

RÉSEAUX D'INTERACTION

TP2 - ÉTUDE D'UN ARTICLE ET RÉALISATION DE MODÈLES

Simon Camille Master 2 IWOCS

16/02/2018

Ce travail présente les modèles NetLogo créés à partir de l'article *Mixing beliefs among interacting agents* de G. Deffuant, D. Neau, F. Amblard et G. Weisbuch¹.

1 Modèle *Mélange complet*

1.1 Principe du modèle

Dans ce premier modèle, on considère une population de N agents. Chaque agent i dispose d'une opinion x_i avec $x_i \in [0, 1]$. À chaque itération du modèle, on sélectionne aléatoirement deux agents. Les deux agents s'accordent sur une opinion commune si la différence de leurs opinions est plus petite qu'un certain seuil d . Posons x et x' les opinions des deux agents. Si $|x - x'| < d$, alors les deux agents accordent leurs opinions selon la règle suivante :

$$\begin{aligned}x &= x + \mu \cdot (x' - x) \\x' &= x' + \mu \cdot (x - x')\end{aligned}$$

La variable μ est le paramètre de convergence, lors des simulations on le fait varier entre 0 et 0,5.

Avec ce premier modèle, l'article présente deux jeux de tests ainsi que leurs représentations graphiques :

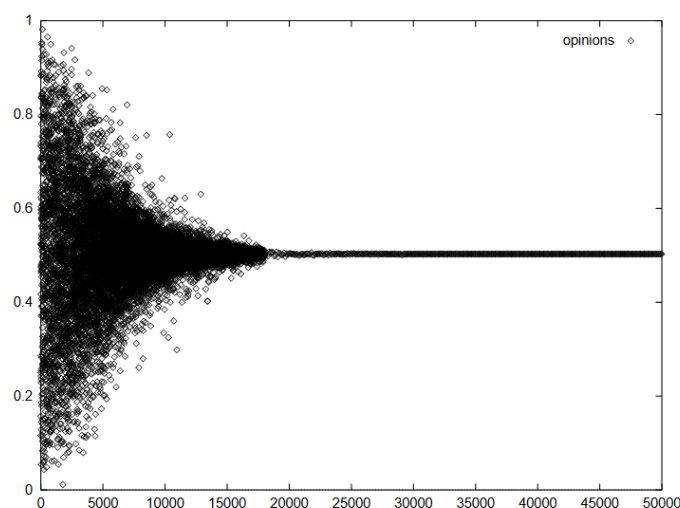


Figure 1: Évolution de l'opinion dans le temps. Résultat obtenu avec les paramètres : $d = 0,5$, $\mu = 0,5$, $N = 1000$.

¹Article consultable à l'adresse : https://eureka.univ-lehavre.fr/file.php/1641/Cours_2018/TP2/mixbel_Deffuant.pdf.

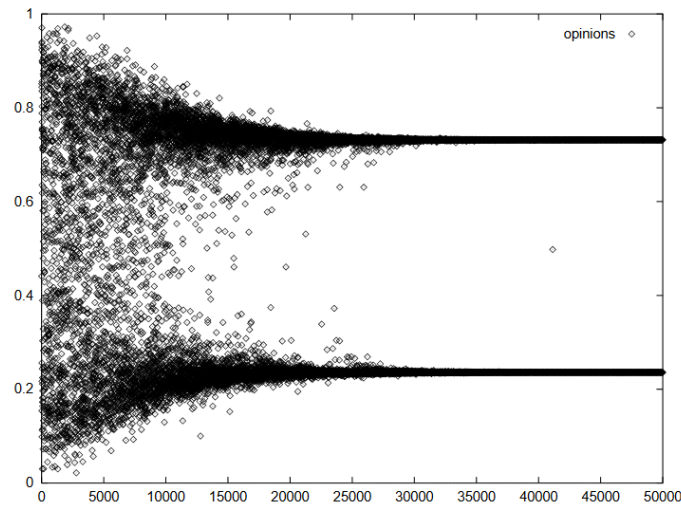


Figure 2: Évolution de l'opinion dans le temps. Résultat obtenu avec les paramètres : $d = 0,2$, $\mu = 0,5$, $N = 1000$.

