



HAL
open science

Robustesse et émergence dans les systèmes complexes : le modèle des automates cellulaires

Jean-Baptiste Rouquier

► **To cite this version:**

Jean-Baptiste Rouquier. Robustesse et émergence dans les systèmes complexes : le modèle des automates cellulaires. Autre [cs.OH]. Ecole normale supérieure de lyon - ENS LYON, 2008. Français. NNT: . tel-00354042

HAL Id: tel-00354042

<https://theses.hal.science/tel-00354042>

Submitted on 18 Jan 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 492

N° attribué par la bibliothèque : 07 ENSL 0492

THÈSE

en vue d'obtenir le grade de
Docteur de l'École normale supérieure de Lyon
Spécialité informatique

Université de Lyon
École doctorale de mathématiques et informatique fondamentale
Laboratoire de l'informatique du parallélisme

présentée et soutenue publiquement le
8 décembre 2008
par

Jean-Baptiste ROUQUIER

Robustesse et émergence dans les systèmes complexes : le modèle des automates cellulaires

devant la commission d'examen formée de :

Directeur de thèse : Michel MORVAN

Rapporteurs : Jacques DEMONGEOT et Eric GOLES

Examineurs : Hugues BERRY, Paul BOURGINE et Bruno DURAND

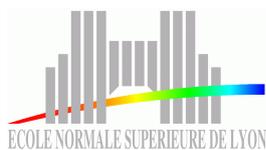


Table des matières

1	Introduction	6
1.1	Systèmes complexes	6
1.1.1	Définition	7
1.1.2	Modéliser les systèmes complexes	9
1.2	Perturbations et robustesse	10
1.2.1	Perturbations	10
1.2.2	Robustesse	10
1.2.3	Contrôle	11
1.3	Le modèle des automates cellulaires	11
1.3.1	Fini ou infini	13
1.4	Simulation	14
1.5	Plan de la thèse	14
2	Notions utilisées	16
2.1	Automates cellulaires	16
2.1.1	Asynchronisme	19
2.2	Transitions de phase et percolation dirigée	20
I	Couplages et percolation dirigée	24
3	Couplage par forçage	28
3.1	Le modèle	29
3.1.1	Lien avec un autre modèle	31
3.2	Étude expérimentale	31
3.2.1	Classification des automates élémentaires	31
3.2.2	Percolation dirigée	33
3.3	Conclusion et perspectives	38
4	Couplage par les mises à jour	40
4.1	Étude formelle	41
4.1.1	Preuves de non-coalescence rapide	41
4.1.2	Preuves de coalescence rapide	44
4.2	Étude expérimentale et transition de phase	47
4.2.1	Classification des automates vis-à-vis de la coalescence	47
4.2.2	Percolation dirigée dans ce modèle	51
4.2.3	Comparaison à un modèle plus simple	53
4.3	Conclusion et perspectives	56

5	Équivalence entre asynchrones et certains probabilistes	58
5.1	Définitions	59
5.1.1	L'automate de Domany-Kinzel	61
5.1.2	Simulation	61
5.2	Simulation d'un automate asynchrone par un automate probabiliste	66
5.3	Simulation d'un automate probabiliste par un automate asynchrone	67
5.3.1	Mesure d'un bit aléatoire	67
5.3.2	Synchronisation	71
5.3.3	Construction complète	73
5.3.4	Autres constructions	74
5.4	Conclusion et perspectives	75
II	Perturbations de la topologie	78
6	Perturbation de la topologie en 1D	82
6.1	Protocole expérimental	82
6.2	Étude exhaustive des automates élémentaires	84
6.2.1	Règles insensibles aux deux perturbations	85
6.2.2	Règles sensibles seulement à la perturbation du synchronisme	85
6.2.3	Règles sensibles seulement à la perturbation de la topologie	88
6.2.4	Règles sensibles aux deux perturbations	88
6.3	Conclusion et perspectives	90
7	Perturbations de la topologie pour la règle minorité	92
7.1	Le modèle : définitions	93
7.1.1	Topologie	93
7.1.2	Énergie et Particules	94
7.2	Graphes réguliers	95
7.2.1	Cycles	95
7.2.2	Graphe complet	96
7.3	Arbres	97
7.3.1	La règle duale	97
7.4	Structure de l'ensemble limite sur les arbres	99
7.4.1	Un algorithme pour l'ensemble limite	99
7.4.2	Structure	100
7.5	Temps transitoire sur les arbres	104
7.5.1	Arbres de degré maximum au plus 3	104
7.5.2	Arbres de degré maximum 4 ou plus	107
7.6	Conclusion et perspectives	112
8	Conclusion	114
	Bibliographie et tables	116

Remerciements

Mon directeur de thèse pour son enthousiasme redoutablement communicatif, pour m'avoir poussé quand j'étais peu convaincu de l'utilité de certaines pistes de recherche, pour les rencontres qu'il a su catalyser.

Les relecteurs pour le travail qui les attend au moment où ils lisent ces mots, et la rapidité avec laquelle ils ont accepté de donner un avis sur cette thèse.

Les membres du jury pour s'être rendus disponibles en cette période chargée, et pour l'intérêt qu'ils portent à ce travail.

Les autres doctorants du laboratoire pour m'avoir montré la voie ou pour avoir participé à la course de qui finirait de rédiger le premier (ils ont gagné), et pour les moments moins productifs (disons, les moments créatifs).

Les membres du laboratoire pour leur accueil, leur disponibilité face à mes questions saugrenues et leur empressement à y répondre, pour leur introduction aux coutumes de la communauté des chercheurs.

Ma famille pour sa confiance malgré le peu d'explications de ma part, elle va enfin avoir tous les détails sur mon travail.

Le public nombreux lors de la soutenance : amis, famille, collègues et potentiels futurs collègues (i.e. étudiants) pour être venu ou s'être excusé de ne pas pouvoir venir.

Et tous ceux que j'oublie (c'est un peu tricher de terminer ainsi, mais je ne fais plus confiance à ma mémoire), avec mes excuses.

Navigation Pour faciliter la navigation dans le document, beaucoup de références ont été introduites. Notamment dans la bibliographie, les derniers chiffres sont les numéros de page où l'article est cité. Lors de la lecture du PDF à l'écran, tous les liens (numéros de figures ou de page, références, URL...) sont cliquables. Un bouton ou le raccourci Alt+gauche dans Adobe Reader (b dans xpdf) permet de revenir à la page où l'on était avant de cliquer sur un lien interne.

Chapitre 1

Introduction

Cette thèse se place dans le cadre de l'étude des systèmes complexes. Nous commençons donc par présenter¹ cette discipline à la section 1.1 avec deux définitions complémentaires de ce qu'est un système complexe. Précisément, nous nous intéressons aux perturbations des systèmes complexes, dont les questions associées et l'intérêt sont présentés à la section 1.2. Nous expliquons en particulier ce que représente une perturbation et ce qu'elle permet d'apprendre. Nous introduisons alors à la section 1.3 le modèle utilisé pour répondre à ces questions : les automates cellulaires. Nous exposons les différentes utilisations de ce modèle et pourquoi nous l'avons choisi. Ce chapitre se termine par un mot sur notre logiciel de simulation et le plan de la thèse.

1.1 Systèmes complexes

De nombreux systèmes sont constitués d'un grand nombre d'entités en interaction. On les qualifie de complexes lorsque, même si les interactions entre ces entités sont bien comprises, il émerge un comportement global qui n'était pas évident à partir des règles d'interaction. En d'autres termes, lorsque l'on veut modéliser ce système, on conçoit un certain nombre de règles d'évolution, puis l'on simule le système en itérant ces règles jusqu'à obtenir un résultat structuré. Un système est informellement dit complexe si le résultat final n'est pas prédictible directement en connaissant les règles. Ce n'est pas une limite due à la longueur des calculs (ce point est le champ d'étude de la théorie de la complexité « computationnelle »), ni d'un comportement aléatoire, ou de la sensibilité aux conditions initiales (qui est étudiée par la théorie du chaos), même si l'étude de ces propriétés sur un système complexe est intéressante. Cette limitation vient de l'impossibilité de modéliser le système par des équations solvables et prédictives. Ce qui est primordial est le nombre de paramètres, et le fait que chacun d'entre eux peut avoir une influence essentielle sur le comportement du modèle. Pour prévoir ce comportement, il est nécessaire de tous les prendre en compte, ce qui revient à effectuer une simulation du système étudié.

