Artificial Intelligence
Assignment - 2

Name: Tanish Pagaria
Roll No.: B21AI040

TA Allocation as a CSP

Provided information

- A list of courses along with their levels {100, 200, 300, 400, 600, 700}, credit structure, number of students registered, and offered for (e.g., the course CSL3090 is offered for UG3 and UG4 students).
- A list of TAs along with their programs and batches (UG3-AI/CSE, UG4-AI/CSE, MTech-AI/CSE first and second year, MTech-PhD AI/CSE first to fifth year, PhD first to fifth year).
- The preference for allocation should be as follows:
  PhD/MTech-PhD second year onwards > MTech second year > PhD/MTech-PhD/MTech first year > UG4 > UG3

Necessary constraints

- A TA assigned to a course is expected to have done that (or a similar) course with a "decent" grade and should be willing to work as a TA in that course.
- Each course must be assigned the required number of TAs. We may expect to have 1 TA per ~90 students per credit (e.g., 2 TAs in a 3-0-0 course with ~60 students) and no TA if there are <20 students, irrespective of the credit structure. This should be taken as an input by your code, i.e., "1 TA per credit per approximately how many students = ".
- One student may be eligible to become a TA in more than one course (i.e., one value may be present in the domain of more than one variable); however, they may be assigned to exactly one course, if at all.
- There should be at least one PhD / MTech-PhD TA per 100 students.
- A course cannot have more than 60% UG TAs.
- A student may be assigned as a TA only to those courses offered to their (academically) junior batches. E.g., a UG3 TA can be assigned to only UG1/UG2 courses, an MTech first year TA can be assigned to only UG courses, etc.

Courses Data Creation

The data for the courses was generated by taking reference to the academic curriculum of the CSE Department at IIT Jodhpur. A total of **23 courses** were used in the dataset creation.

	S. No.	Course Code	Course Name	Offered for	Credits	No. of Students
0	1	CSL1010	Introduction to Computer Science	UG-1	3-0-2	250
1	2	CSL2020	Data Structures and Algorithms	UG-2	3-0-2	250
2	3	CSL2070	Software Engineering	UG-2	3-0-2	125
3	4	CSP2020	Human-Machine Interaction	UG-2	0-0-4	160
4	5	CSL2040	Maths for Computing	UG-2	3-1-2	160
5	6	CSL2050	Pattern Recognition and Machine Learning	UG-2	3-0-2	250
6	7	CSL2080	Principles of Computer Systems - I	UG-2	2-0-2	60
7	8	CSL2090	Principles of Computer Systems - II	UG-2	3-0-2	60
8	9	CSL3020	Principles of Programming Languages	UG-3	3-0-0	125
9	10	CSL3010	Design and Analysis of Algorithms	UG-3	3-1-0	175
10	11	CSL3090	Artificial Intelligence	UG-3, UG-4	3-0-0	75
11	12	CSL3080	Computer Architecture	UG-3	3-0-0	200
12	13	CSL3050	Computer Networks	UG-3	3-0-3	125
13	14	CSL3060	Database Systems	UG-3	3-0-2	125
14	15	CSL4030	Data Engineering	UG-3	3-0-3	60
15	16	CSL3030	Operating Systems	UG-3	3-0-2	125
16	17	CSL4010	Optimization in ML	UG-3, UG-4	3-0-3	175
17	18	CSL7070	Dependable AI	MTech-2, PhD-1	3-0-0	15
18	19	CSL7010	Advanced Data Structures & Algorithms	UG-4, MTech-1, PhD-1	3-0-0	150
19	20	CSL4050	Introduction to Computer Graphics	UG-4, MTech-1, PhD-1	3-0-0	150
20	21	CSL7020	Machine Learning	MTech-2, PhD-1	3-0-0	200
21	22	CSL7030	Software and Data Engineering	MTech-1, PhD-1	3-0-0	200
22	23	CSL7050	DL-Ops	UG-4, MTech-1, PhD-1	0-0-2	100

Teaching Assistant Data Creation

Two datasets were created for the teaching assistant applications, one for complete assignments and the other for incomplete assignments (not adequate). The data was randomly generated by using non-repeating names and roll numbers, deriving programs and branches from the roll numbers, and assigning 3-5 preferences randomly from the courses in the courses data created previously.

