

# Tests, limites

## Tests unitaires

Pour écrire des tests unitaires le framework `unittest` a été utilisé. Tous les tests se trouvent dans le dossier `tests`. Pour automatiser le lancement des tests, le `makefile` est disponible. Pour lancer tous les tests en étant dans le répertoire principale du projet (celui avec `main.py`) il suffit d'exécuter la commande `make -C tests all`.

Les tests unitaires vérifient la bonne fonctionnement des algorithmes suivants :

1. Toutes les règles de vote utilisées dans l'application. Les fichiers associés sont `test_approval.py`, `test_borda.py`, `test_exhaustive_ballot.py`, `test_plurality.py`, `test_veto.py`.
2. Tie-break par duels. Les fichiers associés sont `test_borda.py`, `test_veto.py`.
3. Calcul de classement des candidats en fonction de la position d'électeur. Le fichier associé est `test_ranking.py`.

## Limites physiques

On introduit la notation suivante :

- $N$  un nombre d'électeurs participant dans l'élection
- $C$  un nombre de candidats participant dans l'élection

Ainsi, on présente aussi la complexité des algorithmes des règles de vote pour l'analyse (en sachant que la complexité d'algorithme de tri intégré en Python de la liste à  $n$  éléments est de  $O(n * \log(n))$ ) :

- Approbation  $\rightarrow O(C * N + C * \log C)$
- Pluralité à un tour  $\rightarrow O(N + C * \log C)$
- Pluralité à 2 tours  $\rightarrow O(2 * (N + C * \log C))$
- Éliminations successives  $\rightarrow O((C - 1) * (N + C * \log C))$
- Veto  $\rightarrow O(C * N + C * \log C)$
- Borda  $\rightarrow O(C * N + C * \log C)$

Pour les règles de vote Condorcet-cohérentes, on définit aussi la taille de la dictionnaire qui contient les duels  $D = \binom{C}{2} = \frac{C!}{2 * (C - 2)!}$ .

- Condorcet simple, Copeland, Simson  $\rightarrow O(D + C * \log(C))$

## Temps de calcul

Pour chaque règle de vote on doit traiter tous les électeurs. Par conséquent, le temps de calcul dépend largement de nombre d'électeurs participant dans l'élection.

En sachant la complexité des algorithmes des règles de vote présentées ci-dessus, au delà de 100 000 électeurs, le temps de calcul peut commencer à prendre une dizaine de secondes (particulièrement si on sélectionne plusieurs règles de vote). Au delà du million électeurs, il est

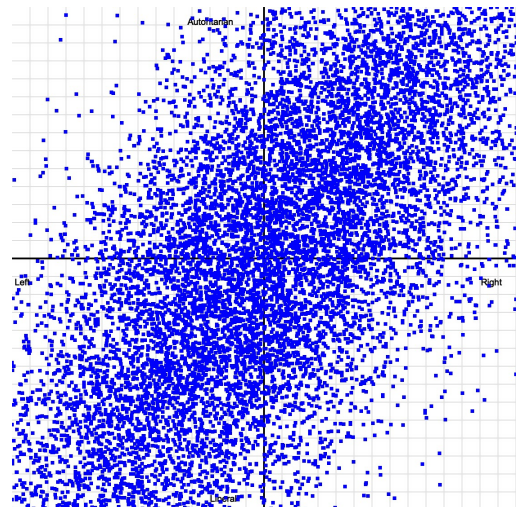
conseillé de ne sélectionner qu'une seule règle de vote. Dans ce cas le temps de calcul se rapprochera quand même de la minute.

*Attention Windows à tendance à indiquer que le logiciel ne répond plus pendant les calculs plus long (e.g. plus long que 5 secondes).*

Ainsi, si le nombre d'électeurs devient trop grand, l'application peut crasher. Par exemple, sur Windows avec plusieurs millions des électeurs, l'application peut crasher.