Une certaine confusion peut apparaître dans le langage courant entre complexe et compliqué. Étymologiquement, compliqué signifie qu'il est difficile de comprendre l'objet d'étude, alors que complexe signifie qu'il y a beaucoup d'intrications, que « tout est lié » et que l'on ne peut étudier une petite partie du système de façon isolée. Les systèmes complexes sont généralement compliqués, mais le contraire n'est pas toujours vrai.

¹La description des systèmes complexes qui suit ressemble à celle que l'on pouvait trouver sur [Wikipédia](#) à l'été 2008. Ce n'est pas un plagiat, car j'ai largement contribué à cette page de Wikipédia !

Les systèmes complexes apparaissent dans de nombreux domaines, comme le montrent les exemples de la table 1.1.

Entités	Interactions	Comportement émergent
grains de sable	collisions	avalanches
gènes	activation et inhibitions	points fixes ^a
protéines	réactions chimiques	vie d'une bactérie
neurones	impulsions électriques	pensée
fourmis	phéromones	fourmilière ^b
étourneaux	évitements et alignement	essaim
investisseurs	transactions	bulle financière ^c
personnes	rencontres	propagation d'épidémies ou de rumeurs

^aUn ensemble de gènes exprimés définit un tissu : les gènes activés et inhibés ne sont pas les mêmes dans les cellules de la peau ou dans celles d'un muscle.

^bUne fourmi n'a pas assez de neurones pour connaître le plan de la fourmilière, pourtant une fourmilière émerge des interactions.

^cLa majorité des agents boursiers ne souhaitent pas la crise. Pourtant ce comportement émerge.

TAB. 1.1 – Quelques exemples de systèmes complexes

1.1.1 Définition

Il n'existe pas de définition précise largement acceptée de ce qu'est un système complexe. Afin de néanmoins définir notre sujet d'étude, voici deux définitions, l'une par la constitution du système, l'autre par son comportement. L'un des buts de l'étude des systèmes complexes est de préciser dans quels cas ces deux définitions sont équivalentes.

Constitution

Nous considérons qu'un système est complexe s'il a les caractéristiques suivantes :

- Il est composé d'un grand nombre, fini, d'entités. Typiquement entre 50 et un million.
- Un comportement global structuré émerge.
- Les interactions sont locales, de même que la plupart des informations, il y a peu d'organisation centrale.
- Le graphe d'interaction est non trivial : ce n'est pas simplement tout le monde qui interagit avec tout le monde (il y a au moins des liens privilégiés). On peut noter que ce réseau est souvent intéressant en lui-même et présente des caractéristiques telles que petit monde, invariance d'échelle, communautés...
- Il y a des boucles de rétroaction (en anglais *feedback*) : l'état d'une entité a une influence sur l'état d'autres entités, qui en retour ont une influence sur la première.

Afin de préciser cette définition, voici trois exemples de disciplines proches.

- Selon certains les *systèmes multi-agents* sont un cas particulier de système complexe. Nous souhaitons au contraire distinguer ces deux disciplines :
 - Le champ des systèmes multi-agents cherche généralement à créer des comportements spécifiques, à *concevoir* plutôt qu'à *comprendre* ou *prédire* le comportement d'un système imposé. Un système multi-agents a donc une tâche assignée, ce qui est rarement le cas d'un système complexe.

- Il est possible d'identifier chaque agent et de forcer une organisation prédéfinie, alors qu'un système complexe a généralement toutes ses entités interchangeables et définies à partir de quelques paramètres.
- L'environnement où évoluent les agents est important et fait partie du système : les interactions peuvent parfois ne se faire que via la modification de l'environnement (par exemple le dépôt de phéromones sur le sol), ce qui est appelé stigmergie. Au contraire dans un système complexe, l'environnement est extérieur et a une influence à travers les frontières.
- Un agent peut avoir un comportement évolué tandis que dans un système complexe les règles d'une entité sont simples.
- Les *réactions chimiques* que l'on apprend jusqu'en licence ne sont pas complexes malgré le grand nombre de molécules et d'interactions en jeu, car elles sont parfaitement prédites par quelques équations. La taille du système n'est donc pas un critère de complexité. Cependant, il existe aujourd'hui des systèmes complexes en chimie, par exemple dans les travaux de [Lehn (1999), Lehn (2000), Lehn (2002)].
- La *physique statistique* étudie des particules extrêmement simples, comme les atomes d'un gaz. Elle s'intéresse au comportement limite lorsque le nombre de particules tend vers l'infini, alors qu'un système complexe a toujours un nombre fini d'éléments (même s'il est parfois utile de le modéliser par des équations continues). L'entropie n'est donc pas non plus un critère de complexité : un gaz a une forte entropie car un nombre gigantesque de configurations possibles pour les atomes le constituent, et pourtant il n'est pas complexe : son comportement est simple à décrire, à l'aide d'outils statistiques.
De plus, les entités d'un système complexe peuvent être elles-mêmes des systèmes complexes, faisant ainsi apparaître une hiérarchie.

Comportement

On peut aussi définir les systèmes complexes d'après leur comportement. Un système est complexe s'il présente la plupart des comportements suivants.

- Il y a auto-organisation et émergence de propriétés ou de structures cohérentes, apparition de motifs. Le système a donc une mémoire de son évolution, son état influence les états futurs.
- Le système présente une robustesse locale et fragilité à moyenne échelle : puisqu'il y a de nombreux liens (éventuellement créés ou remaniés par le système lui-même), si un élément est affecté par un événement extérieur, ses voisins le seront aussi. Il s'ensuit que le système est souvent plus robuste à une petite perturbation locale qu'il ne le serait sans les liens. Mais c'est grâce à cette même interdépendance des constituants que modifier globalement le système (et donc potentiellement le contrôler) peut être fait grâce à une perturbation moins grande que dans le système sans liens. Bien cibler cette perturbation est cependant très difficile. Les virus (issus d'une longue sélection naturelle) sont un bon exemple : avec une dizaine de gènes, un virus est capable de modifier profondément (jusqu'à la mort...) un organisme de plusieurs dizaines de milliers de gènes, et ce même avec une faible concentration au départ.
- On observe une brisure de symétrie : la connaissance d'une partie du système ne permet pas d'affirmer que le reste du système est en moyenne dans le même état.
- Plusieurs comportements possibles sont en compétition, certains sont simples, d'autres chaotiques (désordonnés). C'est souvent lorsque le système est à la fron-

tière entre les deux et alterne ces deux types de comportement qu'il est le plus intéressant.

- Plusieurs échelles temporelles et spatiales apparaissent, il y a ainsi une hiérarchie de structures. Les entités peuvent être elles-mêmes des systèmes complexes : une société est composée d'humains, eux-mêmes composés de cellules.
- Le système peut être ouvert et soumis à un extérieur, il y a des flux d'énergie et d'information sur la frontière. Le système dissipe donc de l'énergie et n'est pas à l'équilibre. Cette frontière peut être floue (par exemple, en considérant le système complexe « un humain », à partir de quel instant la nourriture ou l'air absorbés font-ils partie du corps ?).

1.1.2 Modéliser les systèmes complexes

Voici la position de notre travail dans la problématique générale de modélisation des systèmes complexes. Lors de la modélisation de tels systèmes², il nous semble utile de distinguer quatre objectifs que l'on peut poursuivre.

- Le premier est de *comprendre* certains phénomènes. On construit pour cela un modèle du phénomène et on le compare aux observations.
- On cherche ensuite à les *prédire*. Une méthode consiste à utiliser le modèle que l'on vient de construire.

Mais cet ordre n'est aujourd'hui plus toujours respecté : il est des phénomènes que l'on prédit avant de comprendre, ou bien que l'on peut prédire à l'aide d'algorithmes d'apprentissage automatique dont les paramètres n'ont pas de sens concret facile à exprimer. Réciproquement, il existe des systèmes chaotiques qui sont bien compris mais impossibles à prévoir à long terme.

- Quand on sait prédire l'évolution d'un système, il est souvent utile d'influer sur ce système. On peut prédire l'influence de chaque modification apportée et donc choisir celle qui donne le comportement le plus intéressant. C'est *contrôler* le système.
- Enfin, on cherche à *concevoir* d'autres systèmes à partir de notre compréhension, d'évolution souhaitée du système, et de la facilité à le contrôler.

