

Projet APO

Modélisation de systèmes électoraux et dynamiques de vote

Mathieu Lefort

10 & 13 décembre 2021

1 Modalités pratiques

Ce TP est à faire par groupe de 2 et est à rendre pour le 23 janvier 2022.

2 Évaluation

Vous serez évalués sur les points suivants :

- Méthodologie (1 point)
- Conception (8 points)
- Code (8 points)
- Documentation (3 points)
- Extensions optionnelles (? points suivant vos choix)

2.1 Méthodologie

Vous expliquerez votre méthodologie de travail (articulation entre conception et codage, déroulé temporel du projet, ...) et la répartition des tâches entre les différents membres du groupe.

2.2 Conception

Vous utiliserez les différents types de diagrammes vus en TDs à bon escient (i.e. permettant à quelqu'un d'extérieur au projet de le comprendre et de pouvoir le coder). Pour rappel, pour le diagramme de cas d'utilisations, chaque cas d'utilisation doit être décrit par un texte.

Votre conception devra être modulaire et donc permettre une intégration sans trop d'efforts des différentes extensions possibles. Il ne vous est pas demandé de modéliser les extensions non faites, mais d'intégrer leur éventuelle insertion dans votre réflexion lors de votre conception.

De plus, les différents choix (majeurs) de votre conception devront être justifiés.

2.3 Code

Le code devra être clair (indentation, noms de variables pertinentes, ...), utiliser la généricité et être autant que possible efficace. Il devra également et surtout être en accord avec la conception. Penser à bien vérifier et traiter (avec des exceptions) les cas d'erreurs dans vos méthodes.

Pensez à utiliser au maximum les packages pour structurer vos fichiers/classes.

2.4 Documentation

Vous générerez la javadoc de votre code (documentation des classes, méthodes, paramètres, ...). Vous penserez également à commenter votre code là où vous le jugez utile.

2.5 Extensions

Plusieurs extensions optionnelles vous sont proposées. À vous de voir celles qui peuvent vous intéresser. Il est tout à fait possible également de proposer (et d'implémenter) vos propres extensions. Si vous décidez de faire des extensions, votre note maximale potentielle peut dépasser 20. En pratique votre note sera plafonnée à 20 pour le projet, les points supplémentaires seront transférés à votre note de CC avec un barème à définir. L'idée des extensions est surtout une opportunité de pousser l'exploration de Java plus loin si vous le souhaitez.

3 Sujet

L'objectif du projet est d'analyser différents systèmes de votes et certaines dynamiques "politiques" avec des modèles très simplifiés.

3.1 Électeurs et candidats

Les électeurs et les candidats vont être représentés par leurs préférences sur différents axes "sociétaux" ¹ normés entre 0 et 1. Par exemple, on peut considérer 2 axes : "pouvoir d'achat" et "écologie". Un électeur aura, par exemple, une représentation $[0.8, 0.1]$ si il est beaucoup intéressé par son pouvoir d'achat mais peu par l'écologie. De manière similaire, un candidat avec une représentation $[0.7, 0.9]$ aura un programme ayant beaucoup de mesures concernant le pouvoir d'achat et l'écologie. Avec des telles représentations, l'appétance d'un électeur pour un candidat pourra se mesurer par une distance entre leurs vecteurs respectifs. On pourra modéliser l'absention d'un électeur par le fait qu'aucun candidat n'est suffisamment proche de ses préférences.

3.2 Mode de scrutin

Vous implémenterez les modes de scrutin suivants ² en pensant à gérer correctement les cas d'égalité :

- scrutin majoritaire à un tour : chaque électeur vote pour un seul candidat. Le candidat qui a récolté le plus de voix est élu.
- scrutin majoritaire à deux tours : chaque électeur vote pour un seul candidat, les candidats remplissant un certain critère peuvent accéder au second tour (qui correspond à un scrutin majoritaire à un tour). Le critère peut être d'être classé parmi les n premiers ou d'avoir reçu un certain pourcentage de voix.
- vote par approbation : chaque électeur peut voter pour le nombre de candidats qu'il souhaite, le vainqueur est celui qui a récolté le plus de voix.
- vote alternatif : chaque électeur classe l'ensemble des candidats. Celui qui a reçu le moins de premier choix est éliminé et ses votes sont transférés aux candidats restants (via le 2ème choix des bulletins) et ainsi de suite jusqu'à ce qu'il ne reste qu'un seul candidat qui est élu.
- méthode de Borda : chaque électeur classe les candidats qu'il souhaite, le premier reçoit n points, le deuxième $n-1$ et ainsi de suite. Le candidat qui a reçu le plus de points est élu.

Pour information, le théorème d'impossibilité d'Arrow stipule qu'un scrutin dans lequel les électeurs expriment leur opinion de manière qualitative (i.e. en indiquant des préférences relatives entre candidats) ne peut satisfaire un ensemble de propriétés socialement souhaitables (universalité, non dictature, unanimité et indépendance des options non pertinentes). Il n'y a donc pas de système de vote parfait d'où leur diversité selon les élections et pays.

3.3 Dynamique

On peut vouloir modéliser certaines interactions entre les citoyens qui vont faire évoluer au cours du temps leurs préférences (et donc leurs votes). Dans ce projet on s'intéressera aux deux types suivants.

