

IMMERSION RECHERCHE SEMESTRE 3

AUTOMATES CELLULAIRES

Paul Gaudin - Lilou Soulas
LDD MPSI

Décembre 2020

Table des matières

1	Abstract	3
2	Introduction	3
3	Principe Théorique	4
3.1	Vue d'ensemble	4
3.2	Principe général	4
3.3	Algorithmes	5
4	Problématiques et résultats de recherche	7
4.1	Modèle de simulation avec ré-injection	7
4.2	Étude des temps de sortie	7
4.2.1	Effet de la friction et du κ sur le temps de sortie	7
4.2.2	Effet des diagonales sur le temps de sortie	7
4.2.3	Étude statistique des temps de sortie pour une recherche industrielle	9
4.3	Effet faster is slower	10
4.4	Étude de l'évacuation d'une foule composée de deux populations	12
5	Conclusion	15
6	Remerciements	15
7	Références	15
8	Annexe	15

1 Abstract

In this research immersion project we worked on cellular automata and more specifically on the study of different characteristics of a crowd and its movements in a square room with only one exit.

The main objectives of this research were the study of the exit time of all the people making up the crowd according to variations in parameters and coefficients, the measure of the exit time of a room filled with two different populations and observation of the faster is slower effect.

To do so, we modelled the crowd and its movements in an algorithmic way and we drawn realistic computer simulations from them in order to answer our problems.

2 Introduction

Un automate cellulaire consiste en une grille régulière de "cellules" contenant chacune un état défini parmi un ensemble d'états et qui évolue au cours du temps. L'état d'une cellule à un certain temps est en fonction de l'état de certaines cellules adjacentes appelé "voisinage" de la cellule en question. En effet, l'état de chaque cellule est régit par un ensemble de règles qui agissent notamment en fonction de l'état des cellules dans son voisinage (nous regarderons ces règles un peu plus tard). A chaque nouvelle étape (donc chaque unité de temps), les cellules évoluent selon les mêmes règles (qui peuvent être appliquées simultanément ou de manière "différée") sur toute la grille, ce qui produit une nouvelle "génération" de cellules qui dépendent entièrement de la génération précédente.

Les automates cellulaires sont beaucoup étudiés en mathématiques et en informatique théoriques, et sont remarquables par l'écart entre la simplicité de leur définition (donc des règles établies à l'origine), et la complexité que peuvent atteindre certains automates. Dans le cadre de nos recherche sur les mouvements de foules, nous nous baserons entièrement sur un automate cellulaire que nous avons modélisé afin d'étudier le temps de sortie d'une ou plusieurs population dans différents cas et part différents modèle, modifié pour aborder différentes parties du sujet, et basé sur les principes et règles suivantes :

Notre modélisation d'automate cellulaire se base entièrement sur les règles évoquées dans le livre "Crowds in equations" de B.Maury et S.Faure.

3 Principe Théorique

3.1 Vue d'ensemble

L'automate cellulaire modélisé dans le cadre de nos recherches se base sur une grille cartésienne dans laquelle chaque cellule peut être occupée ou non par une unique personne (on considérera un 0 pour une case "vide" et un 1 pour une case "occupée"), la grille elle même représentant une pièce carré de taille choisie. Le livre sur lequel nous nous basons décrit une méthode d'évolution séquentielle, dans laquelle chaque cellule est étudiée l'une après l'autre et va se déplacer sur une case libre selon des probabilités (point que nous aborderons plus tard), mais dans cette recherche nous utiliseront uniquement la méthode d'évolution parallèle dans laquelle toutes les cellules évoluent en même temps, et les conflits (lorsque 2 ou plus individus veulent occuper la même cellule) sont ensuite résolus dans un deuxième temps.

3.2 Principe général

On modélise notre automate cellulaire suivant les règles suivantes :

(1) **Règles individuelles.** Notre but étant d'évaluer le temps d'évacuation d'une pièce grâce aux automates cellulaires, nous avons créé un Static Floor Field (SFF) qui est un champ de scalaires discrets (une unique valeur est attribuée par cellule) et qui modélise l'attractivité de chaque cellule. Dans notre situation, le SFF est le même pour tous les agents dans la pièce et l'attractivité de chaque cellule est en fonction de la distance de cette cellule à la sortie. Cette attractivité va ensuite jouer un rôle de biais dans la probabilité de chaque agents d'aller vers une cellule (le SFF sera modélisé de telle sorte que si aucun autres biais n'est ajouté, tous les agents iront spontanément vers la sortie). La Figure 1 représente un SFF tel que modélisé dans nos recherches appliqué à une pièce 50×50 .

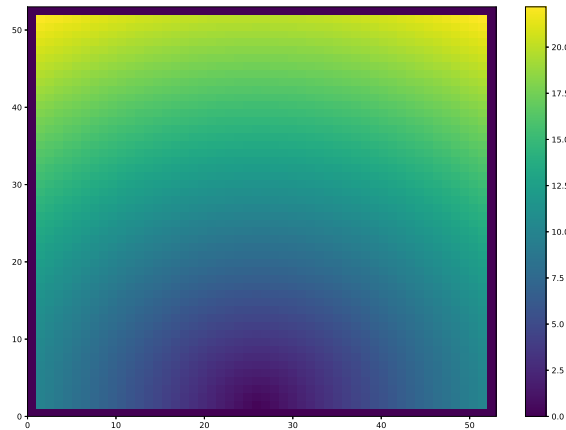


FIGURE 1 – Static Floor Field pour une pièce carrée de côté 50 cellules.