The following programs were considered from the **`AI`** and **`CSE`** branches while creating the datasets:

- `UG-3 (B21)`
- `UG-4 (B20)`
- `MTech-1 (M23)`
- `MTech-2 (M22)`
- `PhD-1 (P23)`
- `PhD-2 (P22)`
- `PhD-3 (P21)`
- `PhD-4 (P20)`
- `PhD-5 (P19)`
- `MTech-PhD-1 (D23)`
- `MTech-PhD-2 (D22)`
- `MTech-PhD-3 (D21)`
- `MTech-PhD-4 (D20)`
- `MTech-PhD-5 (D19)`

Assumptions

- The seniority (priority) among the programs based on the constraints:
  - `UG-1: 0`
  - `UG-2: 0`
  - `UG-3: 1`
  - `UG-4: 2`
  - `MTech-1: 3`
  - `PhD-1: 3`
  - `MTech-PhD-1: 3`
  - `MTech-2: 4`
  - `PhD-2: 5`
  - `MTech-PhD-2: 5`
  - `PhD-3: 5`
  - `MTech-PhD-3: 5`
  - `MTech-4: 5`
  - `PhD-4: 5`
  - `MTech-PhD-4: 5`
  - `MTech-5: 5`
  - `PhD-5: 5`
- The grades of the students who applied lie in the range of **A** to **B-**.

We considered only the applications for the students having grades **A**, **A-** or **B** as eligible for becoming TAs and omitted the rest in a particular course.

The complete assignment data contained information about **150 TA applicants**, and the incomplete assignment data contained only information about **25 TA applicants** (not adequate according to the courses data).

Algorithm for solving the Constraint Solving Problem (CSP)
Extracting useful information from the datasets

- Credit calculation using the L-T-P of course:

  $$Credits = L + T + (P/2)$$

- The following information for all the courses was calculated using the constraints provided:
  - Course code
  - Course name
  - Minimum number of TA required, calculated by considering the constraints:
    - No TAs required if ***number of students < 20***
    - 1 TA per credit per approximately how many students
    - At least one PhD/MTech-PhD TA per 100 students
  - Programs to which the course is offered
  - Number of students in the course

- The following information for all the TAs was extracted, corresponding to the courses they have applied for:
  - Seniority (ranking) of the program of the TA
  - Preference number of the course by the TA
  - Roll No. of the TA
  - Grade of the TA

Generating the CSP

- The above information was used to generate the possible combinations of TAs that could be assigned to each course. The obtained information was stored. The combinations can be seen as the **DOMAIN** from which **VARIABLES** can be assigned to the courses, which are seen as the **STATES**.

  The combinations were sorted, giving first priority to the seniority of the program of the student and second priority to the preference number of the student.

- The **CONSTRAINTS** are the previously mentioned constraints to be used while assigning a value from the domain to a state.

- **SOLUTION** refers to the successful assignment of the students as TAs following the constraints.

Algorithm to solve the CSP

- The algorithm starts by considering the first course and an initially empty TA assignment structure.

- It recursively explores different possibilities for assigning TAs to courses. At each step, it looks at the current course and checks if there are available TAs for that course.
  If there are no TAs required in the course, it skips to the next course and continues the search.

- If there are available TAs for the current course, it checks if these TAs are available to be assigned based on certain constraints. If all TAs are available, it proceeds to assign them to the course and marks them as "assigned." It then makes a recursive call to continue the search with the updated TA assignment structure.

- If the algorithm reaches a point where it cannot assign TAs to the current course while satisfying the constraints, it backtracks. It unassigns TAs (removes the assignment) and goes back to the previous step to explore alternative assignments.

- The process continues until all courses have been assigned TAs, or the algorithm exhausts all possibilities. If all courses are assigned TAs while satisfying the constraints, the algorithm returns the valid assignment. Otherwise, it returns "None" to indicate that no valid assignment is possible.

Complete TA Assignment

- Number of student applications = 150
- 1 TA per credit per approximately how many students = 200
- Minimum number of TAs required based on the course dataset (considering the constraints) = 66