Cette thèse se place dans la phase amont, c'est-à-dire dans la compréhension de systèmes complexes, même si nous ferons parfois quelques remarques sur les autres axes.

Face à un système complexe particulier, on cherche souvent à en construire un modèle qui reproduise sa constitution, dans le but de reproduire le comportement du système en le simulant par le modèle. On peut aussi bien chercher un modèle précis afin de *prédire* quantitativement le comportement du système que chercher un modèle simple qui reproduise qualitativement le système, afin de *comprendre* le système et d'identifier les mécanismes de base produisant ce type de comportement. La première approche concerne certains scientifiques aux prises avec un système complexe d'un domaine précis, tandis que la seconde est courante chez ceux qui cherchent des méthodes et définitions générales applicables à de nombreux systèmes différents. Le présent travail contribue à la seconde approche : nous utilisons des modèles génériques afin d'identifier des concepts généraux potentiellement utiles dans de nombreux systèmes complexes, les raisons des phénomènes observés et les ingrédients minimaux pour les reproduire. Nous cherchons donc des modèles simples car ils sont plus facile à comprendre et laissent espérer des résultats génériques. Ce n'est bien sûr qu'une aide, un point de vue différent sur un

²Ces questions peuvent parfois être appliquées plus généralement à la modélisation en sciences dures.

problème : l'étude d'un système particulier ne peut se faire qu'avec des spécialistes du domaine. Les deux approches sont complémentaires et l'étude des systèmes complexes est donc interdisciplinaire.

1.2 Perturbations et robustesse

Dans le cadre des systèmes complexes, nous nous intéressons plus précisément aux perturbations et à la robustesse à ces perturbations. Cette section présente l'origine et l'intérêt de ces questions puis détaille ce que nous entendons par robustesse.

1.2.1 Perturbations

Une modélisation n'est jamais qu'une idéalisation, une approximation du système réel. On peut espérer appréhender la différence entre modèle et système comme l'ajout d'une perturbation au modèle, notamment pour étudier les différences de comportement. Les questions typiques sont donc les suivantes. La première question concerne le modèle : l'évolution est-elle fortement modifiée si l'on dispose d'un modèle un peu plus précis ? L'approximation consistant à ignorer l'extérieur, à isoler le système, est-elle justifiée ? D'autres questions concernent le système : si le système n'évolue pas de façon isolée, quel est l'effet de petites perturbations externes ? Si certains éléments du système disparaissent ou ne fonctionnent plus comme prévu, le comportement global change-t-il, et de combien ?

Une discipline qui a étudié ces questions avec succès est la théorie du chaos. Elle s'intéresse aux perturbations des conditions initiales dans des systèmes déterministes. Ce travail cherche à appréhender d'autres formes de perturbations, notamment celles se produisant après l'initialisation.

1.2.2 Robustesse

Étudier l'effet de perturbations sur un système, c'est étudier les réactions à ces perturbations. Dès que l'on cherche à prédire ces réactions, à savoir si le système ne va quasiment pas changer de comportement ou au contraire va être profondément modifié, c'est ce que nous appelons étudier la *robustesse* du système.

Les systèmes réels n'étant pas aussi réguliers et parfaits (i.e. sans défaut) que les modèles, un bon modèle doit donc être robuste à de petits défauts ou variations locales. En d'autres termes, un modèle est toujours une simplification du système réel, il fait donc des approximations. Ces petites différences entre le modèle et la réalité peuvent être vues comme des perturbations. Bien sûr, on souhaite que le modèle ne soit pas dépendant de certains choix arbitraires effectués pendant la modélisation, un bon modèle doit donc être robuste à ces perturbations. Par exemple, pour un système réel qui (comme la plupart) n'a pas de mécanisme global de synchronisation entre les constituants, il faut vérifier que les différentes façons de choisir l'ordre dans lequel les composants du système sont mis à jour n'influent pas sur le comportement global. Si le modèle est robuste aux choix de modélisation, les perturbations peuvent alors donner des enseignements sur le système modélisé, et non plus sur le modèle.

Un système peut être robuste et avoir toujours le même comportement quelle que soit la perturbation appliquée. Il peut au contraire être très sensible et changer radicalement à la moindre modification des paramètres. Un cas plus intéressant est celui

des systèmes robustes à un grand nombre de perturbations, mais sensibles à d'autres perturbations. Le premier point assure qu'ils ont une existence durable même confrontés à un extérieur, mieux, qu'ils ont un comportement prédictible si on les protège de certaines perturbations. Le second point peut les rendre contrôlables, ou bien adaptatifs à un environnement qui évolue.

1.2.3 Contrôle

On peut parfois choisir les perturbations que l'on applique au système. On les choisit alors dans le but d'obtenir un comportement particulier du système, on cherche donc à le *contrôler*. Par exemple, on vise à rapprocher le système d'une évolution connue. La trajectoire des satellites est ainsi contrôlée grâce à d'infimes perturbations soigneusement choisies.

On étudie alors si la réponse aux perturbations est toujours la même. Ou bien s'il existe des perturbations moins coûteuses que d'autres pour obtenir un même résultat, ou de façon équivalente si l'on peut amener le système dans un état donné avec une limite sur la force des perturbations que l'on peut appliquer.

1.3 Le modèle des automates cellulaires

Nous utilisons les automates cellulaires comme modèle de système complexe. Nous choisissons ce modèle car il est bien formalisé, et sa simplicité permet d'espérer des résultats théoriques et de portée générique. Ce modèle présente également une grande souplesse : il est possible de modifier la définition d'un ingrédient du modèle sans modifier les autres. On peut donc étudier l'effet spécifique d'une ou deux perturbations. La gamme des perturbations applicables est de plus très large.

Une définition est donnée à la section 2.1 ; informellement, un automate cellulaire consiste en une ligne ou un grille de cellules, chaque cellule pouvant être dans un nombre fini d'états (souvent représentés par des couleurs). À chaque pas de temps, les cellules calculent leur nouvel état en fonction des états des cellules voisines. Toutes les cellules utilisent la même fonction et sont mises à jour en même temps.

Les automates cellulaires ont tout d'abord été, et sont encore aujourd'hui, très étudiés comme modèle de calcul (voir par exemple [Kari (2005)] qui passe en revue cette approche des automates cellulaires). La question fondamentale de ce domaine est « que peut-on calculer avec un automate cellulaire ? ». Plusieurs problèmes ont ainsi été posés et résolus : *firing squad* (synchronisation d'une ligne de cellules à partir des interactions locales), calcul des nombres premiers, construction de diverses formes géométriques, simulation d'autres modèles de calcul... De façon complémentaire, des résultats d'indécidabilité ont été montrés, ainsi par exemple il ne peut pas exister d'algorithme pour décider dans tous les cas d'une propriété aussi simple que la nilpotence (i.e. si le calcul global mène toujours à une configuration où toutes les cellules sont dans le même état).

Dans une approche de type système dynamique, on a cherché à classifier les automates cellulaires, d'abord par une simple observation visuelle [Wolfram (1984)] puis de façon formelle. Une revue de l'approche des automates cellulaires en tant que systèmes dynamiques est disponible dans [Blanchard, Kůrka, and Maass (1997)]. Une introduction aux automates cellulaires (déterministes et probabilistes) du point de vue de la physique (exposant de Lyapunov, transitions de phase et percolation dirigée, propagation d'erreur, diffusion, méthodes de simulation in silico, méthode du champ moyen...) se trouve dans les notes de cours [Bagnoli (1998)].

Les automates cellulaires sont enfin étudiés de façon plus appliquée comme paradigme de modélisation, notamment en biologie, sociologie, physique, etc. Voir à ce sujet la seconde partie de [Ganguly, Sikdar, Deutsch, Canright, and Chaudhuri (2003)] où l'on trouve des références vers un certain nombre d'articles de modélisation par automate cellulaire dans plusieurs domaines, ou de [Bandini, Mauri, and Serra (2001)] qui détaille quelques projets fructueux de modélisation par automate cellulaire.

