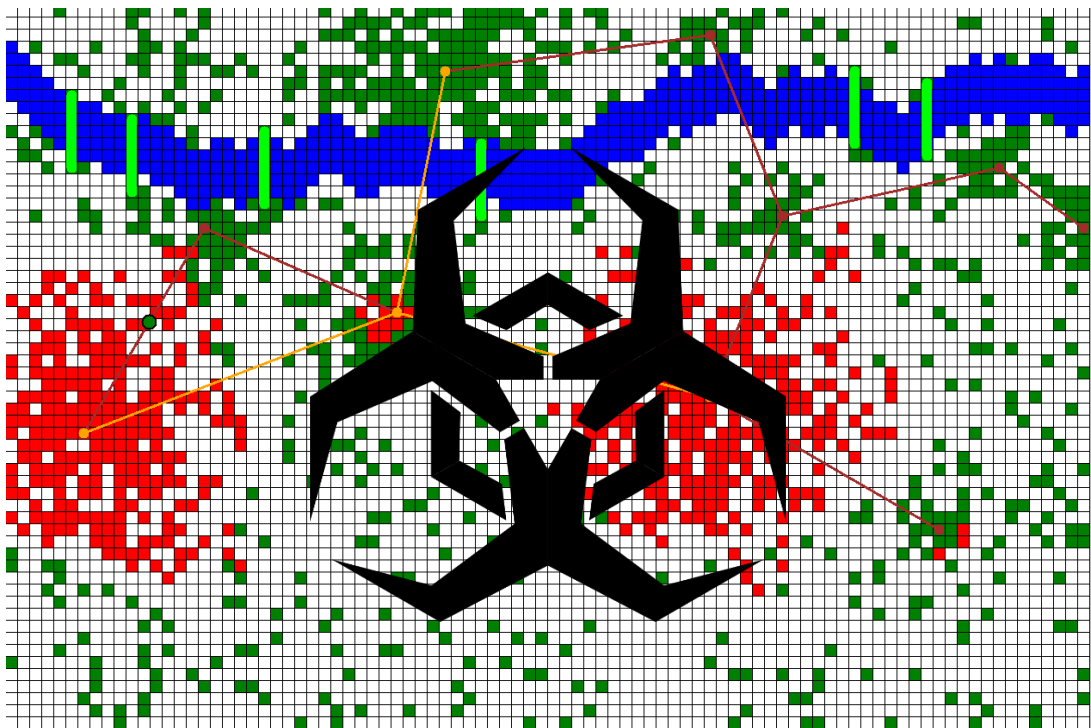


Modélisation mathématique

Module 3202C

Automates cellulaires



Nicolas MEYNIEL, Damien PIEDANNA, Axel PISANI, Aurélien ROBINEAU

Groupe 3B – 2nd année

Module M3202C – Modélisation mathématique

Enseignant responsable : Patricia GAITAN

Mercredi 7 Novembre 2018

Table des matières

Choix du sujet.....	4
Les automates cellulaires	4
Évolution du projet.....	5
Etat final	6
Fonctionnement général du programme.....	7
Génération de la grille	7
Génération des zones urbaines.....	7
Génération des cellules	7
Génération du fleuve	8
Génération des déplacements	8
Propagation du virus	8
Contacts entre cellules voisines	8
Voyages	9
Incidence proportionnelle.....	10
Matrices.....	11
Génération aléatoire réaliste	11
Algorithme de Prim	11
Densité des zones urbaines.....	12
Conclusion	12

Choix du sujet

Lorsque les premiers sujets nous ont été proposés nous étions partis sur la réalisation de formes géométriques en 3D et en mouvement mais cela nous paraissait trop classique, nous voulions réaliser une représentation d'un fait réel tel qu'un feu ou des mouvements de populations. Là nous est venue l'idée d'utiliser les automates cellulaires et de représenter la propagation d'un virus dans une population avec des facteurs définis. Petit à petit les idées nous sont venues afin d'améliorer et rendre le plus réaliste possible la simulation.