	S. No.	Roll No.	Name	Program	Branch	Preference-1	Grade	Preference-2	Grade.1	Preference-3	Grade.2	Preference-4	Grade.3	Preference-5	Grade.4
0	1	P23CS070	Sarita Malik	PhD-1	CS	Principles of Computer Systems - II	B-	Data Engineering	A-	Software Engineering	B-	NaN	NaN	NaN	NaN
1	2	P20CS023	Ankur Kumar	PhD-4	CS	Operating Systems	A	Introduction to Computer Graphics	B-	Data Structures and Algorithms	A-	Principles of Programming Languages	A	Introduction to Computer Science	B-
2	3	D21AI021	Vishal Singh	MTech-PhD-3	AI	Software Engineering	B	Maths for Computing	B	Principles of Computer Systems - I	B	Design and Analysis of Algorithms	B	NaN	NaN
3	4	D19CS047	Monika Choudhary	MTech-PhD-5	CS	Software and Data Engineering	B-	Principles of Programming Languages	B	Data Engineering	A-	Advanced Data Structures & Algorithms	B-	NaN	NaN
4	5	P19CS026	Alok Singhania	PhD-5	CS	Human-Machine Interaction	B-	Introduction to Computer Graphics	B	Principles of Computer Systems - I	B	Introduction to Computer Science	B-	Software Engineering	B
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
145	146	D23AI021	Smita Yadav	MTech-PhD-1	AI	Operating Systems	A-	Principles of Computer Systems - I	A-	Principles of Computer Systems - II	B-	NaN	NaN	NaN	NaN
146	147	P20CS062	Priya Singhania	PhD-4	CS	Advanced Data Structures & Algorithms	B-	Human-Machine Interaction	B	Artificial Intelligence	A	Introduction to Computer Science	B	NaN	NaN
147	148	P19CS098	Alok Jaiswal	PhD-5	CS	Computer Architecture	B-	Artificial Intelligence	A	Computer Networks	A	NaN	NaN	NaN	NaN
148	149	P19CS070	Rahul Chauhan	PhD-5	CS	Design and Analysis of Algorithms	A	Data Structures and Algorithms	B	Pattern Recognition and Machine Learning	B-	Optimization in ML	B	Database Systems	A
149	150	P20CS017	Sarita Garg	PhD-4	CS	Principles of Computer Systems - II	B-	Software and Data Engineering	B-	DL-Ops	B-	Operating Systems	B	NaN	NaN

150 rows × 15 columns

Observations (before applying the algorithm)

The data used had a significantly higher number of TA applications than the minimum requirements; hence, there was a high probability that all the constraints would be satisfied (assuming randomness in our favor) and the assignment would be performed completely.

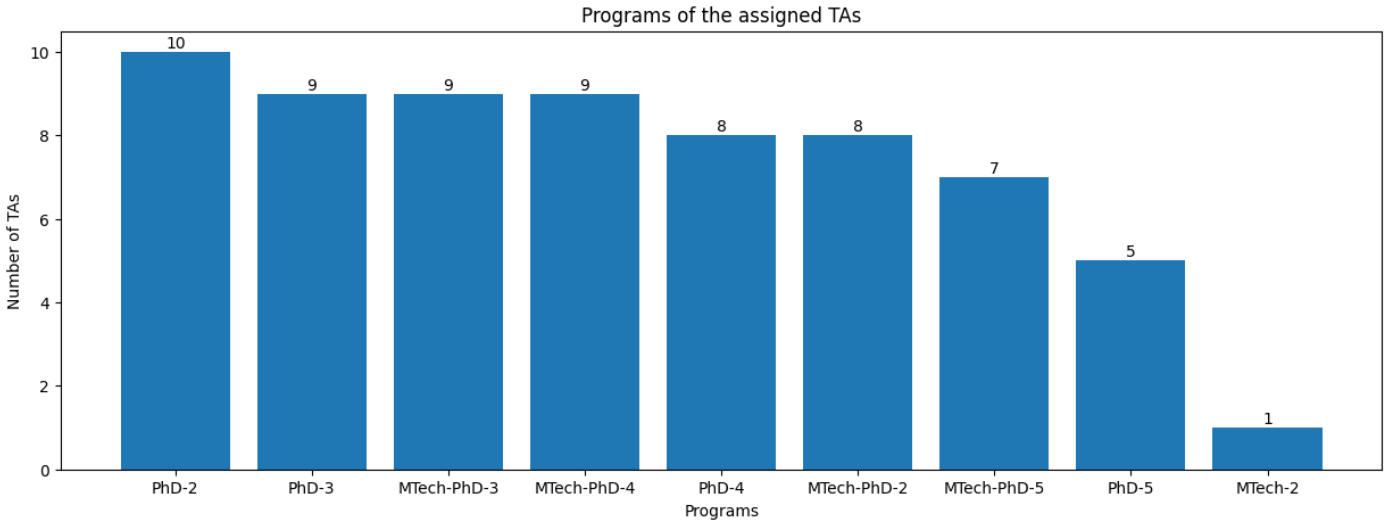
Observations (after applying the algorithm)

- Number of TAs assigned successfully = 66
  It was the same as the minimum number of TAs required (according to the constraints).