La grande majorité des travaux évoqués ci-dessus utilisent un modèle très régulier : dynamique synchrone (toutes les cellules sont mises à jour en même temps), cellules disposées sur une ligne ou sur une grille 2D, règle de mise à jour uniforme, etc. Comme annoncé à la section 1.2.1, nous allons introduire des modifications à ce modèle pour en étudier la robustesse. Ces perturbations ont été peu étudiées (leur analyse théorique est d'ailleurs difficile), mais voici quelques travaux sur le sujet.

Perturbation de la règle de mise à jour Historiquement, les premières études de perturbations d'automates cellulaires semblent être les travaux de [Toom (1974)], [Toom (1980), Gács (1986), Toom et al. (1990)]. L'idée initiale est de considérer un automate cellulaire où à chaque transition (à chaque calcul élémentaire d'une cellule), il y a une probabilité ε de faire une erreur (de ne pas respecter la règle) et que la cellule passe dans un état imprévisible. Ces auteurs ont alors cherché à construire des règles permettant de résister à ces erreurs (de mener à bien le calcul), pourvu qu' ε soit suffisamment petit. Toom a proposé une règle très simple, qui permet sur une grille 2D de stocker un bit d'information (ce qui peut sembler peu). Cette construction a alors été utilisée pour construire un automate cellulaire 3D permettant de faire des calculs fiables même en présence d'erreur, [Gács and Reif (1988)], résultat qui a ensuite été amélioré pour obtenir un automate cellulaire 1D subtil (il s'auto-organise en une hiérarchie) lui aussi capable de calculs universels [Gács (2001)]. Les preuves de ces résultats sont particulièrement compliquées.

Perturbation de la dynamique Ici il s'agit de ne plus mettre à jour toutes les cellules en même temps; il y a pour cela principalement trois possibilités : les cellules sont mises à jour une par une (plusieurs ordres éventuellement aléatoires sont possibles), chaque cellule a une probabilité de se mettre à jour, on définit des blocs fixes de cellules qui sont mises à jour en même temps. Ces trois possibilités de dynamique asynchrone sont appelées respectivement dynamique séquentielle (ou totalement asynchrone), dynamique partiellement asynchrone, dynamique bloc-séquentielle. [Ingerson and Buvel (1984), Schönfisch and de Roos (1999)] ont identifié expérimentalement l'intérêt d'étudier une dynamique asynchrone. Ont suivi des études expérimentales de la résistance à l'asynchronisme [Bersini and Detours (1994), Kanada (1994), Fatès (2004)], puis des analyses mathématiques d'une classe d'automates cellulaires à une dimension [Fukš (2004)], [Fatès, Regnault, Schabanel, and Thierry (2006)].

Perturbation de la topologie La topologie est ici la définition de l'espace et la spécification des voisinages : un composant ne peut interagir qu'avec ses voisins. Sur ce point, il semble n'exister que quelques études expérimentales : [Serra and Villani (2002)] étudient la règle majorité (en dynamique totalement asynchrone) sur une ligne avec quelques liens redirigés vers des cellules aléatoires, le graphe résultant étant un modèle de petit-monde; [Mamei, Roli, and Zambonelli (2005)] adoptent le point de vue du contrôle; tandis que

[Fatès and Morvan (2004)] montrent que couper quelques liens rend le jeu de la vie plus robuste : la dynamique partiellement asynchrone est plus proche de la dynamique synchrone que si on n'avait pas coupé de liens.

Nous perturberons également la dynamique, mais de façon aléatoire, contrairement aux travaux de Toom et Gács qui montrent une robustesse même face aux pires erreurs possibles. Nous perturberons la dynamique dans la lignée des travaux présentés. Enfin, nous perturberons la topologie d'abord par de petites modifications à une grille, puis en considérant une topologie plus exotique, dans les deux cas en dynamique asynchrone (donc en combinant deux perturbations).

1.3.1 Fini ou infini

Nous utiliserons parfois des modèles finis et parfois des modèles infinis, aussi bien en ce qui concerne le temps (le nombre d'étapes de calcul) que l'espace (le nombre de cellules). Voici les enjeux de ces choix.

Espace

Les systèmes complexes ne font intervenir qu'un nombre fini d'entités. C'est le cas de certains automates cellulaires utilisés en modélisation. Nous utilisons donc dans ce travail le plus souvent des configurations finies.

Comme l'ont montré par exemple [Fatès, Regnault, Schabanel, and Thierry (2006)] de façon analytique, certains comportements sont fortement dépendant du caractère fini ou non des configurations, et le comportement dans un modèle ne peut pas toujours être déduit de celui dans l'autre modèle. Il nous semble donc que ce modèle fini est également intéressant et mérite d'être étudié pour lui-même. La finitude est d'ailleurs nécessaire pour définir des temps de convergence comme nous en manipulons aux chapitres 4 ou 7. Une généralisation de ces concepts aux configurations infinies serait sans doute intéressante pour de futurs travaux.

Cependant, les automates cellulaires ont été beaucoup étudiés sur des configurations non bornées (avec un nombre infini de cellules), en particulier dans leur utilisation en tant que modèle de calcul. Certains résultats de cette thèse s'appliquent également au cas des configurations infinies, notamment les résultats concernant la percolation dirigée. Nous vérifierons donc aux chapitres 3 et 4 que varier la taille ne perturbe pas les résultats, et que les restrictions induites par l'utilisation de configurations finies dans les calculs n'interdisent pas la généralisation des résultats aux configurations infinies.

Temps

Les limites de calcul imposent également que le temps de simulation soit fini, et même interdisent qu'il augmente exponentiellement par rapport au nombre de cellules. Cela introduit certaines contraintes sur ce que l'on peut observer ou non. Mais comme nous allons le voir, ce que l'on observe en pratique peut être autant pertinent.

Si l'on pouvait examiner le système «après un temps infini», on observerait le régime asymptotique, qui est l'ensemble des états où se trouve le système après un temps arbitrairement long. Le régime transitoire est l'évolution du système avant d'atteindre ce régime asymptotique.

Les seuls régimes que l'on puisse atteindre sur des systèmes réels, et dans les simulations, se produisent en temps polynomial. Lorsque la convergence se fait en temps exponentiel, il n'est donc pas possible d'observer expérimentalement le régime asymptotique. Dès que le temps transitoire augmente exponentiellement avec la taille du système, l'attracteur n'est donc en pratique jamais atteint. Le régime examiné est alors un « long transitoire », par opposition à un court régime transitoire apparaissant juste après l'initialisation. Dans ce régime, certains observables ont des valeurs stables, notamment la densité que nous allons largement utiliser. Le système reste dans ce « long transitoire » pour un temps plus que polynomial, puis atteint le vrai régime asymptotique. C'est ce comportement transitoire qui nous intéresse ici ; ce ne sont que des fluctuations autour de ce comportement qui font que le système touche l'ensemble absorbant « final ». On peut comparer cela aux systèmes vivants où le vrai régime asymptotique est la mort, mais où la plupart des études se concentrent sur le « long transitoire » et les fluctuations autour de leur état vivant.

Nous verrons dans les chapitres 3 et 4 un phénomène où le système atteint l'ensemble limite en un temps soit « court », soit exponentiel par rapport à la taille du système. La distinction entre les deux cas se fait selon un paramètre du modèle. Si on laisse le système évoluer pendant un temps suffisamment long (et donc parfois exponentiel), le système converge toujours vers cet ensemble absorbant : il n'y a qu'un seul régime asymptotique. Mais dans le cas où cette convergence prend un temps exponentiel, elle ne peut pas être observée. On peut donc distinguer deux comportements différents du système.

1.4 Simulation

Les questions que nous venons de présenter sont étudiées ici à la fois de façon théorique et à l'aide de simulations. Tout au long de ce travail, nous avons développé un outil de simulation permettant de perturber indépendamment les différents constituants d'un automate cellulaire (configuration initiale, dynamique, règle de transition, topologie...) et d'intégrer de nouvelles idées facilement. En effet, il n'existait pas à notre connaissance d'outils permettant de réaliser ces deux opérations. (Nous avons commencé par utiliser celui de Nazim Fatès, disponible sur <http://nazim.fates.free.fr/Logiciel.htm>, mais nous avons besoin de stocker et échantillonner facilement les paramètres, et surtout nous avons besoin d'une exécution plus rapide que ce que permet Java.) Ce logiciel n'a cessé de s'enrichir au fur et à mesure des besoins qui apparaissaient. Il compte maintenant un peu plus de 5 000 lignes de code en Caml (noter qu'un programme en Caml est typiquement 4 fois plus compact que son équivalent en C). Le code source est disponible sur <http://cimula.sf.net>.