Les automates cellulaires

Les automates cellulaires sont utilisés pour modéliser des systèmes complexes. Ils consistent en une grille régulière de « cellules » contenant chacune un « état », choisi parmi un ensemble fini et qui peut évoluer au cours du temps. L'état d'une cellule au temps $t+1$ est fonction de l'état au temps t d'un nombre fini de cellules appelé son « voisinage ». À chaque nouvelle unité de temps, les mêmes règles sont appliquées simultanément à toutes les cellules de la grille, produisant une nouvelle « génération » de cellules dépendant entièrement de la génération précédente. Les exemples sont nombreux : trafic routier, réseaux informatiques, formation d'essaims d'oiseaux, occupation de nids d'abeille, tissus cellulaires...

Du côté informatique, les automates cellulaires peuvent également être vus comme un modèle abstrait représentant le fonctionnement des ordinateurs (précisément, ils constituent un modèle de calcul ayant la puissance des machines de Turing).

Évolution du projet

Au début, nous n'avions qu'une simple grille peu fonctionnelle avec des cellules réparties aléatoirement et de couleurs différentes représentant des populations saines(vertes) ou infecte(rouge).

Ensuite, diverses fonctionnalités ont été ajoutées tel que pouvoir choisir quelles cellules de la grille allaient être infectées par le virus ou non grâce aux clics souris, et pouvoir choisir le nombre de cellules de la grille en hauteur et en largeur.

Par la suite nous avons implémenté la fonction qui permet de propager le virus à travers la population à partir d'un point donné. Ce point de départ détecte les cellules de populations autour de lui et avec des probabilités définies par nous-même, déterminait si le virus allait se propager ou non.

Le projet évoluant nous avons rajouté des zones peuplées, des villages, villes et métropoles représentés graphiquement par des zones où le nombre de cellules de populations y est plus dense. Au centre d'une ville la population sera donc plus dense qu'à l'extérieur et cela influencera la propagation du virus. Des fleuves ont aussi été rajoutés avec des ponts qui permettent le passage d'un côté à l'autre grâce à des ponts.

La fonction de propagation du virus a été améliorée avec une fonction qui s'appelle l'incidence proportionnelle, elle comprend deux composantes :

- Un décompte du nombre de contacts ayant lieu à l'instant T ,
- Et une description de la probabilité qu'un tel contact, lorsqu'il a lieu, se traduise par la transmission de la maladie.

Les cellules de populations ont été perfectionnées en y ajoutant un âge moyen variant de 0 à 105 ans qui permettra d'interagir avec la fonction de propagation afin de lui donner plus de réalisme.

Pour rendre notre simulation encore plus réaliste nous avons ajouté des déplacements tels que les routes, les trains ou encore l'avion (qui n'apparaît qu'entre deux métropoles) qui permettent aux populations de se déplacer entre différentes zones urbaines et ainsi d'y répandre le virus, mais ce n'est pas tout, nous avons aussi ajouté un menu permettant à l'utilisateur de définir la taille souhaitée de la simulation et de choisir le virus qui se propagera sachant que le virus interagit différemment avec les différentes classes d'âge des populations.

Etat final

Au final, la simulation comporte tous les éléments suivant :

- Différents types de cellules:
 - o les cellules saines (verte)
 - o les cellules infectées (rouge)
 - o les cellules vide (blanche)
 - o les cellules guéries (verte claire)
 - o les cellules mortes (grises)
 - o les cellules d'eau (bleu)
- Différentes zones urbaines créées aléatoirement:
 - o des zones peuplées
 - o des villages
 - o des villes
 - o des métropoles
- Des moyens de déplacements :
 - o des routes (marron) entre toutes les zones urbaines sauf les zones peuplées
 - o des lignes de train (orange) entre les villes et la métropole
 - o des voies aériennes (rose) entre deux métropoles
 - o des ponts pour piétons (vert fluo)
- Un menu au lancement de la simulation permettant de choisir la taille en hauteur et largeur de la simulation ainsi que le virus qui se propagera (sachant que certains sont plus ou moins létal).
- Les trois clics souris on chacun une action dédiée :
 - o le clic gauche infecte la cellule cible
 - o le clic droit transforme la cellule cible en cellule saine si elle est infectée
 - o le clic molette permet d'avancer d'une étape lorsque la simulation est en pause
- Une fonction de propagation qui permet au virus (précédemment choisis par l'utilisateur) de se répandre. Pour déterminer si une cellule va en infecter une autre on utilise l'incidence proportionnelle qui à partir du nombre de cellules saines proches, du nombre d'infecté proche et des propriétés du virus détermine si au prochain tour des cellules seront infecté ou non. Ensuite il y a une probabilité propre à chaque virus d'en mourir ou d'en guérir au bout d'un certain temps.

