Automates cellulaires et robustesse aux erreurs : une perspective mathématique

Irène Marcovici [1]

La difficulté de concevoir des systèmes robustes à des erreurs est une problématique commune à différents domaines de l'informatique. Comment mener à bien un calcul si des erreurs sont susceptibles de se produire ? Comment détecter et corriger des anomalies dans un réseau, en l'absence d'autorité centrale permettant de contrôler l'ensemble du fonctionnement ? Alors que les organismes vivants présentent tous une certaine capacité à se réparer lorsqu'ils sont soumis à une perturbation, c'est rarement le cas des systèmes artificiels, pour lesquels une petite perturbation locale peut mener à un dysfonctionnement complet.

Les automates cellulaires offrent un modèle de choix pour étudier de manière théorique dans quelle mesure un système informatique distribué peut avoir la capacité de se stabiliser en présence d'un bruit aléatoire. Formellement, on suppose que l'information est encodée sur un réseau régulier (ruban infini dans le cas de la dimension 1, grille de dimension 2 ou plus), constitué de *cellules*, qui contiennent chacune un état parmi un ensemble fini d'états possibles. L'action d'un automate cellulaire consiste alors à mettre à jour les états de toutes les cellules simultanément, en appliquant pour chaque cellule une même règle locale de mise à jour, qui dépend seulement de l'état de la cellule et de celui de quelques-unes de ses voisines. Les automates cellulaires constituent ainsi un modèle de calcul parallèle, à la fois suffisamment simple pour permettre une étude mathématique rigoureuse, et suffisamment riche pour apporter un éclairage sur les comportements de systèmes informatiques concrets.

1. Institut Élie Cartan de Lorraine, université de Lorraine.

DOI:10.48556/SIF.1024.19.133 **1024** – Bulletin de la SiF, numéro 19, Avril 2022, pp. 133–136

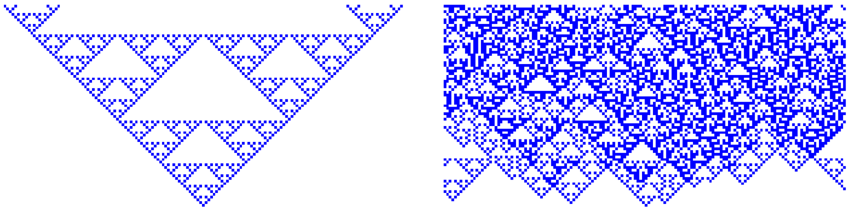

FIGURE 1. Évolution (du bas vers le haut) de l'AC dont la règle locale consiste à calculer la somme (modulo 2) des voisines de gauche et de droite (les carrés bleus représentent les 1, les carrés blancs les 0). À partir d'une configuration initiale constituée d'un unique carré bleu, on observe un triangle de Sierpiński en l'absence d'erreurs *(gauche)*, tandis qu'avec une probabilité $\varepsilon = 0.01$ d'erreurs *(droite)*, l'AC converge vers un état d'équilibre qui ne dépend pas de la configuration initiale.

La figure 1 présente l'évolution d'un automate cellulaire (AC) de dimension 1, à états binaires. Cet exemple élémentaire illustre déjà l'écart entre la simplicité de la règle locale et la complexité qu'elle peut engendrer au niveau macroscopique. Il permet également d'entrevoir l'intérêt d'une perspective mathématique des automates cellulaires.

Calcul bruité

On peut montrer qu'il est possible de simuler l'évolution de n'importe quelle machine de Turing à l'aide d'un automate cellulaire. Mais lorsque des erreurs se produisent, le calcul perd souvent toute signification. Plus précisément, supposons qu'à chaque étape de l'évolution de l'automate cellulaire, chaque cellule est susceptible d'être mise à jour avec un état qui ne correspond pas à la valeur donnée par la règle locale. Même si la probabilité d'une telle erreur est extrêmement petite, au cours de l'évolution, on observe généralement que l'automate cellulaire oublie progressivement toute l'information qui était encodée dans la configuration donnée en entrée, et converge vers un état d'équilibre qui ne dépend plus du tout de la configuration initiale (une illustration en est donnée en figure 1 : en présence d'erreurs, l'automate cellulaire atteint rapidement un comportement *stationnaire*, avec des motifs similaires qui se répètent d'une ligne à l'autre). En termes mathématiques, on dit que le système est *ergodique*. En dimension 1, la célèbre conjecture des taux positifs affirmait que tout automate cellulaire soumis à des erreurs aléatoires est ergodique. Peter Gàcs [Gàcs01] a réfuté cette conjecture en 2001, en proposant un contre-exemple extrêmement sophistiqué, dans un article de plus de 200 pages. Mais si on se limite aux familles d'automates cellulaires les plus simples, nous avons montré récemment dans un travail avec Mathieu Sablik et Siamak Taati [MST19] qu'ils ne sont effectivement pas robustes à des erreurs aléatoires. En dimension supérieure ou égale à 2, des familles de contre-exemples avaient été proposées par Andrei Toom [Toom80] dès 1980, mais le problème n'est pas trivial pour autant, et encore aujourd'hui, l'ergodicité des automates cellulaires bruités soulève de nombreuses questions, qui intéressent aussi bien les informaticiennes et les informaticiens que les spécialistes de probabilités et de systèmes dynamiques.

