

TP Caml n° 8 Majorité qualifiée

Au Conseil des Ministres de l'Union Européenne, certaines délibérations du Conseil des Ministres sont prises à la "majorité qualifiée". Chacun des $p = 15$ pays a son vote pondéré par les valeurs suivantes :

- Allemagne, Royaume-Uni, France et Italie : 29
- Espagne : 27
- Pays-Bas : 13
- Grèce, Belgique, Portugal : 12
- Suède, Autriche : 10
- Danemark, Finlande, Irlande : 7
- Luxembourg : 4

Le vote est acquis s'il réunit un total d'au moins 169 voix sur les $n = 237$ voix possibles. Deux autres conditions existent : au moins la moitié des états membres doit être favorable (c'est automatique dès que les 169 voix sont atteintes), et les voix doivent représenter au moins 62% de la population européenne. Cette dernière condition ne change la donne que dans un tout petit nombre de situations et nous n'en tiendrons pas compte ici.

Avant de commencer à coder, récupérez le fichier TP8.ml.

1 La procédure de vote

On dispose d'une liste de poids

```
let euro_poids = [29; 29; 29; 29; 27; 13; 12; 12; 12; 10; 10; 7; 7; 7; 4]
```

et d'un seuil `let euro_majorité = 169`. Chaque pays peut voter pour (`true`) ou contre (`false`) et on appelle "configuration" une liste de booléens représentant les votes de tous les pays.

a. Écrire une fonction `total_votes : int list -> bool list -> int` prenant la liste des poids et une configuration et calculant le nombre total de votes pondérés recueillis.

b. En déduire une fonction `résultat_vote : int list -> int -> int bool -> bool` telle que `résultat_vote poids majorité config` donne le résultat du vote.

2 Votes rapportés à la population

Pour essayer de comparer les “pouvoirs” réels des pays sur les votes, on peut tout simplement comparer les pondérations avec les populations de chaque pays. Je vous fournis une liste des populations des Quinze :

```
let euro_populations =  
    [82164; 59623; 59225; 57679; 39441; 15863; 10545;  
     10239; 9997; 8861; 8102; 5330; 5171; 3776; 435]
```

- c. Écrire une fonction `pop_votes1` telle que `pop_votes1 poids populations` renvoie une liste de flottants comprenant pour chaque pays le quotient de son poids par sa population.
- d. Ces résultats n’étant pas très lisibles, on décide de normaliser les listes de flottants en les traitant de manière à ce que le premier élément de la liste soit 1.0. Écrire la fonction `normalise : float list -> float list`. En déduire une fonction normalisée `pop_votes`.

3 Indices de Banzhaf-Coleman

On pourrait penser que la vraie démocratie est atteinte lorsque les poids des votes sont proportionnels à la population. Cependant on peut observer deux phénomènes curieux :

- dans une entreprise pilotée par trois actionnaires des poids `[49; 49; 2]` et où les décisions se prennent à la majorité absolue, chacun des trois actionnaires a en fait le même pouvoir puisqu’il y a besoin de deux actionnaires, et n’importe lesquels, pour avoir la majorité;
- au contraire, dans une assemblée dont les divers partis auraient en nombre de siège `[362; 30; 148; 21; 15]`, le premier parti, ayant la majorité absolue a lui tout seul, dispose d’un pouvoir sans limites.

On cherche donc à avoir une meilleure estimation du “pouvoir” réel des votants dans un système de votes. L’indice de Banzhaf-Coleman est une de ces estimations possibles, partant du principe que le pouvoir d’un pays P_0 est proportionnel au nombre de fois où son vote est décisif dans toutes les situations possibles.

L’indice de Banzhaf-Coleman b dénombre ainsi, parmi les 2^{p-1} configurations possibles des votes des $p - 1$ autres pays, celles où l’apport des voix du pays P_0 permet de passer au-dessus du seuil.

Considérons par exemple le premier actionnaire (49) pour les poids `[49; 49; 2]`. Il y a $2^{3-1} = 4$ configurations possibles pour les deux autres actionnaires, dont deux (`[true; false]` et `[false; true]`) où c’est au premier actionnaire d’arbitrer. Son indice de Banzhaf-Coleman est donc 2. En poursuivant le raisonnement pour les deux autres actionnaires, les indices sont ainsi `[2; 2; 2]` (ou `[1; 1; 1]` une fois normalisés).

De même, dans le cas de l’assemblée avec parti majoritaire, on arrive à des indices de `[1; 0; 0; 0; 0]`.

e. Que valent les indices de Banzhaf-Coleman pour un système de votes avec poids [5; 6; 7; 8] à la majorité absolue (14)?

3.1 Indice pour un pays

f. Écrire une fonction `compte_configs : int -> int -> int list -> int` telle que l'appel `compte_config mini maxi poids` renvoie le nombre de configurations de poids compris entre les valeurs `mini` et `maxi`.

Pour calculer l'indice d'un pays particulier de poids `poids1`, on suppose que l'on a ôté le poids en question de la liste des poids pour obtenir une liste `poidssauf1`.

g. Dédurre de la question précédent une fonction `bancol1` telle que `bancol1 majorité poids1 poidssauf1` calcule l'indice du pays en question.

3.2 Indice pour tous les pays

h. Réaliser une fonction `bancol : int -> int list -> int list` qui calcule les indices de Banzhaf-Coleman pour tous les pays. Tester `bancol` sur les valeurs fournies ainsi que sur les exemples de la page 2.

i. Écrire une fonction `pop_bancol` telle que `pop_bancol majorité poids populations` donne les indices de Banzhaf-Coleman rapportés à la population et normalisés. Écrire aussi une fonction `écart` qui calcule le quotient entre le maximum et le minimum des indices.

j. Quelle est la complexité de ces fonctions?

4 Recherche des poids optimaux

On souhaite réduire l'écart observé entre les pays afin que chaque pays dispose d'un indice de Banzhaf-Coleman proportionnel à sa population. On convient de ne pas changer le seuil `majorité` ni le total des poids n .

k. Combien il y a-t-il de systèmes de poids donnant n votes à p pays?

Il est donc difficile de tester l'ensemble des configurations. On va utiliser un algorithme un peu simple mais efficace : pour une liste de poids donnée, on calcule les indices de Banzhaf-Coleman rapportés aux populations, et on enlève une voix au pays le plus favorisé pour la donner au pays le plus défavorisé.

l. Écrire `iter_bancol` telle que `iter_bancol majorité poids populations` réalise une itération de cet algorithme. Faire des tests.

m. Cette méthode peut-elle converger? Essayer d'écrire une fonction `bancol_optimal` qui donne des poids optimaux pour l'indice de Banzhaf-Coleman. Ne pas hésiter à modifier l'algorithme pour obtenir de meilleurs résultats.

5 Le point de vue des citoyens

Au lieu de comparer les indices à la population de chaque pays, une autre méthode est de considérer ces indices du point de vue de chaque citoyen européen en dénombrant le nombre de fois où

- (1) changeant d'avis, il fait basculer la majorité de son pays,
- (2) son pays changeant d'avis, il fait basculer la majorité européenne.

n.* Proposer des solutions, au besoin approchées, pour cette approche.

Sources

M. Balinski, R. Laraki, "Le pouvoir des votes", *Pour la Science*, numéro 294, avril 2002.