Fonctionnement général du programme

Lorsque l'utilisateur lance le programme, cela crée une fenêtre qui lui permet d'entrer les conditions de départ de la simulation. Il s'agit de la taille de la matrice (plus elle est grande, plus la simulation est réaliste, mais plus le temps d'exécution est long), et du virus à propager (choisi parmi une liste prédéfinie).

Génération de la grille

On commence par générer la grille sur laquelle aura lieu la simulation.

Génération des zones urbaines

Les cellules étant organisées en zone urbaines, il faut d'abord définir ces zones avant de générer les cellules.

La fonction *genererZonesUrbaines()* génère aléatoirement des zones sur la grille : entre 0 et 2 métropoles, entre 2 et 4 villes, à 5 et 10 villages, entre 0 et 2 zones peuplées.

Pour créer une nouvelle zone, l'algorithme procède ainsi :

- Choisi aléatoirement la position du centre de la zone sur la grille
- Selon le type de zone, choisi un rayon aléatoirement entre 2 bornes
- Test si la zone ne chevauche pas une zone déjà existante. Si oui, tente de recréer une zone à un autre endroit. Abandonne au bout de 10 essais.

Génération des cellules

Pour générer les cellules on parcourt la matrice contenant les cellules. Pour chaque position, on teste si la cellule est dans une zone urbaine. Si ce n'est pas le cas, on considère qu'on se trouve dans une campagne, la cellule a donc une chance sur huit d'être saine, sinon elle est vide. Si on se trouve dans une zone urbaine, la probabilité qu'une cellule soit saine ou vide dépend de sa distance au centre de la zone (fonction de probabilité expliquée plus bas).

A chaque création d'une cellule saine, celle-ci se voit attribuer un âge moyen. On tire aléatoirement n âges aléatoires (n étant le nombre de personne par cellule) entre 0 et 100 et on calcule la moyenne. Les âges sont tirés selon leur proportion en France en 2018.

La grille est donc composée pour le moment de cellule saines et vides.

Génération du fleuve

C'est ensuite le fleuve qui est généré.

Le fleuve démarre d'une position tirée aléatoirement en haut de la grille (selon l'axe des x) ou à gauche (selon l'axe des y). D'abord, le tracé du fleuve est généré. A partir de la position de départ, on sélectionne la cellule suivante en avançant d'un sur l'axe des y si le fleuve part du haut, ou l'axe des x si le fleuve part de la gauche. Sur l'autre axe, on se décale d'une cellule à gauche ou à droite (aléatoirement afin que le fleuve ne soit pas rectiligne).

Une fois le tracé généré, on étend le fleuve de chaque côté des cellules du tracé de la moitié de la largeur du fleuve (aussi tiré aléatoirement dans un intervalle défini).

Génération des déplacements

Les déplacements sont générés avec l'algorithme de Prim. A partir du graphe complet dont les sommets sont le centre de toutes les zones urbaines concernées par le type de déplacement généré, il génère les déplacements (arêtes du graphe) reliant tous les sommets, tel que le poids du graphe soit minimal (le poids de chaque arête est la distance en nombre de cellules entre deux centres de zones).

Propagation du virus

Il faut ensuite propager le virus. Pour cela on applique à chaque cellule par unité de temps la fonction qui détermine l'état que doit avoir la cellule au prochain tour de boucle (unité de temps).