1.5 Plan de la thèse

Les enjeux et résultats de chaque chapitre seront bien sûr présentés dans leur introduction et rassemblés dans les conclusions, contentons-nous ici de donner une vue d'ensemble de cette thèse sous l'angle des perturbations appliquées. Chacun des ingrédients d'un modèle que l'on définit est un point où l'on peut étudier les perturbations :

- Une première forme de perturbation de la règle locale fait l'objet du chapitre 3. On y trouve un modèle de percolation dirigée et la discussion d'une conjecture de Grassberger. Le modèle utilisé étudie dans quels cas il est possible de forcer deux instances d'un système à devenir identiques.

- Un effet de l’asynchronisme est étudié au chapitre 4. Dans ce chapitre, le phénomène de *coalescence* observé par Fatès est formalisé. Nous démontrons les premières propriétés sur cette classe d’automates, établissons expérimentalement que l’on y rencontre à nouveau la percolation dirigée, puis proposons quelques interprétations en termes de systèmes dynamiques et de simulations numériques.
- Le chapitre 5 cherche une validation théorique des phénomènes de percolation dirigée observée dans les chapitres 3 et 4. Pour cela nous proposons une notion de simulation préservant la densité asymptotique et montrons une équivalence selon cette notion entre les automates asynchrones déterministes et une classe d’automates probabilistes. Cette construction repose sur la lecture d’un bit aléatoire dans la suite des mises à jour, lecture effectuée par l’automate.
- Les perturbations de topologie sont le sujet de la partie II. Le chapitre 6 étudie une perturbation en une dimension consistant à couper quelques liens. Cette perturbation est dynamique, c’est-à-dire qu’elle évolue avec le temps. Nous étudions les interactions de cette perturbation avec la perturbation du synchronisme. Nous observons des comportements non linéaires voir non monotones, une perturbation rendant parfois robuste à l’autre perturbation. Ici encore, il semble qu’il y ait une transition de phase.
- Le chapitre 7 compare différentes topologies pour une règle fixée, de façon analytique. La comparaison s’effectue sur la structure de l’ensemble limite et l’espérance du temps pour atteindre ce dernier. L’étude porte en particulier sur les arbres où il apparaît des comportements délicats à caractériser. Nous observons également un seuil : pour un degré maximal au moins égal à 4, le temps de convergence peut être exponentiel, tandis que pour un degré maximal au plus 3, il est polynomial.

Chapitre 2

Notions utilisées

Ce chapitre présente les notions que nous utiliserons dans plusieurs chapitres de la thèse. Les notions spécifiques à un chapitre seront présentées au moment où elles seront nécessaires. Nous présentons ici les automates cellulaires et leur version asynchrone, puis les transitions de phase et la percolation dirigée.

2.1 Automates cellulaires

Définition 1. Un *automate cellulaire* est un vecteur (Q, d, V, δ) où

- Q est un ensemble fini, l'ensemble des *états* ;
- $d \in \mathbb{N}^*$ est la *dimension* ;
- $V = \{v_1, \dots, v_{|V|}\}$, le *voisinage*, est un ensemble fini de vecteurs de \mathbb{Z}^d ;
- $\delta : Q^{|V|} \rightarrow Q$ est la *règle de transition*.

Pour alléger le discours, nous écrivons désormais « automate » au lieu de « automate cellulaire » en dehors des définitions, résultats, introductions et conclusions.

L'ensemble des *cellules* est soit $\mathcal{U} := \mathbb{Z}^d$ pour un automate infini, soit $\mathcal{U} := (\mathbb{Z}/n\mathbb{Z})^d$ pour un *automate cellulaire fini* (i.e. ayant un nombre fini de cellules), où n est la taille de l'automate. Nous utiliserons le plus souvent des automates à une dimension, pour lesquels n sera donc le nombre de cellules.

Certains préfèrent utiliser le terme au pluriel (automates cellulaires) en considérant que chaque cellule est un automate. C'est en effet plus naturel lorsque l'on généralise la topologie : si l'on n'utilise plus la grille régulière \mathbb{Z}^d ou $(\mathbb{Z}/n\mathbb{Z})^d$ mais un autre graphe, par exemple un arbre ou un graphe de Cayley, on parle alors de *réseau d'automates*.

Une *configuration* définit l'état de chaque cellule, c'est donc une fonction $c : \mathcal{U} \rightarrow Q$. On utilise souvent des configurations périodiques. En une dimension, la configuration a^* dénote le mot a répété à l'infini, ou bien autant de fois que nécessaire pour atteindre n cellules. Par exemple, $(01)^*$ est la configuration 01010101... tandis que 0^* est la configuration ne contenant que des 0.

L'évolution d'un automate cellulaire est la suivante. Étant donnée une configuration c , la configuration suivante c' est obtenue en appliquant simultanément¹ la règle de transition δ à toutes les cellules :

$$c'(z) := \delta(c(z + v_1), \dots, c(z + v_{|V|}))$$

On étend la notation δ aux configurations en posant $\delta(c) := c'$.

¹Nous verrons le cas asynchrone à la section 2.1.1. Nous présentons ici le modèle le plus utilisé.

Une partie finie d'une configuration c est un *motif*. C'est donc une restriction de la fonction c à un sous-ensemble fini de \mathcal{U} . En une dimension, si le sous-ensemble est un intervalle, un motif est une suite finie d'états (encore appelée *mot*). C'est le seul cas que nous utiliserons.

Un motif de la forme $c(z+v_1), \dots, c(z+v_{|V|})$ est une *configuration locale* centrée en z . δ est donc une fonction des configurations locales vers les états.

Un motif q_1, \dots, q_a est un *antécédent* d'un motif q'_1, \dots, q'_b si $b = a - |V| + 1$ et pour tout entier i de l'intervalle $\llbracket 1, b \rrbracket$, $\delta(q_i, \dots, q_{i+|V|}) = q'_i$:

$$\underbrace{q'_1, q'_2, \dots}_{q_1, \dots, q_{|V|}}, q_{|V|+1}, \dots, \underbrace{\dots, q'_b}_{q_{a-|V|+1}, \dots, q_a}$$

Définition 2 (diagramme espace-temps). Un diagramme espace-temps est une suite de configurations. C'est donc une fonction $c(\cdot, \cdot) : \mathcal{U} \times \mathbb{N} \rightarrow Q$, où $c(x, t)$ est l'état de la cellule x à l'instant t .

Nous emploierons parfois le terme de diagramme espace-temps pour parler d'une suite de configurations dont nous n'avons pas encore prouvé qu'elle respecte la règle d'un automate cellulaire. On parle au contraire d'*orbite* lorsque le diagramme espace-temps respecte les règles d'un automate. Une orbite est donc intuitivement un « calcul » effectué par l'automate, avec comme entrée la configuration initiale.

Un diagramme espace-temps est la façon habituelle de représenter l'orbite d'un automate à une dimension. On utilise alors deux dimensions : la seconde dimension est utilisée pour le temps. On empile les configurations : chaque configuration est tracée horizontalement au-dessus de la précédente. Le sens dans lequel on juxtapose les configurations dépend en réalité de l'auteur, nous utiliserons le sens « de bas en haut » assez répandu, ou parfois le sens « de gauche à droite » pour des raisons de mise en page, en le précisant chaque fois que nécessaire. La figure 2.1 donne un exemple de diagramme espace-temps.

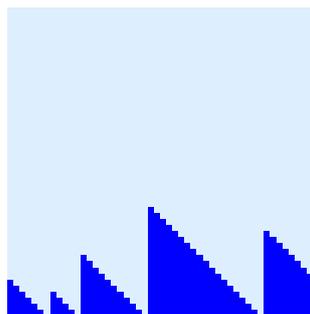


FIG. 2.1 – Exemple de diagramme espace-temps. La règle est ici 136 (cf. définition 3), qui signifie « passer dans l'état 0 si mon voisin de droite est dans l'état 0. ». Le temps va de bas en haut.