(2) **Interactions.** Les interactions entre plusieurs agents sont basées sur 2 règles en théorie (mais seulement une des deux sera modélisée dans nos recherches) :

1cm(a) *Principe d'exclusion.* Comme dit précédemment, il ne peut y avoir qu'un seul agent dans chaque cellule. Cela signifie que lorsque les mouvements sont opérés, une cellule ayant des cases adjacentes occupées

va avoir une probabilité nulle d'aller sur une case déjà occupée, et comme nous nous basons sur le mode d'évolution parallèle, après avoir généré tout les déplacements, certaines cases peuvent être ciblées par 2 agents. On appelle ces cas des conflits et ces conflits seront par la suite réglés, selon une méthode abordé plus tard.

1cm(b) *Interaction à longue distance.* Un modèle de notre type peut également se voir implanter un modèle d'interaction à longue distance, via l'implantation d'une Dynamic Floor Field (DFF) ajouté au SFF, qui modélise la tendance des populations à suivre le mouvement de foule. Cependant pour des considérations de temps, nous avons décidé de ne pas inclure le DFF.

3.3 Algorithmes

Description de la modélisation. Dans cette section nous décrivons les algorithmes et équations utilisées pour établir les règles de modélisation de notre automate cellulaire : Voici une description informelle de l'algorithme :

On évolue sur une grille cartésienne dans laquelle une case représente à peu près 0.4m et avec une distribution initiale de N particules respectant le principe d'exclusion. Le SFF est défini comme la distance à la sortie.

On va d'abord considérer le mouvement d'un agent :

D'abord on définit comme 0 la probabilité d'aller sur une cellule déjà occupée et, pour toutes les autres, on modélise le déplacement de façon à ce qu'une avancée vers la sortie soit favorisée. Lorsqu'une personne bouge sur la porte de sortie (modélisée par 2 cases au centre de la ligne la plus basse), elle disparaît simplement. Si deux agents veulent aller sur la même cellule, on résout le conflit en comparant la détermination de chaque agent, et ce via la friction. On prévoit ainsi tout les mouvements des cellules de cette manière.

Voici maintenant une version formelle de l'explication précédente :

Équations de modélisation. On cherche à évacuer une pièce rectangulaire $L_x \times L_y$ couverte par des grilles régulières de taille $\Delta x = 0.4m$. On modélise la position de chaque agent par un couple $(i, j) \in [1, N_x] \times [1, N_y]$, avec $N_x = L_x/\Delta x$ et $N_y = L_y/\Delta x$. On considère la configuration initiale comme :

$$p_1^0, p_2^0, \dots, p_N^0, \text{ avec } p_v^0 = (i_v^0, j_v^0)$$

On modélise maintenant le SFF comme suit :

$$S_{ij} = |x_{ij} - x_0|$$

Avec $x_{ij} = ((i-1/2)\Delta x, (j-1/2)\Delta x)$, le centre de la cellule (i, j) et x_0 le centre de la porte (la porte faisant 2 cases, on prendra en réalité x_0 tel que $S_{ij} = \min(|x_{ij} - x_{01}|, |x_{ij} - x_{02}|)$ avec x_{01} et x_{02} le centre des cases correspondant aux deux portes).

On se place à l'étape k (en supposant que l'on connaisse la configuration $(p_1^k, p_2^k, \dots, p_N^k)$), on prévoit le mouvement d'un agent l suivant les équations suivantes :

On définit d'abord une plage de mouvement : en effet on peut considérer un mouvement selon le voisinage de Von Neumann (uniquement les cases strictement à côté de la case occupée par l'agent, c'est cette méthode qui sera retenue en premier lieu) ou celui de Moore (on inclut en plus les diagonales). On définit donc la plage de mouvement par l'ensemble suivant :

$$H = \{(0, 0), (-1, 0), (1, 0), (0, -1), (0, 1)\}$$

A noter que cet ensemble suit le voisinage de Von Neumann, pour suivre le voisinage de Moore on ajoute simplement les mouvements en diagonale : $(-1, -1), (-1, 1), (1, -1), (1, 1)$

On définit maintenant le poids associé à chaque mouvement possible (donc à chaque $h \in H$) :

$$w_h = \exp(-\kappa S_{ij+h}) \delta_{ij+h}, \quad h = (h_x, h_y) \in H \text{ et } ij + h = (i + h_x, j + h_y)$$

Le paramètre δ_{ij+h} vaut 0 si la case est occupée ou si elle est en dehors de la pièce (donc rend nul le poids associée à ce mouvement) et 1 si la case est inoccupée et dans la pièce. Le paramètre κ modélise quant à lui un biais qui peut être vu comme l'intelligence ou la détermination de l'agent à rejoindre la porte, qui va l'amener à aller de manière plus ou moins directe vers la sortie (pour un κ nul, on aura même un déplacement aléatoire dans la pièce, donc une volonté/détermination nulle de sortir), on peut observer l'influence de κ sur la Figure 2.

