SYSTÈMES DE VOTE

1. Raffinement de la méthode de Condorcet

1.1. Paradoxe de Concorcet

Soient $A_1, \ldots, A_n$, n candidats. Si pour tout $1 \leq i \leq n$: $A_i \prec A_{i+1}$ et $A_n \prec A_1$ alors aucun candidat n'est vainqueur par la méthode de Condorcet.

1.2. Raffinement

Définition 1.1
Plusieurs raffinement possibles :
- Le vainqueur est celui qui gagne le plus de duels (méthode de Copeland).
- Le vainqueur est celui dont la pire défaite est « meilleure » que celle des autres (méthode minimax).
- On peut appliquer la méthode de Bourda après celle de Condorcet.
- On applique la méthode de Kemeny-Yang [1§].
- On utilise la méthode des paires ordonnées.

2. Méthode des paires ordonnées

2.1. Rudiments de théorie des graphes

Définition 2.1
Un*graphe fini*, $G = (V, E)$, est la donnée :

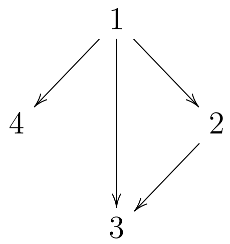
1§. Pour chaque ordre possible $C_{i_1} \prec C_{i_2} \prec \ldots \prec C_{i_n}$ avec $(i_1, \ldots, i_n)$ une permutation de $\{1, \ldots, n\}$. On calcule le score comme étant :

$$\sum_{j<n} \mathrm{Comp}(C_j, C_{j+1})$$

et on prend le maximum.

> 1. d'un ensemble fini V (ensemble des sommets) ;
> 2. d'un ensemble $E \subset \mathcal{P}(V \times V)$ (ensemble d'arêtes).

Exemple. —

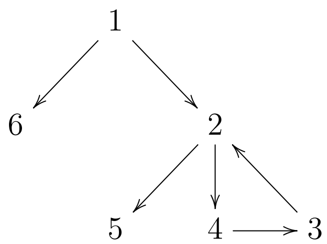


$$V = \{1, 2, 3, 4\} \ ; \ E = \{(1, 2), (2, 3), (1, 3), (1, 4)\} .$$

> Définition 2.2
> Un graphe $G = (V, E)$ est non orienté si pour tous $x, y \in V$, $(x, y) \in E$ implique $(y, x) \in E$.

> Définition 2.3
> Soit $G = (V, E)$ un graphe, soit $\{x_1, \dots, x_n\} = S \subset V$. $(x_1, \dots, x_k)$ est un *chemin* dans G si pour tout $j < k$, $(x_j, x_{j+1}) \in E$.
> S est un *chemin simple* si S est un chemin et pour tous $i, j < k$, $x_i \neq x_j$ si $i \neq j$.

Exemple. —



$(1, 2, 4, 3, 2)$ est un chemin, $(1, 2, 4, 3)$ est un chemin simple.

> Définition 2.4
> S est un *cycle* (ou circuit) de G si c'est un chemin et $(x_k, x_1) \in E$.
> S est un *cycle simple* si c'est un chemin simple et un cycle.

Remarque. — Les définitions sont plus simples dans le cas de graphes non orientés.

Exemple. — Est un chemin est alors de la forme $x_1 - x_2 - x_3 - x_4 - \dots$.

Remarque. — Si $G = (V, E)$ est un graphe orienté, on peut définir son graphe non orienté sous-jacent (appelé aussi symétrisé) de la façon suivante : $\overline{G} = (\overline{V}, \overline{E})$ tel que $\overline{V} = V$ et

$$\overline{E} = \{(x, y), (y, x) \mid (x, y) \in E\} .$$

Définition 2.5
Si $\overline{G}$ est un graphe non orienté. $\overline{G}$ est *connexe* si pour tous $x, y \in E$ il existe S dans V tel que :

1. $S = \{x_1, \ldots, x_n\}$;
2. $x_1 = x$ et $x_k = y$;
3. $(x_i, x_{i+1}) \in \overline{E}$ pour tout $i < k$.

Définition 2.6
G est connexe si, et seulement si, sont symétrisé $\overline{G}$ est connexe.

