning only 1 course load to X3 professors instead of the expected 1.5 course loads. This inconsistency contradicts the specified category-based constraints.

Additional Notes: While the code consistently meets the condition that professors must be assigned courses only if they are in their preferences and correctly handles X1 and X2 category professors, the issue arises specifically in the assignment of course loads to X3 professors. The challenge lies in addressing this inconsistency due to my limited expertise in using OR-Tools effectively.

The provided examples illustrate the variability in outcomes, emphasizing the need for further investigation to achieve a stable and reliable optimization outcome, specifically for X3 professors. You can reproduce these examples by running `ProfAllocatorSingleOutput.py` with input as `ModelInput0.csv` and `ModelInput1.csv` respectively.