Nous utiliserons parfois des paires de configurations évoluant en même temps et avec la même règle, le but étant d'étudier les différences entre les configurations. Un diagramme espace-temps est toujours adapté pour représenter ces orbites. Précisément, nous superposons le diagramme espace-temps de chacune des configurations, avec le code couleur suivant (nous n'utiliserons que le cas où il y a deux états : 0 et 1) :

- clair (blanc ou bleu clair) signifie que la cellule est dans le même état (respectivement 0 ou 1) dans les deux configurations ;
- foncé (bleu foncé ou violet) signifie que la cellule est dans un état différent entre les deux configurations (il y a deux possibilités : (0, 1) et (1, 0)).

Le dessin résultant est appelé un *diagramme espace-temps d'accord*. La figure 2.2 en est un exemple.



FIG. 2.2 – Diagramme espace-temps d'accord de la règle 128 (qui signifie « passer dans l'état 0 si au moins un voisin est dans l'état 0 »). Il s'agit de la superposition de deux diagrammes espace-temps obéissant à la même règle 128, mais pour deux configurations initiales différentes. Voir le corps du texte pour le code couleur.

Définition 3 (Automates cellulaires élémentaires). Les automates cellulaires élémentaires sont la classe des automates cellulaires les plus simples :

- $Q = \{0, 1\}$
- $d = 1$
- $V = \{-1, 0, 1\}$

Compter le nombre d'automates élémentaires est facile : il y a $2^3 = 8$ configurations locales possibles. Pour chaque configuration locale, il faut choisir le résultat de la règle, il y a donc $2^8 = 256$ règles différentes. En pratique on n'en étudie que 88. En effet,

- Si deux règles δ et δ' sont identiques modulo une symétrie de la configuration, c'est-à-dire si $\forall q_1, q_2, q_3 \quad \delta(q_1, q_2, q_3) = \delta'(q_3, q_2, q_1)$, alors en connaissant les diagrammes espaces-temps de δ on peut déduire ceux de δ' par une simple symétrie. Cette opération préservant toute les propriétés habituellement étudiées, il n'est donc pas nécessaire d'étudier δ' .
- Si deux règles δ et δ' sont identiques modulo un échange des deux états 0 et 1, c'est-à-dire si $\forall q_1, q_2, q_3 \quad \delta(q_1, q_2, q_3) = 1 - \delta'(1 - q_1, 1 - q_2, 1 - q_3)$, alors en connaissant les diagrammes espaces-temps de δ on peut déduire ceux de δ' par un simple échange des deux états. De même, il n'est alors pas nécessaire d'étudier δ' .

En ne gardant qu'un représentant de chaque classe d'équivalence modulo ces deux symétries, il reste 88 règles.

Nous utilisons la notation introduite par Wolfram [Wolfram (1983)], qui numérote les règles de 0 à 255. Dans cette notation, une règle δ est notée par le nombre $2^7\delta(1, 1, 1) + 2^6\delta(1, 1, 0) + 2^5\delta(1, 0, 1) + 2^4\delta(1, 0, 0) + 2^3\delta(0, 1, 1) + 2^2\delta(0, 1, 0) + 2^1\delta(0, 0, 1) + 2^0\delta(0, 0, 0)$. Par exemple, le code de la règle « minorité », c'est-à-dire « passer dans l'état le moins présent dans la configuration locale » est 23 ; il est obtenu en lisant la dernière ligne de la table 2.1 comme un nombre en base 2.

TAB. 2.1 – Table de transition de la règle 23.

configuration locale	111	110	101	100	011	010	001	000
résultat de δ	0	0	0	1	0	1	1	1

2.1.1 Asynchronisme

La perturbation que nous allons le plus étudier est celle du synchronisme. Elle consiste à ne plus mettre à jour toutes les cellules en même temps. Pour noter quelles cellules de \mathcal{U} sont mises à jour, on utilise l'ensemble $\{0, 1\}^{\mathcal{U}}$, avec la convention que 1 représente la mise à jour d'une cellule et 0 une absence de mise à jour. Les éléments de $\{0, 1\}^{\mathcal{U}}$ sont donc les différentes façons de choisir une partie des cellules pour les mettre à jour.

Définition 4. Un *automate cellulaire asynchrone* est un vecteur $(Q, d, V, \delta, n, \mu)$ où

- Q, d, V, δ sont définis comme pour un automate cellulaire synchrone
- μ , la dynamique, est une mesure de probabilité sur $\{0, 1\}^{\mathcal{U}}$.

μ est la loi qui permet de tirer au sort, à chaque pas de calcul, un élément de $\{0, 1\}^{\mathcal{U}}$ déterminant l'ensemble des cellules à mettre à jour.

L'évolution est alors modifiée de la façon suivante. Soit $\{M(\cdot, t) \mid t \in \mathbb{N}\}$ une *histoire*, c'est-à-dire une suite de variables aléatoires indépendantes identiquement distribuées telle que chaque fonction $M(\cdot, t) : \mathcal{U} \rightarrow \{0, 1\}$ soit tirée selon la loi μ . Soit $c(\cdot, t)$ la configuration à l'étape t . La configuration au temps $t + 1$ est définie par

$$c_{t+1}(x) := \begin{cases} c_t(x) & \text{si } M_t(x) = 0 \\ \delta(c_t(x + v_1), \dots, c_t(x + v_{|V|})) & \text{si } M_t(x) = 1 \end{cases}$$

En d'autres termes, pour chaque cellule x , on applique la règle de transition δ si $M(x, t) = 1$ et l'on « gèle » la cellule (on conserve son état) si $M(x, t) = 0$.

Il y a de nombreuses possibilités pour choisir une loi μ et donc une dynamique asynchrone particulière. Nous utilisons les deux plus classiques [Fatès (2007)], qui sont les suivantes.

La dynamique partiellement asynchrone

Cela consiste à tirer au sort pour chaque cellule si elle va être mise à jour, indépendamment des autres cellules.

Pour cela, on fixe $0 < \alpha < 1$. Chaque cellule est mise à jour avec probabilité α , indépendamment des autres. μ est donc le produit de lois de Bernoulli. En effet, pour $m \in \{0, 1\}^{\mathcal{U}}$, notons $|m|_1 := \#\{z \in \mathcal{U} \mid m(z) = 1\}$ le nombre de cellules mises à jour. Alors $\mu(m) := \alpha^{|m|_1} (1 - \alpha)^{|m|_0}$ (avec par convention $0^0 = 1$). On peut noter que si l'on choisit $\alpha = 1$, on met à jour toutes les cellules à chaque étape, on obtient donc un automate synchrone. Un automate synchrone est ainsi un cas particulier d'automate asynchrone. La figure 2.3 est un exemple de diagramme espace-temps pour un automate en dynamique partiellement asynchrone.

La dynamique totalement asynchrone

On appelle souvent ce cas la dynamique séquentielle. Cela consiste, à chaque étape, à tirer au sort une cellule et à la mettre à jour. Cela définit $\mu(m)$ comme $1/n$ si $|m|_1 = 1$ et $\mu(m) := 0$ sinon. Cette dynamique peut être vue comme la limite en $\alpha \rightarrow 0$ de la dynamique partiellement asynchrone. Elle est également équivalente à la dynamique en temps continu suivante. Chaque cellule est munie d'une horloge, qui décompte le temps avant la prochaine mise à jour. Quand une horloge arrive à 0, sa cellule est mise à jour et l'horloge est réglée sur un temps aléatoire de loi exponentielle. Puisque la loi exponentielle est sans mémoire, on obtient la même loi de probabilité sur les valeurs des

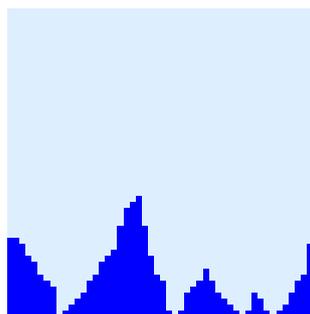


FIG. 2.3 – Diagramme espace-temps de la règle 128 (qui signifie « passer dans l'état 0 si au moins un voisin est 0 ») en dynamique partiellement asynchrone. $\alpha = 0,5$ et la configuration initiale a une majorité de 1 (cellules sombres). Les diagrammes espace-temps de tous les automates élémentaires sont disponibles sur <http://www.rouquier.org/jb/recherche/eca>, pour diverses valeurs de α .

horloges en faisant un nouveau tirage pour *toutes* les horloges chaque fois qu'une cellule est mise à jour. Ce qui revient à tirer au sort la prochaine cellule qui sera mise à jour, puisque avec probabilité 1 tous les temps sont différents.