Définition 2.7
Soient $G = (V, E)$ et $x \in V$. x est une *source* si pour tout $y \neq x$ dans V, $(y, x) \notin E$.

Exemple. —

$$\begin{tikzcd} & 1 \arrow[dl] \arrow[dr] & \\ 2 & & 3 \arrow[ll] \end{tikzcd}$$

1 est une source.

Définition 2.8
Un graphe G est sans cycle s'il n'admet pas de cycle simple orienté.

Exemple. —

$$\begin{tikzcd} & & 1 \arrow[dl] & \\ 4 \arrow[r] & 2 \arrow[rr] & & 3 \arrow[ul] \end{tikzcd}$$

a un cycle $(1, 2, 3)$.

$$\begin{tikzcd} & & 1 \arrow[dl] \arrow[dr] & \\ 4 \arrow[r] & 2 & & 3 \arrow[ll] \end{tikzcd}$$

n'a pas de cycle.

Cette propriété dépend donc de l'orientation du graphe.

Proposition 2.9
Si G est fini et acyclique et $|E| > 1$, alors G a une source.

Démonstration
Supposons que G n'a pas de source. Dans ce cas, on montre que G admet un chemin de longueur arbitraire. Soit $k \in \mathbf{N}$ la longueur maximale d'un chemin de G, soit $(x_1, \ldots, x_k)$ un tel chemin.

$$x_1 \longrightarrow x_2 \longrightarrow \ldots \longrightarrow x_k$$

G n'a pas de source donc il existe $y \in V$ tel que $(y, x_1) \in E$. Donc $(y, x_1, \ldots, x_n)$ est un chemin plus grand.
Comme G est fini et comme G admet des chemins de longueurs plus grandes que son nombre de sommets, G a un cycle.

2.1.1. Pour trouver les cycles

Cas d'un graphe non orienté. — Dans le cas d'un tel graphe, on élimine petit à petit les sommets ayant au plus un voisin.

Cas orienté. — On se base sur un algorithme de parcours en profondeur.

La version récursive se base sur un système de marquage. On pose $G = (V, E)$. On attribut N pour non marqué, T pour temporairement marqué et M pour marqué.

```
def Parcours(G):
        N = V
        M = []
        T = []
        while len(N)!=0:
                x = N[0]
                ParcoursProf(G,x)

def ParcoursProf(G,x):
        if found(x,T):
                return x  # il y a une boucle
        if !found(x,M):
                T.append(x)
                for y in [y for (x,y) in E] # = V(x):
                        ParcoursProf(G,y)
                M.append(x)
                T.remove(x)
                return M
```

Exemple. —

$$\begin{tikzcd}
 & 1 \arrow[dl] \arrow[dr] & & \\
2 & & 3 \arrow[dr] & \\
 & 4 \arrow[ur] & & 5 \arrow[ll] \arrow[dl] \\
 & & 6 &
\end{tikzcd}$$

— $1 \notin T$, $T = \{1\}$, $V(1) = \{2, 3\}$; appel de PProf(G,3) :

$$\begin{tikzcd}
 & 1_T \arrow[dl] \arrow[dr] & & \\
2 & & 3 \arrow[dr] & \\
 & 4 \arrow[ur] & & 5 \arrow[ll] \arrow[dl] \\
 & & 6 &
\end{tikzcd}$$

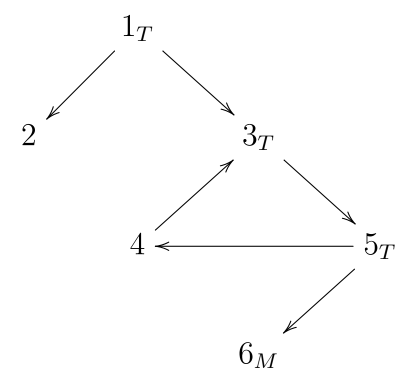
— $3 \notin T$ et $3 \notin M$, $T = \{1, 3\}$, $V(3) = \{5\}$; appel de PProf(G,5) :

$$\begin{tikzcd}
 & 1_T \arrow[dl] \arrow[dr] & & \\
2 & & 3_T \arrow[dr] & \\
 & 4 \arrow[ur] & & 5 \arrow[ll] \arrow[dl] \\
 & & 6 &
\end{tikzcd}$$