Auto-correction de pavages

Avec Nazim Fatès et Siamak Taati [FMT22], nous avons fait un pas de côté pour étudier la capacité des automates cellulaires à corriger des erreurs dans des pavages. En un sens, il s'agit de comprendre comment reproduire sur un modèle artificiel simple certains mécanismes d'auto-stabilisation présents dans la nature.

On se donne un pavage de la grille, c'est-à-dire un coloriage des cellules de la grille bi-dimensionnelle avec une palette finie de couleurs, où l'agencement des couleurs doit respecter certaines contraintes locales. On suppose que les couleurs de certaines cellules sont arbitrairement changées par de nouvelles couleurs, ne respectant pas forcément les contraintes du pavage. Peut-on retrouver une configuration respectant les contraintes du pavage, par une procédure locale ? En d'autres termes, existe t-il un automate cellulaire, qui, à partir de n'importe quelle perturbation finie du pavage, se stabilise au bout d'un temps fini sur une configuration valide ? Même si par essence, les pavages sont définis par un ensemble de contraintes locales, il n'est pas toujours aisé d'élaborer des règles locales d'auto-correction.

Pour nous forger une intuition, considérons le cas des k-coloriages. Un k-coloriage est un coloriage des cellules de la grille avec une palette contenant k couleurs, qui doit respecter la contrainte que deux cellules adjacentes (c'est-à-dire qui ont une arête commune) doivent toujours être de couleurs différentes. Pour $k = 2$, les seules configurations valides correspondent aux deux configurations en damier, et on peut utiliser leur périodicité spatiale pour construire un automate cellulaire permettant de reconstruire rapidement le damier si des erreurs ont été introduites. Pour $k \geqslant 4$, on peut au contraire exploiter le fait que le nombre de couleurs disponibles est suffisamment grand pour offrir une certaine souplesse, ce qui permet également de retrouver rapidement une configuration valide par des opérations locales. Le cas $k = 3$ est quant à lui plus complexe. Nous n'avons pas pu trouver de règle auto-correctrice à ce jour et nous pensons que le problème est difficile. En effet, comme illustré en figure 2, il existe des configurations où seules deux cellules adjacentes ont la même couleur et où, pourtant, il faut modifier une zone arbitrairement grande pour retrouver une configuration valide, ce qui laisse penser que si une solution existe, elle ne saurait être simple.

Si le statut des 3-coloriages reste ouvert à ce jour, on sait cependant que tous les pavages ne peuvent pas être corrigés facilement par un automate cellulaire. Nous

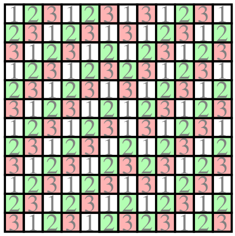

FIGURE 2. Exemple de configuration d'un 3-coloriage où seules deux cellules adjacentes (au milieu) sont de même couleur blanche, mais pour laquelle il est nécessaire de modifier une zone beaucoup plus grande si l'on souhaite retrouver une configuration valide.

avons en effet montré que si P≠NP, il existe des pavages pour lesquels la correction par un automate cellulaire, si tant est qu'elle soit possible, ne peut pas être effectuée en temps polynomial par rapport à la taille de la zone qui a été altérée.

À l'inverse, nous avons présenté différentes familles de pavages pour lesquelles des mécanismes d'auto-stabilisation efficaces peuvent être proposés, typiquement en temps linéaire par rapport à la taille de la modification. Dans ce cas, on obtient même un résultat plus fort : l'automate cellulaire permet non seulement de corriger des erreurs qui auraient été introduites sur une zone finie de la configuration, mais aussi de corriger des erreurs aléatoires réparties sur toute la grille, si la probabilité d'erreur est suffisamment petite.

Notre approche, à la croisée entre mathématiques et informatique, nous a fait étudier le problème de la robustesse aux erreurs dans le cadre abstrait d'un réseau régulier sur lequel l'évolution des différentes cellules est régie par une même règle locale. Les modèles retenus, malgré leur simplicité, apportent un éclairage sur le fonctionnement des réseaux distribués actuels, en fournissant quelques pistes pour développer leurs capacités d'auto-régulation, et en permettant également de mieux comprendre les limites théoriques qui existent.

Références

[Gàcs01] P. Gács. Reliable cellular automata with self-organization. *J. Stat. Phys.*, 103(1–2) : 45–267, 2001.

[MST19] I. Marcovici, M. Sablik, et S. Taati. Ergodicity of some classes of cellular automata subject to noise. *Electron. J. Probab.*, 24 : 1–44, 2019.

[FMT22] N. Fatès, I. Marcovici, et S. Taati. Self-stabilisation of cellular automata on tilings. *Fundam. Inform.*, 2022 (à paraître).

[Toom80] A. Toom. Stable and attractive trajectories in multicomponent systems. *Multicomponent random systems, Adv. Probab. relat. Top.*, 6 : 549–575, 1980.