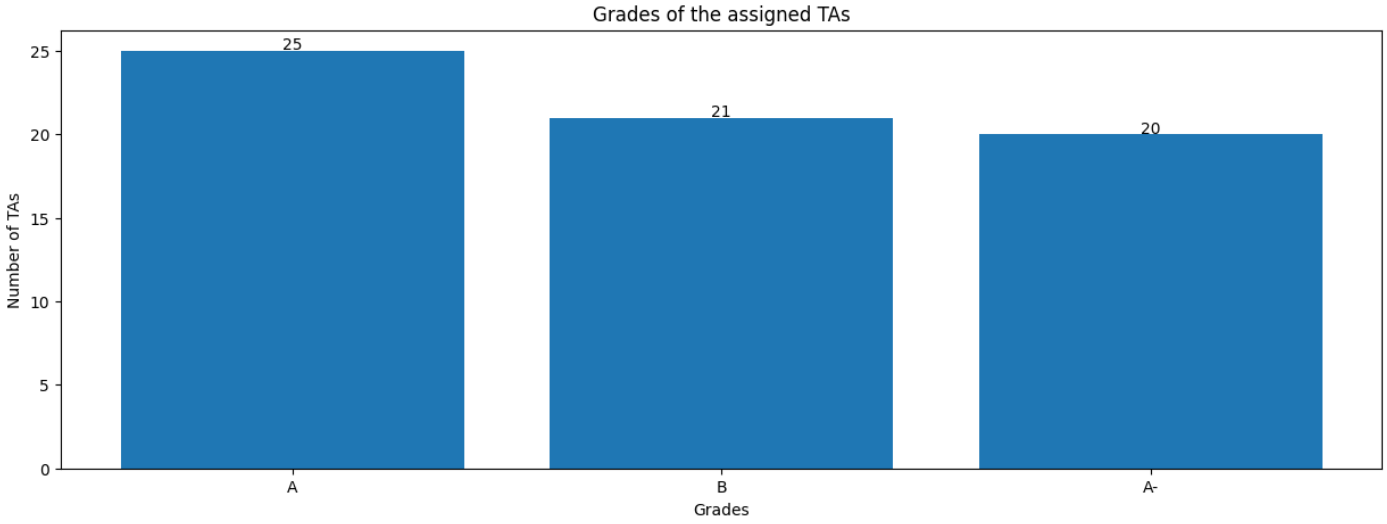
	Course	Roll No.	Name	Program	Branch	Preference Number	Grade
0	Introduction to Computer Science	P20CS056	Sanjay Gandhi	PhD-4	CS	1	B
1	Introduction to Computer Science	P21CS063	Pallavi Dubey	PhD-3	CS	2	B
2	Introduction to Computer Science	D21CS096	Krishna Chauhan	MTech-PhD-3	CS	2	A-
3	Introduction to Computer Science	D21CS036	Rajesh Gandhi	MTech-PhD-3	CS	3	A-
4	Introduction to Computer Science	D19AI097	Ashish Malik	MTech-PhD-5	AI	3	A-
...	...	...	...	...	...	...	...
61	Machine Learning	D20CS083	Savita Agarwal	MTech-PhD-4	CS	2	A-
62	Software and Data Engineering	P22CS036	Rajesh Singh	PhD-2	CS	1	A-
63	Software and Data Engineering	P22CS010	Sangeeta Mishra	PhD-2	CS	1	A-
64	Software and Data Engineering	D19CS064	Ashish Saxena	MTech-PhD-5	CS	1	A
65	DL-Ops	P20CS083	Sanjay Gupta	PhD-4	CS	1	A-

66 rows × 7 columns

- Total execution time: `0.206036 seconds`



A plot showing the number of assigned TAs against the programs to which they belong



A plot showing the number of assigned TAs against the grades they received

Preference Number	Number of TAs who received the preference number	Percentage
1	43	65%
2	15	23%
3	7	11%
4	0	0%
5	1	2%
Total	66	100%

- Most of the TAs were assigned to their first preference.
- Only one TA was assigned their last preference.

The remaining applicants (150 - 66 = 84) remained unassigned.

	Roll No.	Name	Program	Branch
0	P23CS070	Sarita Malik	PhD-1	CS
1	D19CS047	Monika Choudhary	MTech-PhD-5	CS
2	D21AI004	Krishna Malik	MTech-PhD-3	AI
3	M23CS036	Deepak Jha	MTech-1	CS
4	B20AI004	Prakash Verma	UG-4	AI
...	...	...	...	...
79	B20CS099	Deepak Verma	UG-4	CS
80	D23AI021	Smita Yadav	MTech-PhD-1	AI
81	P20CS062	Priya Singhania	PhD-4	CS
82	P19CS098	Alok Jaiswal	PhD-5	CS
83	P20CS017	Sarita Garg	PhD-4	CS

84 rows × 4 columns

Incomplete TA Assignment

- Number of student applications = 25
- 1 TA per credit per approximately how many students = 200
- Minimum number of TAs required based on the course dataset (considering the constraints) = 66

In this case, none of the students were assigned as TAs because the number of applications was inadequate to satisfy the constraints.