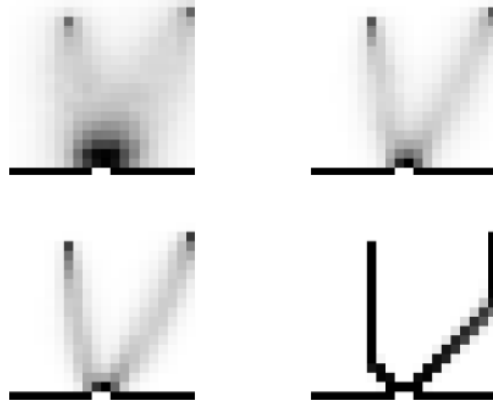


FIGURE 2 – Modélisation de l'influence de κ pour $\kappa = 1, 3, 5, 60 \text{ m}^{-1}$ pour une évacuation de 2 agents

Ici la probabilité de présence sur chaque cellule est modélisé à l'aide d'un fondu gris, on voit bien que pour $= 1 \text{ m}^{-1}$ les trajectoires adoptés sont plus nombreuses et moins directes que pour $= 60 \text{ m}^{-1}$. Les poids sont ensuite re-normalisés pour créer une probabilité de déplacement pour chaque $h \in H$:

$$P(h) = \frac{1}{Z} \times w_h, \quad Z = \sum_{h \in H} w_h$$

On tire enfin un h en fonction de la probabilité de tout les h , et cela détermine le mouvement de l'agent l lors de l'étape $k + 1$.

Maintenant que l'on a prévu les déplacement des agents, on va résoudre les conflits en utilisant la friction : **Friction.** La friction défini une variable $\mu \in [0, 1]$ d'avoir un conflit non résolu (i.e. que les deux agents impliqués dans le conflit retournent à leur case de départ). Quand un conflit se présente on a donc une probabilité μ qu'il ne soit pas résolu, sinon on le résout en comparant le poids des mouvements de chaque agent sur cette case (on récupère w_{h1} et w_{h2} le poids associé au mouvement de 2 agents menant sur la même case et on retiens simplement le plus élevé (l'autre étant renvoyé à la case qu'il occupait au tour k), puis on répète l'opération jusqu'à ce qu'un seul agent finisse son mouvement sur cette case), et en choisissant l'agent ayant le plus fort poids associé à ce mouvement (l'agent le plus déterminé à aller sur cette case).

La mise en oeuvre de cette partie théorique sous forme de code Python se trouve en annexe.

4 Problématiques et résultats de recherche

Dans le but d'étudier les mouvements de foules nous avons modélisé les mécanismes théoriques vu plus haut en langage python. Nous avons utilisé ces fonctions et algorithmes dans plusieurs objectifs de recherche présentés ci après.

4.1 Modèle de simulation avec ré-injection

Tout d'abord, nous avons étudié 3 modèles de simulations : un modèle classique, où l'on place aléatoirement une certaine densité de population dans une pièce, puis on purge la pièce et on mesure le temps de sortie de tout les agents ; un modèle moins aléatoire où l'on détermine le nombre de personne présente dans la pièce et on les places aléatoirement, avant de mesurer le temps de sortie ; et un modèle dit "avec ré-injection" destiné à rendre nul la partie aléatoire du placement des gens dans la pièce, afin d'enlever ce biais potentiel. En effet, cette méthode impliquait qu'à chaque fois qu'un automate passait la porte, il disparaissait et ré-apparaissait au fond de la pièce, de façon à ce qu'une masse continue aille vers la sortie pendant tout le temps d'expérimentation. Ainsi on fait préalablement sortir l'équivalent de 2 fois le nombre de personnes initialement présente dans la pièce, puis à partir de là on mesure le temps mis lors de la troisième purge, qui sera le temps d'évacuation retenu.

Ce modèle devait en plus nous permettre d'étudier les comportements de masses de populations très serrées, mais nous n'avons pas eu le temps de nous y intéresser. De plus, ce modèle augmentait énormément nos incertitudes (qui devenait de l'ordre de 0.5 fois la valeur moyenne contre 0.05 fois la valeur moyenne sans ré-injection, soit un facteur 10), ce qui nous a poussé à le mettre de côté pour la suite de nos recherches. Dans les parties suivantes, nous utiliserons donc le modèle classique (fonctionnant avec la densité de population) ou le modèle moins aléatoire (fonctionnant avec un nombre initial de personnes à mettre dans la salle.)

4.2 Étude des temps de sortie

L'étude des temps de sortie est une part importante de nos recherches puisque ces temps sont des indicateurs importants lors de l'évacuation d'une foule. Voici l'explication de trois études menées sur ces temps.

4.2.1 Effet de la friction et du κ sur le temps de sortie

On illustre l'effet du paramètre κ et μ sur le temps de sortie par la Figure 3. (On notera qu'on a volontairement exclu les valeurs très basses de κ car en dessous d'un certain seuil le temps de sortie croît énormément).