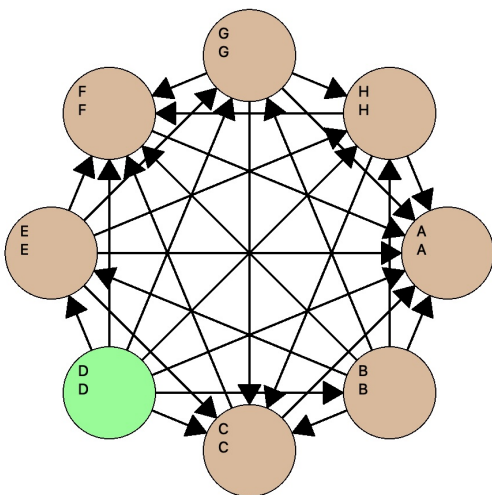
## Visualisation

Ici, on vous présente la carte politique avec 10 000 électeurs générés aléatoirement selon les paramètres par défaut de la loi normale. On voit qu'elle devient plus au moins remplie. Par conséquent, à mesure que le nombre d'électeurs augmente, la carte politique tend à se remplir davantage.

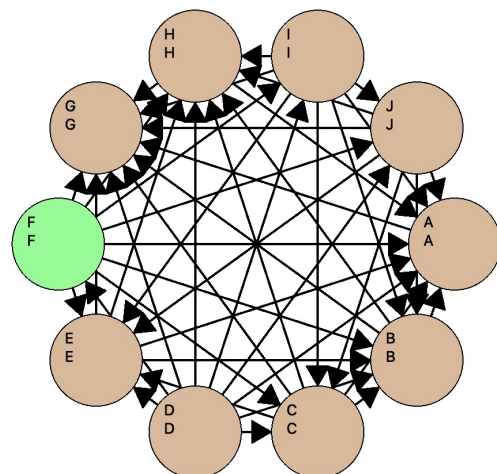


Carte politique avec 10 000 électeurs

En ce qui concerne la visualisation des résultats, la limite concerne principalement des graphes de Condorcet. Ci-dessous, on présente deux graphes de Condorcet avec 8 et 10 candidats. On observe qu'au-delà de 10 candidats, les graphes deviennent peu lisibles. Même si la fenêtre est agrandie, les graphes restent inchangés en raison de leur position fixe. Pourtant, on pense que c'est possible d'implémenter un ajustement automatique de la position des figures en fonction de la taille de la fenêtre, ce qui pourrait être l'une des améliorations potentielles de l'application.



Graphe de Condorcet avec 8 candidats



Graphe de Condorcet avec 8 candidats

Pour les règles de vote à plusieurs tours, la limite est d'environ 25 candidats. Ci-dessous, un diagramme illustre les éliminations successives pour différents nombres de candidats. Les résultats complets ne sont visibles que si la fenêtre est agrandie pour remplir tout l'écran.