1.2 Implémentation NetLogo

Le but de ce premier modèle consistera à retrouver les résultats présentés sur les deux figures ci-dessus. Le fichier correspondant est *mixbel-v1.nlogo*.

1.2.1 Entité

Les agents du modèle, les tortues, vont avoir pour attribut une *opinion*. Cette opinion va être un réel compris en 0 et 1. Afin de faciliter la lecture des résultats, les opinions vont être réduites à des réels à deux décimales². On aura donc une fourchette de 101 opinions possibles pour les agents.

1.2.2 Interface utilisateur

L'utilisateur peut faire varier 3 paramètres de la simulation :

- *number-of-agents* qui indique le nombre d'agents présents dans la simulation.
- d , le seuil.
- μ , μ le paramètre de convergence.

1.2.3 Initialisation du modèle

Les agents sont répartis aléatoirement dans l'espace et on leur attribut une opinion aléatoire. Un gradient de couleur est appliqué sur les agents afin de mettre en évidence les différentes opinions dans la simulation. Plus la couleur d'un agent est claire, plus son opinion est proche de 1. Plus sa couleur est foncée, plus son opinion est proche de 0.

1.2.4 Itération du modèle

À chaque pas de temps, on sélectionne deux agents aléatoirement. Si la valeur absolue de leur différence est plus petite que d , alors on calcule la nouvelle valeur de l'opinion des deux agents suivants l'expression

²Ce choix est davantage expliqué en section 1.4 p.4

mathématique section 1.1 p.2. Puis, la couleur des deux agents est mise à jour.

1.3 Résultats

Au début de chaque itération du modèle, on compte le nombre d'opinions différentes. Cette information est affichée dans un champs *monitor*, elle sert également à tracer un graphique de l'évolution du nombre d'opinions en fonction du temps.

Voici les graphiques obtenus pour les mêmes paramètres que ceux de l'article :

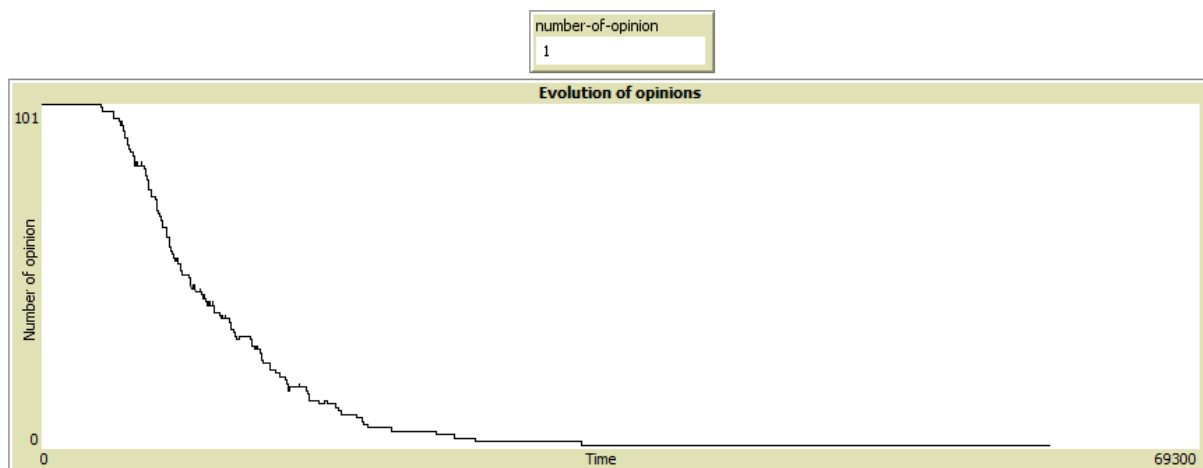


Figure 3: Nombre d'opinions au cours du temps. Résultat obtenu avec les paramètres : $d = 0,5$, $\mu = 0,5$, $N = 1000$.