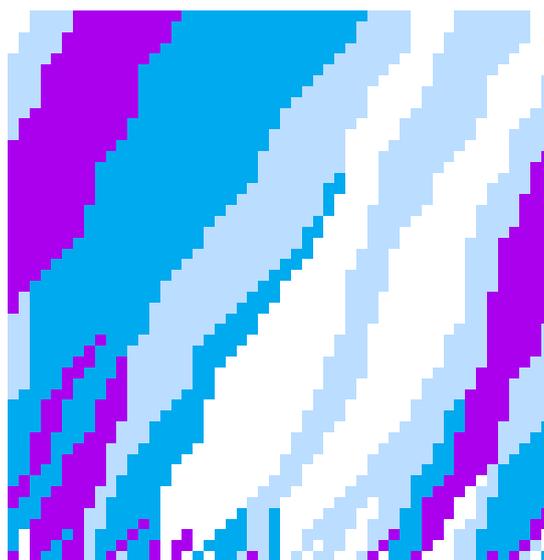


FIG. 2.4 – Diagramme espace-temps d'accord de la règle 170, qui signifie « copier l'état du voisin de gauche ». Le temps va de bas en haut pour 50 étapes, $n = 50$. Les cellules où les deux configurations sont en désaccord sont dessinées sombres, celles où elles sont en accord sont dessinées clair (bleu clair ou blanc selon l'état commun).

2.2 Transitions de phase et percolation dirigée

Nous verrons dans les chapitres 3 et 4 un phénomène de percolation dirigée. Nous proposons dans cette section quelques rappels minimaux sur la percolation dirigée : nous expliquons ce qu'est une transition de phase, présentons un conjecture impliquant que la plupart des transitions que nous étudierons sont dans la classe d'universalité de la percolation dirigée, et décrivons enfin cette classe d'universalité.

Transition de phase

Une transition de phase est un brusque changement d'une propriété macroscopique d'un système, provoquée par un faible changement d'un paramètre, noté par exemple T , autour d'une valeur critique T_c . Des exemples classiques sont la transition entre les états solide, liquide et gazeux (la valeur critique est alors la température de fusion ou d'ébullition), la magnétisation (où la valeur critique est appelée point de Curie), ou la supraconductivité.

Ce travail s'intéresse à des transitions de phase de second ordre, ou continues, où l'observable décrit une courbe continue en fonction des variations du paramètre (ce qui n'est pas le cas pour la transition entre solide, liquide et gaz par exemple). Ces transitions peuvent être caractérisées de la façon suivante. Si l'on fait varier le paramètre T près de la transition (qui se produit en $T = T_c$), tous les autres paramètres étant fixés, un observable que nous noterons C suit une loi de puissance $C \propto |T - T_c|^\beta$ d'un côté de T_c . Plusieurs exposants (dont les plus courants sont appelés β , δ , ν ...) sont définis à partir de divers observables dans les conditions $T = T_c$. Ce sont des *exposants critiques*. De façon remarquable, beaucoup de systèmes sans lien a priori se révèlent avoir les mêmes exposants critiques. Une *classe d'universalité* est définie comme la classe des systèmes ayant le même ensemble d'exposants critiques.

Le modèle de la percolation dirigée

Une classe d'universalité particulière est celle de la percolation dirigée. Une introduction plus détaillée à la percolation dirigée se trouve dans [Fatès (2007)] (noter que cet article cite une autre conjecture de Grassberger que celle que nous utilisons ici). Une vue d'ensemble de la percolation dirigée est disponible dans [Hinrichsen (2000)], qui passe également en revue la propagation d'erreur (ou *damage spreading*).

La percolation isotrope a d'abord été définie en étudiant la propagation d'un fluide à travers un milieu poreux. Un tel milieu a été modélisé par une grille carrée infinie où chaque site a pour voisin les quatre sites les plus proches. Chaque lien entre deux sites voisins peut être ouvert (et laisse alors passer le fluide) avec probabilité p ou bien fermé avec probabilité $1 - p$. Chaque lien est indépendant des autres. On observe alors si le fluide inséré en un point peut traverser le milieu, i.e. si le point de départ fait partie d'un réseau infini de sites connectés par des liens ouverts.

La percolation dirigée apparaît quand on ajoute la gravité au modèle, i.e. que le fluide ne peut se propager que dans une direction (voir figure 2.5 page suivante). On rencontre aussi la percolation dirigée en une dimension, si l'on ajoute le temps comme seconde dimension. Dans ce dernier modèle, certains sites sont *actifs* (ce qui, selon le système modélisé, représente un site mouillé par le fluide, ou infecté, ou contenant une erreur, etc.). Un site actif peut rester actif ou devenir inactif (ce qui est souvent appelé « mourir »), il peut de plus rendre ses voisins actifs à l'étape suivante. Selon la probabilité de ces possibilités, des régions de sites actifs s'étendent ou se résorbent. Les sites actifs ne peuvent avoir une influence que sur l'état futur de leurs voisins, d'où le nom de percolation *dirigée*. Les deux modèles (percolation dirigée en deux dimensions qui ne fait pas intervenir le temps, et percolation dirigée dynamique en une dimension) sont équivalents.

Pour étudier cette transition, on mesure la densité de sites actifs en fonction de p et du temps : $\rho(p, t)$. Cette quantité est nulle dans une phase et non nulle dans l'autre. Il existe donc une probabilité critique p_c qui est la limite entre les deux phases :

- pour $p < p_c$, la densité asymptotique $\rho(p, \infty)$ est 0 ;
- pour $p > p_c$, elle suit une loi de puissance $\rho(p, \infty) \propto (p - p_c)^\beta$;
- pour $p = p_c$ la densité tend vers 0 selon une autre loi de puissance : $\rho(p_c, t) \propto t^{-\delta}$. L'exposant δ est également un exposant critique ; nous le mesurerons pour confirmer qu'un modèle appartient à la classe d'universalité de la percolation dirigée.

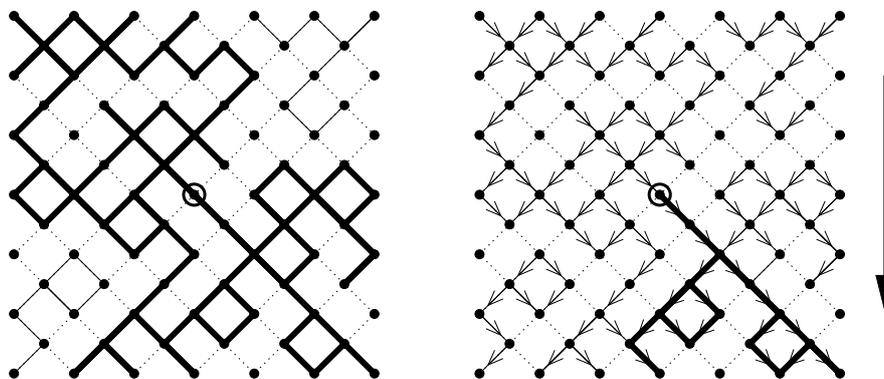


FIG. 2.5 – Percolation isotrope (à gauche) et dirigée (à droite). Figure originale dans [Hinrichsen (2000)].

Une conjecture sur la propagation d'erreur

La théorie du chaos étudie la sensibilité aux conditions initiales de systèmes *déterministes*. Pour étudier l'influence de petites perturbations sur des systèmes stochastiques, les auteurs de [Kauffman (1984)] ont introduit la propagation d'erreur, ou *damage spreading*. Dans ce modèle, deux copies d'un système sont simulées en parallèle avec la même source d'aléatoire, mais en partant d'une configuration initiale différente (souvent les deux configurations ne diffèrent que sur une cellule). Ces auteurs ont étudié cette construction (deux copies du système évoluant en parallèle) sur les réseaux d'automates booléens.

On mesure alors l'évolution temporelle de la proportion de cellules dans des états différents (c'est la distance de Hamming). Si cette densité tombe à zéro, c'est-à-dire si les deux copies deviennent identiques, l'« erreur » initiale a été « corrigée » ; sinon on dit que l'erreur se propage.

Une conjecture de [Grassberger (1995)] affirme que, si une transition entre correction et propagation d'erreur se produit dans un système stochastique de spins² en interaction, alors la classe d'universalité de cette transition de phase est celle de la percolation dirigée.