On remarque comme attendu que l'augmentation de κ entraîne une diminution du temps de sortie (des agents plus déterminés vont logiquement sortir plus vite, sauf dans certains cas (voir effet faster is slower(4.3))), cependant cette diminution a tendance à se stabiliser à partir d'un certain κ .

Quant à μ , on observe qu'à l'inverse de κ , son augmentation entraîne une augmentation du temps de sortie, due au fait qu'un fort coefficient de friction aura tendance à clouer sur place certains agents si trop d'entre eux visent la même case (les cases juste devant la porte de sortie notamment).

4.2.2 Effet des diagonales sur le temps de sortie

Comme présenté dans la partie théorique, deux modèles sont possibles pour modéliser les déplacements : le voisinage de Von Neumann (possibilité de déplacement uniquement sur les 4 cellules directement adjacentes

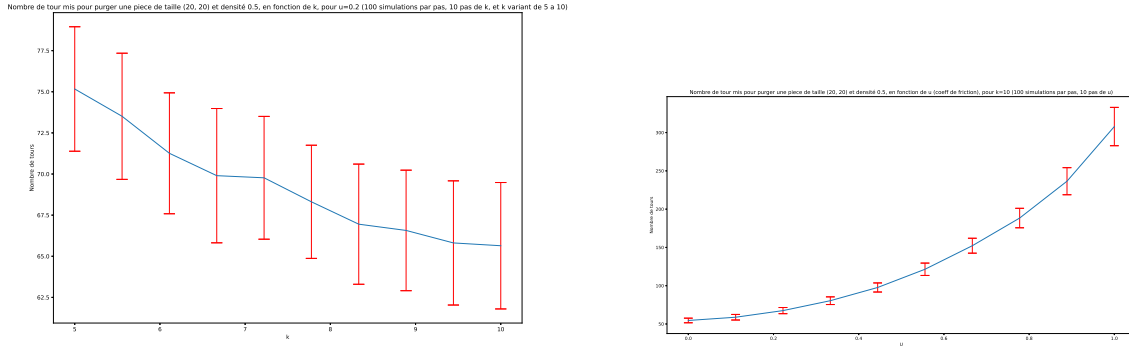


FIGURE 3 – Graphique représentant l'évolution du temps de sortie en fonction de κ à gauche, et en fonction de μ à droite.

à la cellule en question) ou celui de Moore (qui inclut en plus des quatre cellules strictement adjacentes les quatre cellules en diagonales). Dans un premier temps, nous avons travaillé selon le voisinage de Von Neumann et nous avons seulement après, adopté le voisinage de Moore. L'objectif ici était de comparer le temps de sortie d'une foule dans une salle donnée selon ces deux voisinages, et d'en conclure les différences qu'impliquait l'intégration ou non des diagonales dans les possibilités de mouvements. Nous avons donc codé sous python deux fonctions de simulation d'évacuation d'une pièce, chacune fonctionnant selon un des voisinages évoqués.

On fait tourner les deux simulations plusieurs fois et on observe les résultats sur la Figure 4.

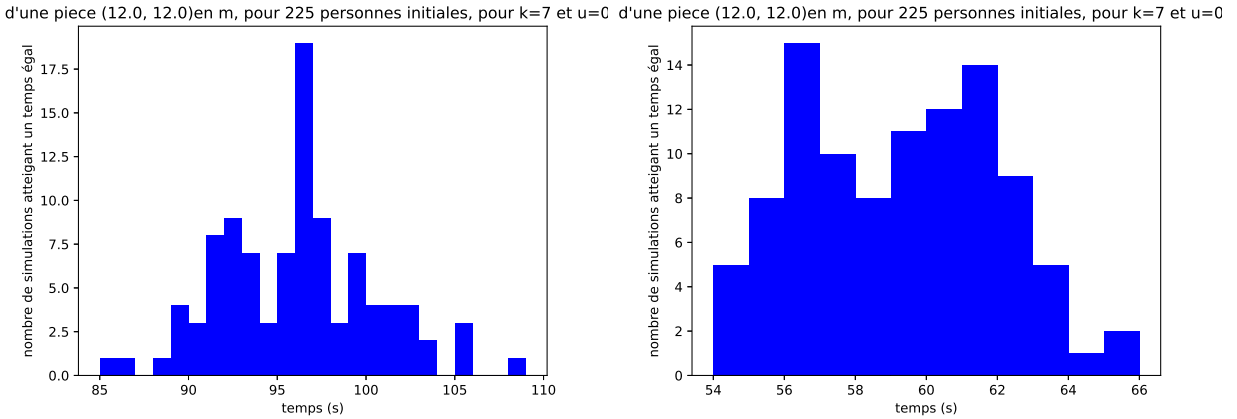


FIGURE 4 – Temps de sortie selon le voisinage de Von Neumann (sans diagonales) à gauche ou Moore (avec diagonales) à droite pour une pièce de 30x30 et 225 personnes initiales, $\kappa=7$, $\mu=0.8$

On peut voir sur les deux graphes ci dessus que lorsque les diagonales sont incluses dans les déplacements autorisés, le temps d'évacuation est plus faible. On voit sur le graphe de gauche (sans diagonales) que les temps de sortie sont compris entre 85s et 110s, alors que pour le graphe de droite (avec diagonales) les temps de sortie vont de 54s à 66s. En effet, avoir des déplacements en diagonale permet d'atteindre les portes avec

des chemins plus directs, cependant cela augmente les conflits : plus il y a de cases où se rendre, plus il y a de chance de tomber sur une case qui est également convoitée par quelqu'un d'autre.