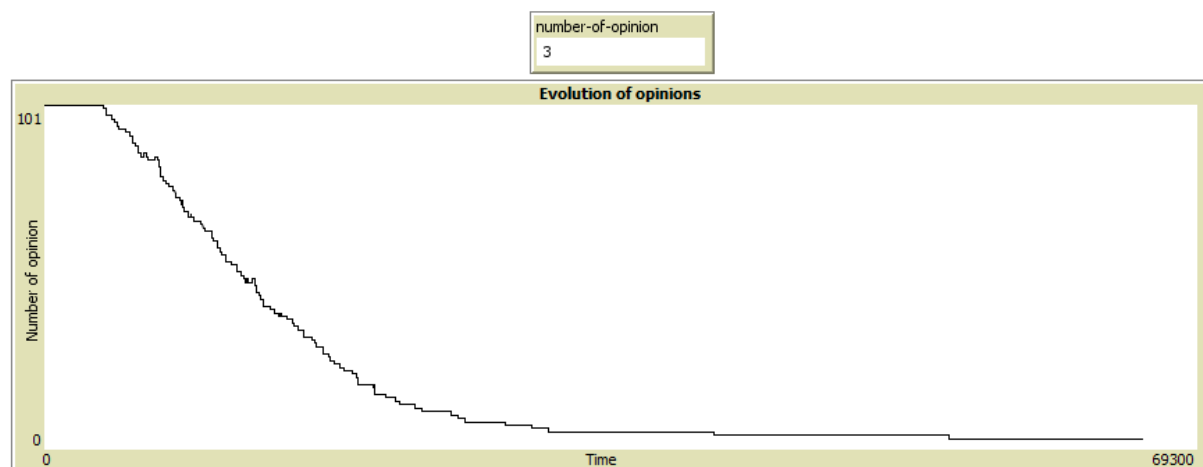


Figure 4: Nombre d'opinions au cours du temps. Résultat obtenu avec les paramètres : $d = 0,5$, $\mu = 0,5$, $N = 1000$.

Sur les figures ci-dessus on observe que le nombre d'opinions décroît à 1 dans le premier cas et à 3 dans le deuxième cas³.

Pour le deuxième cas, l'observation du modèle NetLogo diffère légèrement des résultats de l'article. La figure 2 montre que les opinions des agents doivent converger vers deux valeurs distinctes. Les modèle NetLogo devrait donc compter deux opinions différentes dans *number-of-opinion*.

³Dans l'exemple ci-dessus le nombre final d'opinions est 3, cette valeur peut varier d'une simulation à l'autre mais reste toujours supérieure ou égale à 2.

L'article apporte l'explication suivante : notons $X_{equidistant}$ les agents dont les opinions initiales sont très proches de 0,5 et notons X l'ensemble des autres agents. Lorsque μ est grand, si les $X_{equidistant}$ ne sont pas sélectionnés rapidement par l'algorithme, l'écart entre eux et les autres agents s'accroît si bien qu'à partir d'un certain temps on obtient :

$$|x - x_{equidistant}| \geq d \quad \text{avec } x \in X, x_{equidistant} \in X_{equidistant}$$

1.4 Limite du modèle et perspectives d'évolution

La première limite du modèle porte sur la diversité des opinions. L'opinion d'un agent au temps $t + 1$ est calculée en fonction de son opinion au temps t ainsi qu'une proportion de l'écart entre son opinion et l'opinion de l'autre agent au temps t . La valeur de x_{t+1} peut être un nombre à plus de deux décimales. On a donc une opinion supplémentaire qui est différente des 101 opinions générées à l'initialisation. Si on ne restreint pas la valeur de x_{t+1} à deux décimales, on a une explosion du nombre d'opinions. Cette variété du nombre d'opinions est intéressante mais ne permet pas d'obtenir le résultat voulu qu'est la convergence de l'opinion des agents vers une unique valeur.

