Grundlagen der Künstlichen Intelligenz

Programming Exercise 2: Constraint Satisfaction Problem — 25th November 2021

Gerald Würsching,
Yiran Huang, Zhisheng Zheng and Linge Ai

(last updated 25th November 2021)

Problem 2: Student Presentation in the Course AI

The course **Techniques in Artificial Intelligence** plans to invite 8 students to give presentations of 4 different topics to help others better understand the abstract theoretical knowledge. The topics are: CommonRoad (CR), Constraint Satisfaction Problem (CSP), Logic and Hidden Markov Model (HMM). 8 volunteers will participate in this event: Alice, Bill, Carol, Daniel, Edith, Frank, Grace, Harry. Every volunteer can take part in **at most** 1 presentation. Different time budget will be allocated to students according to different topic:

- CR: 15min/presenter
- CSP: 8min/presenter
- Logic: 12min/presenter
- HMM: 10min/presenter

Note that these are merely 4 topics planned, which don't necessarily have to take place all. Which topic(s) is/are actually going to be presented depend(s) on the given constraints. Now consider the following constraints:

1. The topic CR is complex so that it requires at least 3 presenters, if it is to be presented
2. The topic CSP requires at most 2 presenters, if it is to be presented
3. The topic Logic requires 1-2 presenters, if it is to be presented
4. The topic HMM requires 2-3 presenters , if it is to be presented
5. Alice doesn't present alone. She doesn't want to present with Edith either
6. Alice and Bill don't want to present the topic CSP
7. Carol and Daniel are a couple so they want to present together
8. Edith, Frank and Grace will not present together (neither wants to work with any of the other two)
9. Grace and Harry love challenges so they want to present the topic CR
10. Carol and Harry are good friends and want to present together
11. Bill, Daniel and Edith are in a study group so they want to present as a team
12. Alice is the "Tandem" of Frank so they want to work on the same topic
13. Bill wants to present the topic Logic
14. Considering the limited time of the lecture, the total presentation time shall not exceed 90 min
15. No one will present alone
16. At least 3 topics should be presented
17. All topics should be presented

Model the constraint satisfaction problem in Python. For each of the following subsets of constraints, find the solution, if it exists:

Problem 2.1: { 1 - 10, 13, 16 }

Problem 2.2: { 1 - 5, 7 - 10, 17 }

Problem 2.3: { 1 - 5, 9, 11, 15, 16 }

Problem 2.4: { 1 - 4, 6 - 10, 12, 16 }

Problem 2.5: { 1 - 4, 9, 14 - 17 }

Note that problem 2.5 may not be satisfied.

Programming Framework

For this programming exercise a *Jupyter Notebook* will be used. The template for the exercise can be found in ARTEMIS[1]. To model the constraint satisfaction problem, you should know or look up Python's lambdas, lists and dictionaries. The following steps are required to correctly set up the environment for the programming exercise:

1. **Installation of Anaconda and Download of the AIMA python code** If you do not already have the *Jupyter Notebook* environment installed on your machine, the installation is the first step you have to perform. We recommend to install *Anaconda*, since this will set up the whole environment for you. The template for the programming exercise is based on the code from the *AIMA python*[2] project. Therefore, you first have to download the code from this project before the template can be used. Instructions for installation of Anaconda and AIMA python code can be found in "AIMA Code Installation Instructions" on Moodle[3].
2. **Pull of the template:** Pull the repository with the template from ARTEMIS, which can be done similarly to AIMA python just with the repository link from ARTEMIS. To avoid issues with the relative file paths, we recommend to copy all files contained in the template into the root-directory of the *AIMAcode* project that you downloaded in the previous step.

After completing the above steps, you are all set up to start with the exercise. The main function of the template is the *Jupyter Notebook* **csp.ipynb**, which is also the only file you have to work on. Your task is to model the Student Presentation problem. An example, on how to model a constraint satisfaction problem using the *AIMAcode*, is provided in the notebook **csp_demo.ipynb**. This example is taken from Exercise 3.4.

Submission

For submission, you have to upload the following files in ARTEMIS:

1. Copy **csp.ipynb** (notebook containing your solution for modelling the Student Presentation problem) to the pulled repository.
2. Add and commit the altered notebook and push it to ARTEMIS with

   ```
   git add csp.ipynb
   git commit -m "A_commit_message"
   git push
   ```

   (all within the ARTEMIS repository)

A pass will be awarded only if:

1. you submitted the **correct file** with the **correct name**, as shown above.
2. you **did not zip** your file.
3. you **pushed your files to your ARTEMIS branch**.
4. you **did not change** the **variable names** provided by us within the template.
5. your submitted files can be run in an Anaconda environment (Python 3.7) with the packages provided by the *requirements.txt* in the *aima repository*, the utils.py, the search.py and the csp_programming_exercise.py provided by us **within a reasonable time (under 5 minutes)**.
6. the problem has been modelled correctly using the NaryCSP class from the module csp_programming_exercise.

[1] https://artemis.ase.in.tum.de/courses/159/exercises
[2] https://github.com/aimacode/aima-python
[3] https://www.moodle.tum.de/pluginfile.php/3159598/mod_resource/content/5/AIMAinstallation.pdf

7. like the rest of the programming exercises, this is an individual project and you **must** finish the task on your own. (We will use a plagiarism detection tool and any copied code will annul all bonus exercises from both the copier and the copied person!)

Submission will close on **Friday, 24.12.2021 at 23:59**. Your solution will be graded by ARTEMIS. There will be feedback on formatting errors and rightly solved CSP. Nonetheless, it is very important to follow the instructions exactly!

We offer preliminary checks of your solution and ARTEMIS will show your progress. You can submit your solution multiple times and get feedback for each submission. Your final submission will be checked. We award 1 point if all checks including plagiarism pass.