Ainsi, on peut en conclure que pour que l'évacuation aille plus vite, il faut inclure les déplacements en diagonale. Cela permet également de rapprocher notre simulation d'une situation réelle : en effet, dans une pièce, une personne n'avance pas seulement en avant, en arrière ou sur le côté, mais prend bien la direction qu'elle veut. Se placer selon le voisinage de Moore permet donc une plus grande possibilité d'action et une fluidification des mouvements vers la sortie (mais entraîne plus de conflits).

Dans la suite de nos recherches, nous utiliserons tantôt le voisinage de Von Neumann et tantôt celui de Moore selon les besoins de la simulation.

4.2.3 Étude statistique des temps de sortie pour une recherche industrielle

Dans cette partie, l'objectif fixé était d'aider par nos moyens de simulations informatique à la définition de normes de sécurité en bâtiment. En effet lors de construction de grandes pièces, comme des salles de concert ou de réception, il est utile d'estimer le nombre maximum de gens que l'on peut y faire entrer tout en garantissant la sécurité de tous en cas d'évènement nécessitant une évacuation.

Pour cela, nous avons donc fait un très grand nombre de simulations d'évacuation d'une pièce pour obtenir un temps moyen de sortie associé : cela permet de comparer avec les valeurs de sécurité, et ainsi de déterminer le nombre idéal de personnes à faire entrer dans cette salle. On réalise également sur ce grand nombre de temps de sortie obtenus une étude statistique qui permet de mieux étudier la répartition des différentes valeurs de temps.

Ici, on utilise le voisinage de Moore afin de se placer au plus prêt d'une situation réelle.

On imagine une salle de concert carrée vide de 12m par 12m avec une seule porte de 0.8m de large. (Ici, cette modélisation peut ne pas sembler très réaliste mais les informations qu'elle produit peuvent être utiles, et le changement des grandeurs de la porte se fait facilement dans les codes. Ainsi son utilisation et son amélioration pour des simulations plus réalistes est tout à fait possible.) On place initialement 400 personnes à l'intérieur de la salle et on fixe κ à une valeur de 7 et μ à une valeur de 0,8. On réalise 10 000 simulations de l'évacuation de cette salle. Le graphe obtenue est présenté sur la Figure 5.

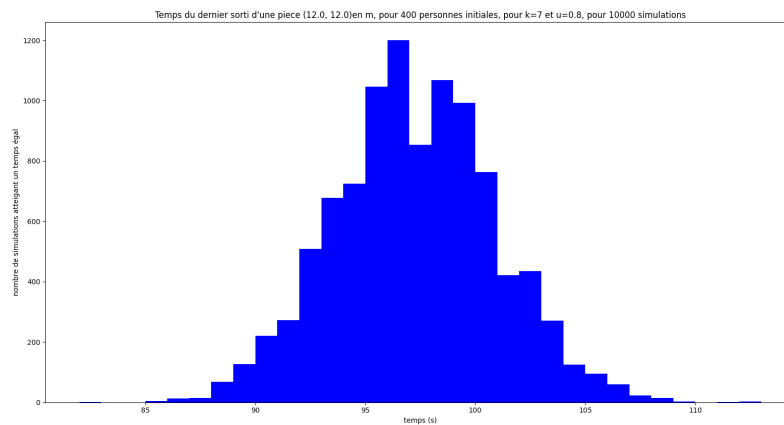


FIGURE 5 – Temps du dernier sorti d'une pièce 8x8m, pour 150 personnes initiales, et des valeurs de $\kappa=5$ et $\mu=0,8$.

On obtient en même temps que le graphe l'étude statistique associée :

- Nombre de valeurs = 10000
- Moyenne = 97.25s
- Médiane = 97.2s
- Variance = 13.28
- Écart Type = 3.64s
- Premier quartile = 94.77s
- Deuxième quartile = 97.2s
- Troisième quartile = 99.63s

Grâce au graphe et à l'étude statistique, on peut estimer avec assez de précision l'intervalle dans lequel le temps d'évacuation de la salle pour la pièce donnée se situe. On peut ainsi valider ou non la norme de sécurité sur l'évacuation de cette pièce pour un certain nombre de personnes à l'intérieur.

Les codes créés lors de ce projet de recherche sont accessible via un lien présent en annexe. Toute modification pour rendre les simulations plus réalistes (pièces rectangles, plusieurs portes, obstacles au milieu, inclusion des diagonales dans les possibilités de mouvement, etc) est la bienvenue.

4.3 Effet faster is slower

Principe de l'effet Faster is Slower. Lorsqu'on s'intéresse à l'étude des foules et plus particulièrement à leur évacuation, on peut observer d'après les expériences un effet assez contre intuitif et pourtant très important : l'effet faster is slower.