La deuxième limite est celle évoquée sur l'analyse du deuxième jeu de test. Le compte du nombre d'opinions ne permet pas de faire ressortir le fait qu'une majorité des opinions va s'accorder sur les mêmes valeurs. Il n'y a que quelques agents qui vont rester à équidistance des valeurs de convergence. Un moyen d'éliminer les agents équidistants serait de faire varier μ au cours de la simulation. Démarrer avec une valeur de μ petite permettrait aux agents de varier rapidement en dehors des valeurs équidistantes. Puis, l'augmentation progressive de μ pendant la simulation assurerait d'atteindre la convergence dans un délai raisonnable.

2 Modèle *Réseaux sociaux*

2.1 Principe du modèle

Dans le modèle précédent, la sélection des agents se faisait aléatoirement parmi l'ensemble des agents possible. Dans ce deuxième modèle, il va s'agir de ne faire interagir un agent qu'avec un certain nombre d'agents faisant partie de son entourage. Le but est de modéliser le fait qu'un individu dans le monde réel interagit avec des personnes de son entourage préférentiellement, plutôt qu'avec des inconnus.

Pour modéliser cela, chaque agent sera représenté par une cellule d'une grille à deux dimensions. Un agent ne peut interagir qu'avec ces quatre voisins. On réalise alors la même procédure que pour le modèle précédent. Dans l'article sont présentés les résultats suivants :

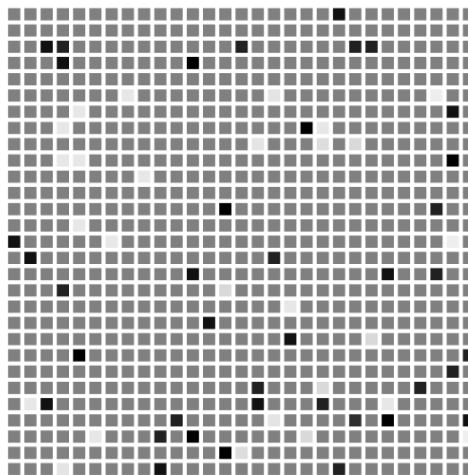


Figure 5: Disposition des opinions finales. Résultat obtenu avec les paramètres : taille de la grille (29x29), $N = 841$, $d = 0, 3$, $\mu = 0, 3$. On observe la percolation d'un grand groupe homogène partageant la même opinion ainsi que quelques éléments "extrémistes".

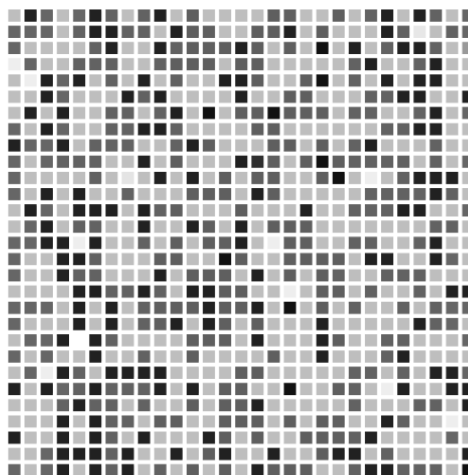


Figure 6: Disposition des opinions finales. Résultat obtenu avec les paramètres : taille de la grille (29x29), $N = 841$, $d = 0, 15$, $\mu = 0, 3$. On observe encore un grand groupe d'opinions homogènes mais également des clusters d'agents d'opinions proches mais non identiques.

2.2 Implémentation NetLogo

Le but de ce deuxième modèle consistera à retrouver les résultats présentés sur les deux figures ci-dessus. Le fichier correspondant est *mixbel-v2.nlogo*. L'interface utilisateur reste la même que pour le modèle précédent.

2.2.1 Entité

Les agents du modèle seront les patches de l'environnement NetLogo. On leur attribut une *opinion* sous la forme d'un réel compris entre 0 et 1. Dans ce modèle, on ne restreint pas le nombre de décimales pour la valeur de l'opinion.

2.2.2 Initialisation du modèle

Les agents sont les patches de l'environnement, on leur attribut une opinion aléatoire. Un gradient de couleur est appliqué sur les agents afin de mettre en évidence les différentes opinions dans la simulation. Plus la couleur d'un agent est claire, plus son opinion est proche de 1. Plus sa couleur est foncée, plus son opinion est proche de 0.