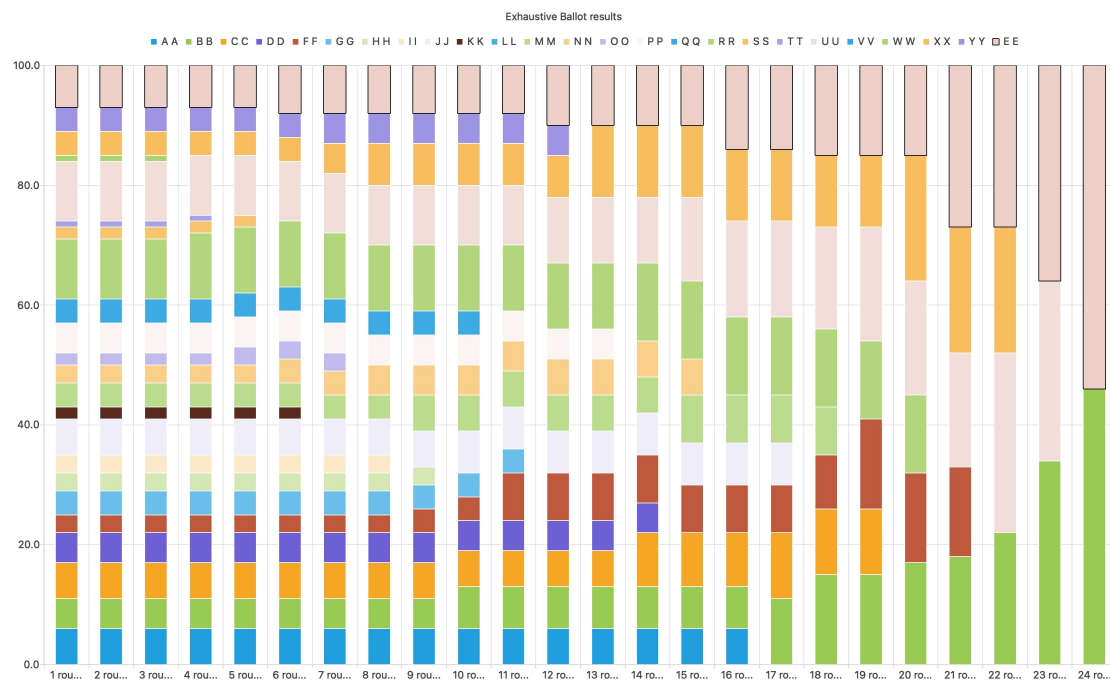


Diagramme pour la règle de vote à un tour avec 25 candidats.

Pour les règles de vote à un tour, la limite n'est pas nécessairement présente si la fenêtre est agrandie, mais les barres de vote deviendront de plus en plus fines si le nombre des candidats augmente.

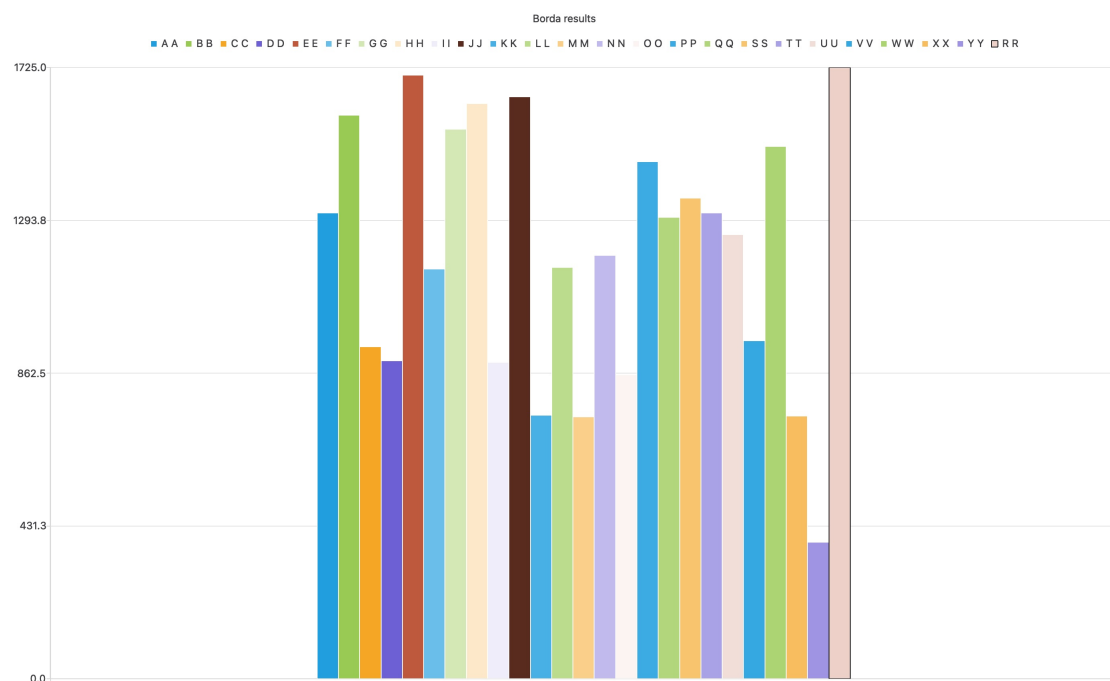


Diagramme pour la règle de vote à un tour avec 25 candidats.

## Démocratie liquide

L'algorithme de la démocratie liquide dépend d'un nombre des électeurs participant dans l'élection. La complexité de l'algorithme est de  $O(N^2)$ . Par conséquent, au delà de 1000 électeurs, les performances sont largement amoindries et il est donc déconseillé de choisir plusieurs règles de vote quand la démocratie liquide est activée.

## Sondages

On a fixé une limite maximale de 12 sondages, en considérant qu'il est possible d'effectuer un sondage chaque mois pendant une année. Les limites de la visualisation et du temps du calcul sont aussi présents.

## Création des candidats

Chaque candidat est identifié par son nom et prénom, et cette paire doit être unique. La comparaison des candidats se fait exclusivement à partir de ces informations. Bien que l'application n'empêche pas la création de deux candidats ayant le même nom et prénom, cela peut conduire à des résultats incohérents.