L'expérience la plus simple pour observer ce phénomène consiste à rassembler une foule dans une pièce puis à lui demander d'en sortir tout en chronométrant l'opération jusqu'à ce que le dernier soit sorti. Ensuite, il suffit de recommencer l'expérience mais cette fois en demandant à chacun de se précipiter sur la sortie comme lors d'une urgence.

On remarque alors que lorsque l'on demande aux gens de se presser, il faut compter en moyenne 25 % de temps supplémentaire pour faire sortir tout le monde que lorsque chacun sort à son rythme.

On arrive à la conclusion plutôt paradoxale que si la foule se dépêche, elle sort moins vite.

Pour que l'évacuation soit efficace, il faut donc que chacun aille vite mais pas trop, sinon cela ralentit le temps de sortie, ce qui est à l'opposé de l'objectif escompté.

Dans notre contexte de recherche, l'objectif était de pouvoir observer, sur un graphe des temps de sortie en fonction du paramètre κ , l'effet faster is slower et de pouvoir quantifier la valeur de κ qui rendrait l'efficacité de l'évacuation maximale.

Principe de génération du graphe. Pour obtenir un graphe intéressant qui nous permettrait d'observer l'effet faster is slower, on s'intéresse essentiellement à la variation du paramètre κ qui représente "la volonté des agents à rejoindre la sortie". Donc plus un agent à un κ élevé, plus il se dirigera rapidement vers la sortie, c'est-à-dire qu'il prendra l'itinéraire le plus efficace et direct qui le relie à la porte.

Nous avons implanter pour ce cas là, la fonction SimulationK suivante :

SimulationsK(d,dim, μ , κ_{min} , κ_{max} ,Npas,Nsim)

Cette fonction prend en paramètre une densité ou un nombre de personnes à injecter "d" (les deux méthodes de créations de salles sont possible dans nos codes), les dimensions de la salle "dim", la valeur du paramètre " μ ", la valeur minimale de κ voulue " κ_{min} ", la valeur maximale de kappa voulue " κ_{max} ", le nombre de pas "Npas" à prendre entre " κ_{min} " et " κ_{max} " ainsi que le nombre de simulations "Nsim" à faire pour chaque pas de κ .

Principe de la fonction. On initialise une salle dont les dimensions "dim" restent inchangées pour toutes les simulations. On fait varier kappa entre " κ_{min} " et " κ_{max} " selon le nombre de pas voulus. On lance, pour chacune de ces valeurs, un nombre "Nsim" de simulation d'évacuation. Cette simulation est achevée en un nombre "Nit" d'itérations. On réalise alors la moyenne de toutes les valeurs "Nit" obtenues pour un κ donné. Une itération de l'algorithme correspond à environ 0.27 secondes lors d'une expérience réelle. On peut donc associer le nombre d'itérations moyen à un temps de sortie moyen, mais pour l'utilisation que nous ferons des graphes, cette conversion n'est pas nécessaire.

On obtient donc pour chaque κ , un nombre d'itération moyen associé. Au départ, cette fonction réalisait un déplacement des agents qui ne prenait pas en compte les déplacements en diagonales. Les graphes obtenus n'étant pas satisfaisants, nous avons donc ici inclus les déplacement en diagonales, qui, comme on l'a vu dans la partie 4.1.2, augmentent les conflits, ce qui pourrait permettre ici d'observer mieux encore l'effet faster is slower.

On peut ainsi tracer le graphe obtenu du nombre d'itérations moyen pour faire sortir tout le monde en fonction de κ ainsi que ses incertitudes.

Voici, sur le graphe de la Figure 6, la courbe du nombre d'itérations en fonction de κ pour évacuer une pièce de taille (20,20) soit 8x8m, contenant 200 personnes.

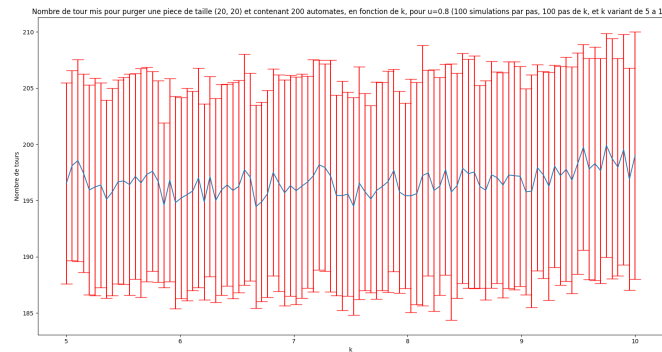


FIGURE 6 – Nombre d'itérations nécessaire pour vider une pièce de taille 8x8m contenant 200 personnes, μ valant 0.8, κ allant de 0 à 10 en 10 pas et avec 100 simulations effectuées par pas.

D'après la théorie, on devrait pouvoir observer sur ce graphe une diminution des temps de sortie quand κ augmente puis une ré-augmentation lorsque κ devient grand : à partir d'une certaine valeur de κ , plus les agents se pressent, plus le temps de purge (proportionnel au nombre d'itérations) est important.