2.2.3 Itération du modèle

À chaque pas de temps, on sélectionne un agent aléatoirement. On sélectionne également l'un de ses quatre voisins. Puis, on calcule la valeur absolue de leur différence, si elle est plus petite que d , alors on calcule la nouvelle valeur de l'opinion des deux agents suivant l'expression mathématique section 1.1 p.2. Puis, la couleur des deux agents est mise à jour.

2.3 Résultats

Dans ce modèle, le graphique a été supprimé car il n'apporte pas d'informations intéressantes. En revanche, la gestion des *ticks* a été implémentée afin d'arrêter la simulation à un instant donné (ici, lorsque $t = 100000$) afin d'avoir les résultats les plus proches de ceux de l'article.

Voici les simulations obtenues pour les mêmes paramètres que ceux de l'article :

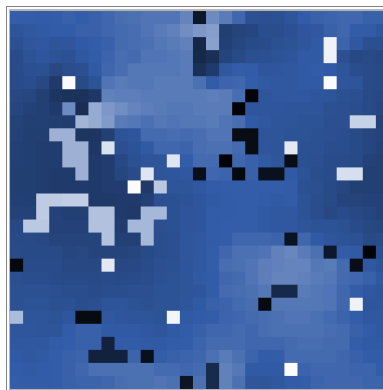


Figure 7: Disposition des opinions finales. Résultat obtenu avec les paramètres : taille de la grille (29x29), $N = 841$, $d = 0,3$, $\mu = 0,3$.

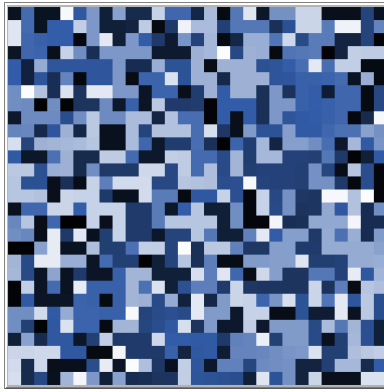


Figure 8: Disposition des opinions finales. Résultat obtenu avec les paramètres : taille de la grille (29x29), $N = 841$, $d = 0, 15$, $\mu = 0, 3$.

Sur la figure 7, le résultat est similaire à celui de la figure 5. On a bien un large groupe d'agents d'opinions homogènes ainsi que quelques éléments de couleur très foncée ou très claire.

En revanche, sur la figure 8 les clusters sont plus petits que ceux observés sur la figure 6. Avec un temps de simulation supérieur à 100 000 itérations, des clusters plus grands ont le temps de se former.

En ce qui concerne les améliorations possibles, les deux vues dans le paragraphe 1.4 p.4 à propos du modèles précédent ne s'appliquent pas ici. En effet, la restriction sur les décimales a été levée. De même, les valeurs d'opinions à équidistance n'existent pas dans ce modèle. Cela est dû au choix de sélection d'agents adjacents.

3 Modèle *Vecteur d'opinions*

3.1 Principe du modèle

Dans ce modèle, on s'intéresse aux vecteurs d'opinions. Plutôt que d'attribuer à chaque agent une unique opinion, on va lui attribuer un vecteur d'opinions de taille m où chaque élément du vecteur représente une opinion sur un sujet. Afin de simplifier le modèle, le vecteur d'opinion est sous formes binaires.

La sélection des agents se fait aléatoirement. Pour déterminer si deux agents ont des opinions proches, on calcule la distance de Hamming entre les deux vecteurs binaires.

Lorsque deux agents ont des opinions proches, l'évolution de leurs opinions se fait de la façon suivante : les opinions communes sont conservées; lorsque l'opinion sur un sujet diffère, un des agents est sélectionné aléatoirement et change son opinion seon une probabilité μ .

3.2 Implémentation NetLogo

Le fichier contenant ce modèle est *mixbel-v2.nlogo*. Malheureusement, le modèle n'est pour l'heure pas complet. En effet, lorsqu'il est exécuté en l'état, bien que les paramètre soit les mêmes que dans l'article, il n'y a pas de convergence visible de l'opinion des agents.