Fundamentals of Artificial Intelligence

Programming Exercise 2: Constraint Satisfaction Problem — 17 November 2023

Roland Stolz,
Vladimir Popa, Yang Tang (in alphabetical order by last name)

(last updated 20th November 2023)

Problem 2: Traveling Plans

Task Desription

Eight TUM students (**Adam, Bella, Charlie, David, Emily, Fiona, George and Helen**) are planning to travel around the world during the Christmas holiday. There are five available destinations, namely **Barcelona, Beijing, New York City (NYC), Paris and Rome**. You need to help them decide which cities to travel to.

Figure 1: Five cities: Barcelona[1],Beijing[2], NYC[3], Paris[4]and Rome[5]

Relevant information about the cities is listed below:

City	Price of the flight ticktet(€)
European cities	100
Beijing	300
New York City	350

Table 1: The prices of fight ticket from Munich to other cities

Categories	Cities
European cities	Paris, Rome, Barcelona
Cities famous for Skyscrapers	Beijing, NYC
Cities famous for their Coast	NYC, Barcelona
Old Cities	Beijing, Rome
Luxurious Cities	Paris, NYC
Cities famous for their museums	Paris, Barcelona, NYC

Table 2: Categories of cities

Note that: Every student can travel to **at most** 1 city.

Now consider the following constraints:

1. Adam and Bella are good friends, so they want to travel together.
2. George and Helen do not like each other, so they do not want to travel together.
3. The flight to Beijing only has 3 tickets left.

[1]https://toposmagazine.com/barcelona-city-portrait/
[2]https://www.britannica.com/place/Beijing
[3]https://www.hellenicdailynewsny.com/en-us/travel/new-york-city-ranked-3rd-best-city-in-the-world-for-2023
[4]https://www.cntraveller.com/gallery/things-to-do-in-paris
[5]https://www.cruisemapper.com/ports/rome-city-port-13062

4. Charlie wants to travel with either David or Bella, otherwise he will not travel at all.
5. Emily and Fiona want to travel outside of Europe.
6. Adam, Bella, Charlie, George, and Helen want to travel to Europe.
7. Fiona wants to travel to an old city, but not to a city with skyscrapers.
8. Each city should be visited at least once.
9. Bella and Helen want to visit a coastal city together, otherwise they will not travel.
10. At least 3 people should visit Barcelona.
11. Adam wants to either visit a luxurious city or a Non-European city.
12. George does not want to visit Paris.
13. No one wants to visit old cities.
14. Charlie and David either want to go to the coast or to museums together.
15. No more than 5 people should travel outside of Europe.
16. The budget for the flight is 1500 Euro.

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

Problem 2.1: { 2 - 4, 6, 7, 9 - 12, 14 - 16 }

Problem 2.2: { 1 - 5, 9 - 15 }

Problem 2.3: { 1 - 3, 5, 7, 11, 12, 16 }

Problem 2.4: { 1, 4, 5, 10 - 16 }

Problem 2.5: { 2, 3, 6 - 9, 11, 12 }

Problem 2.6: { 2, 6, 8 - 12, 14, 15 }

Note that not all problems can be satisfied.

Programming Framework

For this programming exercise, a *Jupyter Notebook* will be used. To model the constraint satisfaction problem, you should know or look up Python's lambdas, lists and dictionaries. The main function of the template is in the **csp.ipynb** file, which is also the only file you have to work on. An example, on how to model a constraint satisfaction problem using the *AIMA*, is provided in the notebook **csp_demo.ipynb**. This example is taken from Exercise 3.4. The following steps are required to correctly set up the environment for the programming exercise and submission:

1. **Installation of AIMA**: Work through AIMA installation instructions on Moodle[6] (Using Docker is recommended for beginners)
2. **ARTEMIS**: Log into ARTEMIS[7] with your TUM credentials. Find the exercise *Constraint Satisfaction Problems* and follow the installation and submission instructions.

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 a Docker/Anaconda environment (Python 3.7 at least) 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] https://www.moodle.tum.de/mod/page/view.php?id=2323882
[7] https://artemis.ase.in.tum.de/courses/303/exercises

6. the problem has been modeled correctly using the NaryCSP class from the module csp_programming_exercise.
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, 15.12.2023 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.