En observant le graphe précédent, on peut identifier une très légère remontée de la courbe à partir de $\kappa=9$

environ, mais cette identification n'est pas claire. L'effet *Faster is Slower* est donc un effet difficile à obtenir dans notre cas et nous n'avons pas vraiment réussi à le visualiser sur nos graphes.

Le problème peut venir de causes multiples : valeur de μ trop faible, pièce trop petite, absence de DFF,...

4.4 Étude de l'évacuation d'une foule composée de deux populations

Nous avons jusqu'ici travaillé avec une foule comportant des individus semblables dans le sens où ils possédaient la même "volonté à rejoindre la sortie" c'est à dire la même valeur de κ .

Mais imaginons maintenant que nous ayons, dans une même salle initiale, des agents possédant un κ élevé et d'autres ayant un κ plus faible. Si l'on devait trouver dans la réalité des profils modélisant bien ces deux cas, l'on pourrait prendre l'exemple de ce que l'on appellera les "brutes" et les "polis" :

Les brutes. Ce que l'on va appeler "les brutes" sont les personnes à κ élevé. Ils ont une très forte volonté d'aller vers la porte, et en cas de conflit, ils sont plus costauds et plus virulents, et sont donc ceux qui prennent la place si ils sont face à une personne de κ plus faible.

Les polis. Ce qu'on appellera les "polis" sont donc les agents à κ plus faible. Lors de conflits, quand à eux, ils lèguent systématiquement leur place aux "brutes" et retournent à leur position précédente.

En effet, si l'on se place dans une situation de tous les jours, par exemple au pied d'un escalator dans un super marché, il est facile de visualiser les deux comportements que nous venons de décrire. Imaginons que deux personnes, une "brute" et une "polie", soient à un pas de monter sur l'escalator : une personne âgée, très polie de surcroît, et en face, un homme costaud, très pressé et marchant très vite. D'après vous, quel personnage entrera le premier sur l'escalator ? On peut aisément s'imaginer que le costaud passera en premier, avant la personne âgée. Même si le cas contraire, que l'homme costaud laisse passer la personne âgée, existe sûrement, nous utiliserons dans la suite de cette partie la logique que la "brute" passe avant la personne "polie".

L'objectif ici, toujours dans un contexte d'évacuation d'une salle, était d'étudier les temps de sortie de cette foule à deux populations lorsque l'on faisait varier la proportion de chacun des deux types de personnages. En effet, cette étude a pour but final de déterminer en quelles proportions de "brutes" et de "polis" la foule devrait être constituée pour que l'efficacité de l'évacuation soit maximale.

Selon la théorie, on pourrait se dire que si l'on constitue la foule entièrement de "brutes", alors, il y a de grandes chances que l'évacuation soit ralentie, car étant donné que tout les agents vont se diriger droit vers la sortie, ils vont souvent vouloir se retrouver sur les mêmes cases (près de la sortie notamment), et ainsi entraîner un grand nombre de conflits. Ce cas est similaire à ce que l'on a expliqué dans la partie *faster is slower* (4.2) : plus les agents se pressent et ont une forte volonté d'atteindre la sortie sans se soucier des autres, moins l'évacuation est efficace.

A l'inverse, si l'on constitue la foule intégralement de "polis", là encore les agents auront une plus forte tendance à se balader sans forcément cibler la sortie à chaque tour, ce qui augmentera évidemment le temps d'évacuation. L'on pourrait représenter ce qui arrive entre les deux personnes par un phénomène que l'on a tous sûrement déjà vécu dans la vie de tous les jours : un dialogue poli constitué de nombre de "après vous" et de "à vous l'honneur" qui peut durer parfois plusieurs secondes avant que l'un ou l'autre n'avance en premier. Comme une foule constituée seulement de "brutes", une foule contenant entièrement des "polis" ne donneras pas un temps de sortie des plus efficaces.

Ainsi, théoriquement, la foule doit contenir les deux populations, et nous avons essayé de savoir en quelle proportion cela devait se faire pour que le temps d'évacuation soit maximal.

Pour réaliser cette étude nous avons légèrement modifié nos fonctions de résolutions afin qu'elles prennent

en paramètres une proportion de population $p \in [0, 1]$, $p = 0$ indiquant que seule la population "faible" ou "polie" sera présente, et $p = 1$ indiquant au contraire que seule la population "forte" ou "brute" sera modélisée. La Figure 7 représente une modélisation typique d'une pièce 20×20 avec $p = 0.5$, à l'étape initiale.

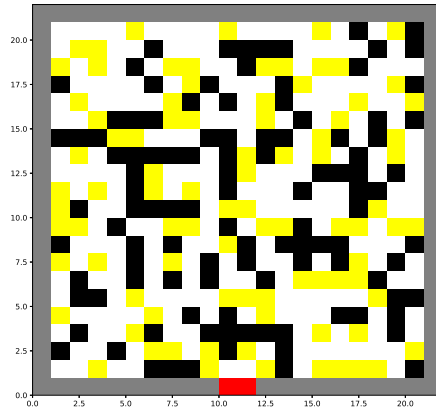


FIGURE 7 – Modélisation d'une pièce contenant 2 population en proportion égale ($p = 0.5$), on remarque en noir la population "faible" et en jaune la population "forte".