— $5 \notin T$ et $5 \notin M$, $T = \{1, 3, 5\}$, $V(5) = \{4, 6\}$; appel de PProf(G,6) :

$$\begin{tikzcd}
 & 1_T \arrow[dl] \arrow[dr] & & \\
2 & & 3_T \arrow[dr] & \\
 & 4 \arrow[ur] & & 5_T \arrow[ll] \arrow[dl] \\
 & & 6 &
\end{tikzcd}$$

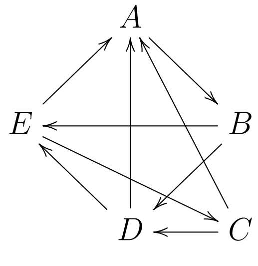
— $6 \notin T$ et $6 \notin M$, $T = \{1, 3, 5, 6\}$ et $V(6) = \emptyset$; terminé pour 6 car pas de voisin donc $M = \{6\}$ et $T = \{1, 3, 5\}$:



— PProf(G,4) puisque 5 a deux voisins : $4 \notin T$ et $4 \notin M$, $T = \{1, 3, 5, 4\}$, $V = \{3\}$ donc on appelle PProf(G,3). 3 est marqué donc on sort et on a un cycle.

2.2. Méthode des paires ordonnées (méthode de Tideman)

Exemple. — On a 5 candidats : $\mathcal{C} = \{A, B, C, D, E\}$.



Comparaison	Score
A, B	55
B, A	45
A, C	35
C, A	75
B, C	12
C, B	88
C, D	76
D, C	24
C, E	33
E, C	67
A, D	21
D, A	79
A, E	49
E, A	51
B, D	70
D, B	30
B, E	52
E, B	48
D, E	53
E, D	47

Aucun candidat n'est vainqueur de CONDORCET.

Tri des duels :

1. C, B : 88
2. D, A : 79
3. C, D : 76
4. C, A : 75
5. B, D : 70
6. E, C : 67
7. A, B : 55
8. D, E : 53
9. B, E : 52
10. E, A : 51

On construit un graphe petit à petit en insérant es sommets par ordre de préférence et en évitant les cycles.

$$\begin{tikzcd}
& C \arrow[d] \arrow[dr] & \\
& D \arrow[dr] & B \arrow[l] \\
E \arrow[uur] \arrow[rr] & & A
\end{tikzcd}$$

On pose

$$T_0 = \left\{ (c_1, c_2, s) \;\middle|\; c_1, c_2 \in \mathcal{C},\ s = \mathrm{comp}(c_1, c_2) = \sum_{v \in V} \mathbf{1}_{f(v_1,c_1) > f(v_1,c_2)} \right\}.$$

1. Dans T_0 on ne regarde que les triplets tels que $\mathrm{comp}(c_1, c_2) > \mathrm{comp}(c_2, c_1)$.
2. On trie T par ordre décroissant selon la valeur de s. Soit S la suite ordonnée obtenue.
3. On construit G de sommets $V = \mathcal{C}$ et d'ensemble d'arêtes $E \subset \{(c_j, d_j) \mid (c_j, d_j, s) \in T\}$ avec E construit par :

```
E = []
while len(S)!=0:
        e = [S[0][0],S[0][1]]
        if estAcyclique(G=(C,E.append(e))):
                E.append(e)
                S = S[1:]
```

Lemme 2.10
Soit $G = (V, E)$ un graphe orienté fini tel que

$$\forall x, y \in V,\ (x, y) \in E \text{ ou } (y, x) \in E$$

et tel que G n'a pas de cycle.
Alors G a une source unique.

Démonstration
Comme G n'a pas de cycle, G a une source.
Soient $x \neq y$ deux sources. Comme G est un tournoi (vérifie la première propriété) alors soit $(x, y) \in E$ soit $(y, x) \in E$ donc ne peut être une source pour l'un des deux.

Corollaire 2.11
La méthode de Tideman produit un et un seul vainqueur.