3.3.1 Interactions socio-politiques

Pendant un certain nombre de pas de temps on va modifier les préférences de chacun des électeurs par interaction avec une autre personne (électeur ou candidat) choisie aléatoirement ³. Si la personne choisie est proche (fixé par un seuil) des préférences de l'électeur concerné, alors ce dernier va légèrement modifier ses préférences vers celles de la personne. Dans le cas contraire si la personne est trop éloignée (fixé par un seuil), alors l'électeur va éloigner ses préférences de celles de la personne.

3.3.2 Sondages

Pendant un certain nombre de pas de temps, on va modifier les préférences de chacun des électeurs en fonction du résultat d'un sondage ⁴. Plusieurs options sont possibles :

- l'électeur va se déplacer vers le candidat le plus proche de ses préférences parmi les n premiers candidats du sondage
- l'électeur va calculer une utilité pour chaque candidat en multipliant sa préférence (inverse de la distance au candidat) par son pourcentage de voix dans le sondage. L'électeur va alors se déplacer vers le candidat ayant l'utilité la plus élevée.
- comme dans la version précédente, une utilité va être calculée pour chaque candidat. Cependant l'électeur va se déplacer vers chaque candidat de manière proportionnelle à son utilité.

1. Pour information il s'agit d'une représentation classique pour ce genre de modèle. Si vous êtes curieux vous pouvez aller voir Enelow J.M., Hinich M.J., The spatial theory of voting, an introduction, Cambridge University Press, 1984

2. Vous pouvez trouver plus de détails et des exemples sur la page wikipédia.

3. Vous pouvez voir l'illustration de quelques équations et dynamiques possibles ici.

4. Vous pouvez voir l'illustration de quelques équations et dynamiques possibles ici.

4 Tronc commun

Vous devez implémenter les différents aspects décrits dans les sections précédentes. En particulier votre programme permettra :

- d’initialiser un certain nombre d’électeurs et de candidats à des valeurs fournies ou aléatoires⁵
- de simuler une élection (le mode de scrutin pourra être choisi) et d’afficher le résultat (de manière textuelle)
- de réaliser un sondage d’opinion (i.e. le résultat d’une élection en prenant aléatoirement un sous ensemble d’électeurs) et d’afficher le résultat (de manière textuelle)
- de pouvoir faire évoluer les opinions des électeurs préalablement à l’élection, possiblement en affichant (textuellement) l’évolution des opinions via sondage

Le choix entre ces différentes fonctionnalités devra être accessible via un petit menu textuel.

5 Extensions possibles

5.1 Environnement de travail (0.5 point)

Vous utiliserez un git (dont l’adresse sera à fournir dans le rapport) pour faire le suivi de versions tout au long de votre projet.

5.2 Installation (0.5 point)

Vous fournirez un fichier build.xml (regardez <https://ant.apache.org/>) permettant de compiler votre code (à partir d’un dossier src), de générer la javadoc (dans un dossier doc) et de supprimer l’installation (fichiers compilés et documentation). Si vous faites également une extension demandant l’utilisation d’un framework hors JRE, le fichier devra également s’occuper de l’installation du framework.

5.3 Tests unitaires (2 points)

Vous implémenterez des tests unitaires des méthodes (principales) de votre programme (regardez <https://junit.org/junit5/>).

5.4 Fichier de configuration (1 point)

Faites en sorte que tous les paramètres de la simulation soient stockés dans un fichier de configuration (humainement lisible et éditable). Ce fichier sera passé en paramètre de votre programme principal.

5.5 Sauvegarde (2 points)

Sauvegardez dans un fichier de type Excel les résultats des simulations (électeurs, candidats, sondages, élections) afin de pouvoir faire des graphiques d’analyse a posteriori.

5.6 Interface graphique (4 points)

En plus d’une interface textuelle, vous proposerez une interface graphique pour la visualisation des résultats. Vous respecterez au mieux le *design pattern* MVC (Modèle Vue Contrôleur), à savoir a minima avoir les classes d’affichage séparées de celles du modèle. Regardez par exemple les bibliothèques d’affichage *PtPlot* ou *JFreeChart*.

5.7 Analyse des modes de scrutin (0.5+1 point)

On veut pouvoir étudier l’influence du mode de scrutin sur le résultat d’une élection. Pour cela on va générer un ensemble de populations dont le centre de gravité (i.e. le barycentre des vecteurs des électeurs) sera donné et ensuite calculer le résultat de ces élections (en moyenne) suivant les différents modes de scrutins.

En cas d’interface graphique, on pourra afficher le résultat pour tous les centres de gravité possibles dans le cas de vecteurs d’opinions en 2 dimensions, i.e une carte de résultat comme montré ici par exemple.

5.8 Spatialisation (0.5 point)

On peut vouloir que les interactions entre électeurs ne se fassent que dans des cercles sociaux (et non pas chaque électeur pouvant interagir avec tout le monde). On va alors donner une position (géographique) à chaque électeur et n’autoriser son influence qu’à des électeurs proches géographiquement.

5. Vous pouvez par exemple essayer de correspondre aux données des sondages des élections présidentielles de 2017 ou 2022.