On a ensuite fait tourner des batterie de tests comme précédemment, afin d'en retirer un graphique, visible sur la Figure 8, représentant l'évolution de la vitesse d'évacuation en fonction de la proportion des populations, à μ , κ_1 et κ_2 fixés :

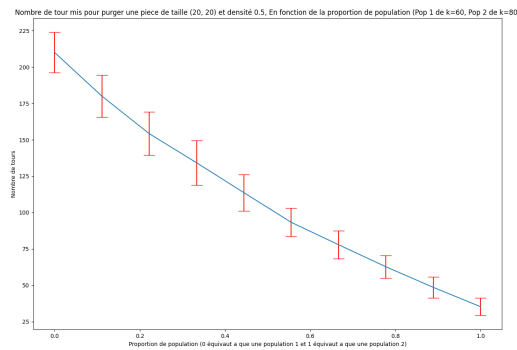


FIGURE 8 – Graphique représentant l'évolution de la vitesse d'évacuation en fonction de la proportion des populations.

On observe sur le graphe que, contrairement à toutes nos théories, le temps d'évacuation le plus faible est pour une foule constituée intégralement de brutes. Le résultat n'est pas celui attendu, et donc au final, on ne peut pas être certain que le mélange des population soit ce qui est le plus efficace. Nos calculs ici semblent « montrer » que l'idéal est une foule composée de brutes, et pour confirmer cette théorie ou pour au contraire l'invalider, il faudrait creuser encore plus. On pourrait par exemple estimer, pour avoir une

idée plus précise, le nombre de conflit qui se sont produits, plus précisément le nombre de conflits mixtes (poli-brute) ou homogène (poli-poli ou brute-brute).

Cette étude de foule à deux population étant très exploratoire et la réponse à cette étude n'ayant pas encore été trouvée, il reste beaucoup de choses à tester ou à modéliser pour pouvoir affirmer qu'un mélange ou au contraire une foule composée uniquement d'une seule population est ce qui mène à un temps d'évacuation le plus efficace.

5 Conclusion

Dans ce projet d'immersion recherche nous nous sommes donc intéressé à un automate cellulaire permettant de modéliser l'évacuation d'une foule. A partir des éléments théoriques à notre disposition nous avons codé en langage python plusieurs fonctions nous permettant de réaliser des simulations d'évacuation, et de tracer des graphes. En effet, ces productions nous ont permis de pouvoir étudier le temps de sortie d'une foule selon différents paramètres comme la friction, le paramètre κ ou encore la possibilité ou non de déplacement en diagonale. Nous avons pu conclure que l'évacuation sera plus efficace si κ est plutôt grand et si les conflits sont souvent résolus (μ petit). Nous avons également pu réaliser une étude statistique sur un grand nombre de temps de sortie dans le but de pouvoir obtenir un temps d'évacuation moyen nécessaire à la définition de norme de sécurité en bâtiment. Ensuite, nous avons tenté d'observer l'effet *faster is slower* sur un graphe, étude que nous n'avons pas réussi à mener avec succès mais dont nous avons pu déduire que si κ est trop grand, l'évacuation sera moins efficace.

Pour finir, nous nous sommes penché sur l'étude d'une foule composée de deux populations différentes afin de déterminer en quelle proportion de chaque population l'évacuation de la foule était la plus efficace. N'ayant pas abouti à un graphe en accord avec la théorie, il faudra creuser d'avantage sur ce domaine pour arriver à un résultat satisfaisant.

Comme nous avons pu le voir, l'étude des foules est un sujet de recherche immense où l'utilisation des automates cellulaire est indispensable. Modéliser les mouvements de la foule nous a permis de comprendre les différents paramètres essentiels qui entrent en jeu. Et enfin, tirer de ces modélisations des simulations nous a montré que ces recherches peuvent permettre d'aider dans de nombreux domaines concrets, notamment en lien avec la sécurité.

Pour finir sur nos recherches dans ce projet, nous pouvons dire que nous avons passé beaucoup de temps sur le codage et sur l'exploration de différents domaines. Cela nous a permis de toucher à nombre de choses mais nous a peut-être empêché d'explorer en profondeur l'un de ces domaines. De plus, nous sommes resté dans des valeurs de κ et μ toujours assez similaires. Rester dans ces plages de valeurs et explorer beaucoup d'éléments mais, plus ou moins en surface, nous a peut-être fait passer à côté de l'observation de certains phénomènes comme l'effet *faster is slower* ou encore un graphe sur deux populations correspondant véritablement à nos attentes. Malgré cela, le code existe toujours et il ne tient qu'à nous (et vous ?) d'explorer encore plus de configurations.

6 Remerciements

Nous remercions chaleureusement Bertrand Maury pour ses explications, sa confiance, son intérêt et son aide tout au long de ce projet d'immersion recherche.

7 Références

(liens en hypertexte)

- "Crowds in equations" de Bertrand Maury et Sylvain Faure.
- "Le dilemme de l'évacuation" de la chaîne Fouloscopie (Mehdi Moussaid).

8 Annexe

- Le code que nous avons créé dans le cadre de cette recherche, permettant toutes les simulations

et modèles présentés.