Contacts entre cellules voisines

On commence par parcourir la matrice de cellules. Pour chaque cellule saine, on détermine le taux de contact avec le virus (calcul expliqué dans la partie « Mathématiques »).

On utilise ensuite la fonction d'incidence proportionnelle (expliquée dans la section « Mathématiques » qui donne la probabilité qu'une cellule soit infectée au contact du virus (donc en fonction du taux de contact calculé précédemment). Cette probabilité est multipliée par le taux de sensibilité d'une cellule au virus en fonction de son âge moyen.

Pour chaque cellule infectée, les possibilités d'évolution sont :

- La cellule reste infectée
- La cellule meure
- La cellule guérie

Chaque cellule infectée possède un compteur déterminant depuis combien de jour elle est infectée.

Si ce nombre de jour est situé dans l'intervalle de durée du virus en cours, la cellule a une probabilité de changer d'état (cette probabilité vaut 1 si le nombre de jour dépasse l'intervalle). Une probabilité est appliquée à la cellule pour déterminer si elle meure ou guérie. Cette probabilité est une propriété de la maladie appelée taux de létalité.

Voyages

A chaque itération, des voyages sont déclenchés aléatoirement sur les déplacements de la grille. L'état du voyageur est défini en fonction de l'état des cellules de la zone de départ.

A chaque changement d'état d'une cellule le nombre de cellules infectées, saines, guéries et mortes est mis à jour. Cela permet d'avoir un chaque instant le pourcentage de cellules dans un état précis au sein de la zone. Ce pourcentage est la probabilité qu'un voyageur de cet état parte de la zone. Si le voyageur est dans l'état « mort », le voyage n'a pas lieu.

Lorsque le voyageur arrive dans la zone d'arrivée, la cellule d'arrivée est infectée pendant un tour seulement, ce qui peut permettre au virus de se transmettre aux cellules voisines. Pour cela, l'état d'origine de la cellule est enregistré, et restitué au tour suivant.

Les mathématiques dans le code

Incidence proportionnelle

En épidémiologie, l'incidence décrit, avec la prévalence, l'importance d'une maladie dans une population et signifie chaque nouveau cas d'une maladie par unité de temps dans une population définie. On suppose que les individus composant la population peuvent être dans l'un de deux états :

- susceptibles à la maladie s'ils n'ont pas encore contracté la maladie
- infectieux s'ils ont contracté la maladie et qu'ils la propagent.

Soit :

- $S(t)$ le nombre de susceptibles (au temps t)
- $I(t)$ le nombre d'infectieux (au temps t)
- β est le coefficient de transmission (correspond au taux de reproduction de base R_0 d'une maladie infectieuse).

Quand un contact a lieu entre un individu infectieux et un individu susceptible, la maladie peut se transmettre.

La fonction $f(S, I)$ qui décrit ce processus s'appelle l'incidence. Elle consiste en deux composantes :

- un décompte du nombre de contacts ayant lieu,
- une description de la probabilité qu'un tel contact, lorsqu'il a lieu, se traduise par la transmission de la maladie.

On obtient la formule suivante :

$$F(S, I) = \beta (SI / S+I)$$

On rajoute la moyenne d'âge qui permet au virus de se propager différemment en fonction de l'âge moyen d'une cellule.

Ce qui nous donne cela dans le programme :

$$r0 * ((NbSains * tauxContact) / (NbSains + tauxContact)) * ageMoyen$$

```

if (cptSain+tauxContact > 0):
    tauxInfection = self.virus.tauxReproduction*((cptSain*tauxContact)/(cptSain+tauxContact)) * tauxAge
  
```

Pour chaque cellule saine, on détermine le taux de contact avec le virus dans un rayon de 2 cellules. Chaque cellule comprise dans ce rayon ajoute une valeur spécifique au taux de contact avec le virus (voir schéma ci-contre).

