

The contents of this folder are provided as supplementary materials for "Computational Implementation" by Mehmet Barlo and Nuh Aygun Dalkiran.

The programming language used for all the codes in this folder is Python 3.8. To use these codes, we recommend that you install the program called Anaconda, "a free and open-source distribution of the Python and R programming languages."

Below, we provide the instructions to run the codes in this folder. Please make sure that all the codes are in the same folder before running these codes.

STEP 1. Please launch "Spyder" (The Scientific Python Development Environment) platform using Anaconda navigator. (If you are familiar with programming in Python, you can use another integrated development environment (IDE) as well.)

STEP 2. Open and run "ConstructIndividualsAlternativesStatesInput.py"

This code will ask you to enter (i) the number of individuals, (ii) the number of alternatives, and (iii) the number of states and produces the output "Individuals_Alternatives_States.xlsx".

Please enter the required information in the "Individuals_Alternatives_States.xlsx" before running the next code.

FYI: To see a filled in example of "Individuals_Alternatives_States.xlsx" you can refer to the "Examples_in_the_Manuscript\" folder.

STEP 3. Open and run "ConstructRankingProfiles.py"

This code will produce the outputs "Social_Choice_Correspondence.xlsx" and "Table_of_Rankings.xlsx".

Please fill in the "Social_Choice_Correspondence.xlsx" and SAVE AS "Social_Choice_Correspondence_DATA.xlsx",

Please fill in the "Table_of_Rankings.xlsx" and SAVE AS "Table_of_Rankings_DATA.xlsx" before running the next code.

FYI: To see filled in examples of "Social_Choice_Correspondence_DATA.xlsx" and "Table_of_Rankings_DATA.xlsx" you can refer to the "Examples_in_the_Manuscript\" folder.

STEP 4. Open and run "FromRankingstoChoices.py"

This code will produce the output "Table_of_Choices_DATA.xlsx", which will be used as an input by "ComputeConsistentCollections.py" in Step 5.

STEP 5. Open and run "ComputeConsistentCollections.py"

This code will produce the output "Consistent_Collections.xlsx" that presents the list of all Consistent Collections of the specified SCC.

STEP 6. Open and run "EconomicEnvironment.py"

This code will produce the output "Economic_Environment.xlsx" that presents whether each Consistent Collection satisfies the EE* property.

STEP 7. Open and run "NoVetoPower.py"

This code will produce the output "No_Veto_Power.xlsx" that presents whether each Consistent Collection satisfies the NVP* property.