

Affectations stables

May 1, 2023

```
[3]: import random
      from itertools import permutations
```

```
[4]: def extract_instance_from_file(file):
      D_agences={}
      D_candidats={}
      with open(file,"r",encoding="utf8") as f:
          n = int(next(f))
          for _ in range(n):
              line = next(f)
              line_list= line.strip().split(":")
              D_agences[line_list[0]] = line_list[1:]

          for _ in range(n):
              line = next(f)
              line_list= line.strip().split(":")
              D_candidats[line_list[0]] = line_list[1:]
      return n,D_agences,D_candidats
```

```
[5]: extract_instance_from_file('stable.txt')
```

```
[5]: (5,
      {'A1': ['C3', 'C2', 'C5', 'C4', 'C1'],
       'A2': ['C1', 'C3', 'C2', 'C5', 'C4'],
       'A3': ['C4', 'C5', 'C2', 'C1', 'C3'],
       'A4': ['C3', 'C4', 'C2', 'C5', 'C1'],
       'A5': ['C5', 'C4', 'C2', 'C1', 'C3']},
      {'C1': ['A3', 'A5', 'A1', 'A2', 'A4'],
       'C2': ['A5', 'A1', 'A3', 'A4', 'A2'],
       'C3': ['A3', 'A5', 'A1', 'A4', 'A2'],
       'C4': ['A4', 'A3', 'A2', 'A5', 'A1'],
       'C5': ['A5', 'A3', 'A4', 'A2', 'A1']})
```

```
[6]: def generate_random_instance(n, version_number=1):
      with open(f"GSentries_rand{n}_{version_number}", "w", encoding="utf8") as f:
          f.write(str(n) + "\n")

          list_candidates=["C" + str(x) for x in range(1,n+1)]
```

```

        for agency in range(1, n+1):
            random.shuffle(list_candidates)
            f.write("A" + str(agency) + ":" + ":".join(list_candidates)+
↪ "\n")

list_agencies=["A" + str(x) for x in range(1,n+1)]
for candidate in range(1, n+1):
    random.shuffle(list_agencies)
    f.write("C" + str(candidate) + ":" + ":".join(list_agencies)+
↪ "\n")

```

```
[7]: generate_random_instance(6, version_number=1)
```

```
[10]: def number_of_non_stable_couples(agencyes_assign, candidates_assign,
                                         agencyes_choices, candidates_choices):

    nombre_couple_instable = 0

    for a in agencyes_assign.keys():
        c = agencyes_assign[a]
        index_of_c = agencyes_choices[a].index(c)
        if index_of_c > 0:
            liste_potentiel = agencyes_choices[a][: index_of_c]
            for cand in liste_potentiel:
                agency_of_cand = candidates_assign[cand]

                if candidates_choices[cand].index(a) <
↪ candidates_choices[cand].index(agency_of_cand):
                    nombre_couple_instable+= 1

    return nombre_couple_instable

```

```
[11]: instance = extract_instance_from_file('stable.txt')
n = instance[0]
a_assign = { f'A{i}':f'C{i}' for i in range(1,n+1)}
c_assign = { f'C{i}':f'A{i}' for i in range(1,n+1)}

a_choices = instance[1]
c_choices = instance[2]

```

```
[12]: number_of_non_stable_couples(a_assign, c_assign, a_choices, c_choices)
```

```
[12]: 3
```

```
[13]: def generate_random_assignment(agency_choices, candidates_choices):
    list_of_candidates=list(candidates_choices.keys())
    random.shuffle(list_of_candidates)

    agencies_assignment = { a : c for a,c in zip(agency_choices.
→keys(),list_of_candidates)}
    candidates_assignment = { c : a for a,c in agencies_assignment.items()}
    return agencies_assignment,candidates_assignment

[14]: generate_random_assignment(a_choices, c_choices)

[14]: ({'A1': 'C4', 'A2': 'C2', 'A3': 'C5', 'A4': 'C1', 'A5': 'C3'},
{'C4': 'A1', 'C2': 'A2', 'C5': 'A3', 'C1': 'A4', 'C3': 'A5'})

[15]: def comptes_assignements(agency_assignment,agency_choices):
    l = []
    for key in agency_choices.keys():
        if agency_assignment.get(key) == None or agency_assignment[key] ==
→"None":
        l.append(key)
    return l

[16]: def test_dico(dico,correspondance):
    return dico.get(correspondance) == None or dico[correspondance] == "None"

[17]: def gale_shapley_algorithm(agency_choices, candidates_choices):
    # tant qu'il existe une agence qui n'a pas de candidats on continue
→l'algorithme
    # Initialisation d'un dictionnaire d'assignement
    agencies_assignment = {}
    candidates_assignment = {}
    agences_non_assignees =
→comptes_assignements(agency_assignment,agency_choices)
    while(len(agenes_non_assignees) > 0):
        agence_act = agences_non_assignees[0]
        l_agence_act = agency_choices[agence_act]
        candidat_act_i = 0
        while(test_dico(agency_assignment, agence_act)):
            candidat_act = l_agence_act[candidat_act_i]
            if test_dico(candidates_assignment, candidat_act):
                agencies_assignment[agence_act] = candidat_act
                candidates_assignment[candidat_act] = agence_act
            elif candidates_choices[candidat_act].index(agence_act) <
→candidates_choices[candidat_act].index(candidates_assignment[candidat_act]):
                agencies_assignment[candidates_assignment[candidat_act]] =
→"None"
            agencies_assignment[agence_act] = candidat_act
```

```

        candidates_assignment[candidat_act] = agence_act
        candidat_act_i += 1
        agences_non_assignees = []
        ↪comptes_assignements(agencies_assignment, agencies_choices)

        return (agencies_assignment, candidates_assignment)

```

```
[18]: gale_shapley_algorithm(a_choices, c_choices)
```

```
[18]: ({'A1': 'C3', 'A2': 'C1', 'A3': 'C2', 'A4': 'C4', 'A5': 'C5'},
      {'C3': 'A1', 'C1': 'A2', 'C4': 'A4', 'C5': 'A5', 'C2': 'A3'})
```

```
[ ]:
```