0.25	0.5	0.5	0.5	0.25
0.5	0.75	1	0.75	0.5
0.5	1		1	0.5
0.5	0.75	1	0.75	0.5
0.25	0.5	0.5	0.5	0.25

Matrices

Un automate cellulaire est une grille constituée de cellule autonome ayant un état et un comportement, l'application logique à cela est de créer une « matrice de cellule » ce donc ce que nous avons fait. Cette matrice est ensuite parcourue et chaque cellule en fonction de son état et de sa position va subir une transformation (dans notre cas une cellule peut se faire infecter, guérir ou mourir). (Voir incidence proportionnelle)

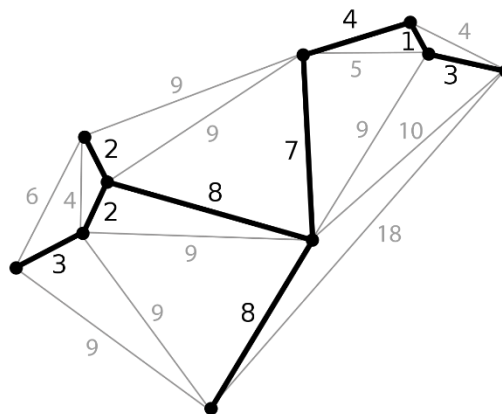
Génération aléatoire réaliste

Chaque génération de grille dans notre modèle est aléatoire mais restreint par des conditions. Ainsi la génération des villes et fleuves n'est jamais identique d'une simulation à l'autre tout en étant réaliste, cela permet d'obtenir une quasi infinité de modélisation possible et de tenir des conclusions en fonction de l'organisation du terrain.

Algorithme de Prim

L'algorithme de Prim est un algorithme dit glouton, qui calcule un arbre couvrant minimal d'un graphe connexe valué et non orienté. En d'autres termes, cet algorithme trouve un sous-ensemble d'arêtes formant un arbre sur l'ensemble des sommets du graphe initial, tel que la somme des poids de ces arêtes soit minimale.

Dans notre projet, chaque sommet est représenté par le centre des villages, villes et métropoles. Afin de créer des liaisons, ici représentés pour les axes routiers, ferroviaires et aériens, nous avons implémenté l'algorithme de Prim et utilisé pour chaque axe. Le poids minimal est la distance entre chaque ville afin d'obtenir un rendu plus réaliste approchant un trafic réel.



Densité des zones urbaines

La population de la zone urbaine (hors Zone Peuplée) doit être de moins en moins dense plus l'on approche du bord de la zone. On veut qu'au centre, la population soit très dense. Il aura donc 100% de chance que la cellule au centre de la zone soit peuplée. On veut qu'en bordure de zone une cellule ai 25% de chance d'être peuplée. Si on représente cela par une fonction linéaire on a :

$f(x) = ax + b$ (pourcentage de chance d'être peuplée en fonction de la distance de la cellule au centre de la zone)

$$f(0) = 100$$

$$f(\text{rayonZone}) = 25$$

donc

$$f(0) = a \cdot 0 + b = 100$$

$$\Leftrightarrow b = 100$$

donc

$$f(\text{rayonZone}) = a \cdot \text{rayonZone} + 100 = 25$$

$$\Leftrightarrow a \cdot \text{rayonZone} = -75$$

$$\Leftrightarrow a = -75 / \text{rayonZone}$$

donc

$$f(x) = (-75 / \text{rayonZone})x + 100$$

```
probaPop = (-75 / zoneUrbaine.rayon) * zoneUrbaine.distanceAuCentre(Position(x,y)) + 100
```

Conclusion

L'idée du projet reposait d'abord sur des calculs de probabilité, permettant de simuler avec réalisme la propagation d'un virus au sein d'une population. Cependant, après les recherches effectuées sur ces probabilités, il s'est avéré qu'elles étaient bien trop complexes pour notre niveau. Nous avons donc dû revoir à la baisse nos objectifs, en simplifiant beaucoup les probabilités d'infection.

Les simulations effectuées à partir de notre programme ne peuvent pas permettre de simuler avec réalisme la propagation de virus, mais permet tout de même de visualiser des différences d'évolution entre les épidémies des différents virus.

Avec plus de temps et de connaissances, notre projet aurait donc pu évoluer en une simulation réaliste permettant de réduire la propagation de virus dans le monde.