Il y a quelques conditions pour appliquer la conjecture :

1. les interactions sont locales dans le temps et l'espace,
2. la topologie est une grille invariante par translation,
3. pour tout site différent entre les deux copies, la probabilité de corriger l'erreur est supérieure à une constante (qui dépend du système),
4. la transition ne coïncide pas avec une autre transition de phase comme c'est le cas par exemple dans le modèle d'Ising [McCoy and Wu (1973)].

²Un spin est l'équivalent d'une cellule qui peut prendre les états 0 et 1. Cette conjecture s'applique aux automates cellulaires, mais aussi à des modèles proches où la règle locale détermine le nouvel état de plusieurs cellules simultanément.

Première partie

Couplages et percolation dirigée

Il est naturel lorsque l'on s'intéresse aux perturbations de comparer deux instances du même système. Par exemple, on perturbe l'une des deux instances et l'on compare son orbite à celle de l'autre instance, non perturbée. Ou bien l'on introduit des perturbations similaires dans les deux instances puis l'on étudie les corrélations entre les deux instances, ce qui indirectement permet de mesurer l'influence des perturbations. Pour effectuer ces comparaisons, nous utilisons dans cette partie la notion de *couplage*, qui consiste à faire évoluer deux configurations de même taille, selon la même règle, pour le même nombre d'étapes de calcul. Les deux configurations sont dites couplées car on ajoute au modèle un ingrédient qui lie les deux configurations.

Dans le chapitre 3, il s'agira de forcer certaines cellules à être dans le même état entre les deux configurations. C'est-à-dire que pour un ensemble de coordonnées tirées au sort, pour chacune de ces coordonnées, les cellules de chacune des deux configurations à cette coordonnée sont forcées à être dans le même état. On étudie alors si imposer entre les deux configurations une telle égalité sur quelques cellules à chaque étape permet d'obtenir une égalité sur toutes les cellules.

Dans le chapitre 4, le couplage est plus léger et porte cette fois seulement sur les mises à jour. On utilise la dynamique partiellement asynchrone qui met à jour un sous-ensemble des cellules à chaque étape. Le couplage consiste simplement à choisir le même sous-ensemble pour les deux configurations.

Dans ces deux modèles, on observe que pour certains paramètres les deux configurations deviennent identiques. Nous appelons ce phénomène *coalescence*. Nous étudions alors plus précisément quels paramètres font apparaître la coalescence, et l'on observe une transition de phase. Dans les deux modèles, nous montrons expérimentalement que cette transition de phase appartient la plupart du temps à la classe d'universalité de la percolation dirigée.

Le chapitre 5 cherche alors une validation formelle de ce dernier résultat. Pour cela, nous considérons d'une part les automates cellulaires asynchrones (pour lesquels la percolation dirigée a été identifiée par [Fatès (2004)]), et d'autre part un modèle largement accepté de percolation dirigée : l'automate de Domany-Kinzel [Domany and Kinzel (1984)]. Nous montrons alors une simulation formelle entre deux classes d'automates cellulaires. Le résultat obtenu s'applique pour l'instant aux modèles utilisant une seule configuration ; les modèles de couplage entre deux configurations pourront faire l'objet de futurs travaux.

Chapitre 3

Couplage par forçage

La première forme de perturbation que nous étudions consiste à modifier l'état de certaines cellules au cours de l'évolution. Ici nous étudions une forme de perturbation qui n'est pas arbitraire mais au contraire une perturbation qui ait du sens par rapport au système, qui soit plus pertinente par rapport au système. Pour cela, on considère deux systèmes soumis à la même dynamique, et les perturbations apportées à un système sont inspirées de l'évolution de l'autre. Précisément, on considère deux configurations initiales aléatoires, et on les fait évoluer selon la même règle. La perturbation d'une cellule à une étape consiste à modifier (avec une certaine probabilité) l'état de cette cellule pour qu'elle soit dans le même état entre les deux configurations. (Bien sûr, si la cellule est déjà dans le même état entre les deux configurations, la perturbation n'a pas d'effet).

Cette perturbation a déjà été étudié dans [Morelli and Zanette (1998)]. Une revue de plusieurs modèles est proposée par les mêmes auteurs dans [Zanette and Morelli (2003)]. Les articles de la littérature sur ce modèle n'étudient qu'une ou deux règles, choisies pour leur facilité d'implantation ou pour leur comportement chaotique quand elles ne sont pas perturbées. Nous verrons dans ce chapitre que cette restriction est pertinente, car seuls deux comportements sont observés sur les automates élémentaires : l'un est trivial, l'autre est celui des règles chaotiques. Cependant, passer en revue tous les automates élémentaires permet de confirmer qu'il n'y a pas d'autre comportement, et de plus montre qu'il n'est pas nécessaire que la règle de base soit chaotique pour obtenir un comportement intéressant (ici une transition de phase) avec cette perturbation.

Une autre façon de considérer ce chapitre est le point de vue du contrôle : perturber un système est aussi un premier pas vers son contrôle, comme nous l'avons vu dans l'introduction à la section 1.2.3. En effet, en choisissant les perturbations apportées, on peut tenter d'amener le système vers la configuration désirée. Dans ce modèle, on étudie dans quels cas il est possible de rendre deux configurations identiques en n'exerçant qu'un contrôle partiel.

Plan du chapitre Le modèle est défini à la section 3.1, et les expériences décrites à la section 3.2. Nous proposons dans cette section une classification de tous les automates cellulaires élémentaires (section 3.2.1) ; nous voyons que seuls deux comportements sont possibles : synchronisation dès la plus petite perturbation, ou bien synchronisation seulement au-delà d'un seuil du paramètre. Nous étudions alors plus en détail la seconde classe et testons l'hypothèse de la percolation dirigée (section 3.2.2).

Ce travail a été publié dans [Rouquier (2008)].

3.1 Le modèle

Voici la perturbation que nous étudions. Étant donné un automate cellulaire, on considère deux configurations initiales c_0^1 et c_0^2 . Ces configurations sont aléatoires et l'état de chaque cellule est indépendant des cellules voisines et tiré selon une loi uniforme parmi tous les états. À chaque étape, chacune des deux configurations est mise à jour selon une règle commune δ (cette règle δ est un paramètre du modèle). Puis on applique une fonction aléatoire F_q sur le couple de configurations, dans le but de rapprocher les deux configurations.

F_q consiste, pour chaque cellule z indépendamment, à ne rien faire avec probabilité q , et à forcer les deux configurations à être dans le même état avec probabilité $1 - q$. Dans ce dernier cas, l'état commun aux deux cellules est choisi aléatoirement uniformément entre $c^1(z)$ et $c^2(z)$:

Pour chaque cellule z indépendamment,

$$F_q(c^1(z), c^2(z)) := \begin{cases} c^1(z), c^2(z) & \text{avec probabilité } q \\ c^1(z), c^1(z) & \text{avec probabilité } \frac{1-q}{2} \\ c^2(z), c^2(z) & \text{avec probabilité } \frac{1-q}{2} \end{cases}$$

En combinant F_q et δ , on obtient :

$$c_{t+1}^i(z) := \begin{cases} \delta(c_t^i)(z) & \text{avec probabilité } q & (a) \\ \delta(c_t^1)(z) & \text{avec probabilité } \frac{1-q}{2} & (b) \\ \delta(c_t^2)(z) & \text{avec probabilité } \frac{1-q}{2} & (c) \end{cases}$$

Pour chaque cellule z indépendamment, le même choix parmi (a), (b) et (c) est fait pour chacune des deux configurations. Un exemple est donné sur la figure 3.1 page suivante.

La probabilité q est un paramètre du modèle. Le cas $q = 1$ correspond à l'automate non perturbé, i.e. deux configurations évoluant indépendamment. Le cas $q = 0$ implique la coalescence dès la première étape.

On peut noter que si les deux états sont déjà égaux, la phase de forçage n'a aucun effet. On peut donc définir le modèle en ne décidant de forcer les cellules que si elle sont dans des états différents. C'est cette présentation équivalente qui est retenue dans [Morelli and Zanette (1998)].

La quantité que nous étudions est la densité de cellules différentes entre les deux configurations, et donc si les configurations finissent par devenir identiques (par *coalescer*) ou non. La pertinence de cette mesure globale a été montrée par [Fatès (2003)]. Formellement, la densité est

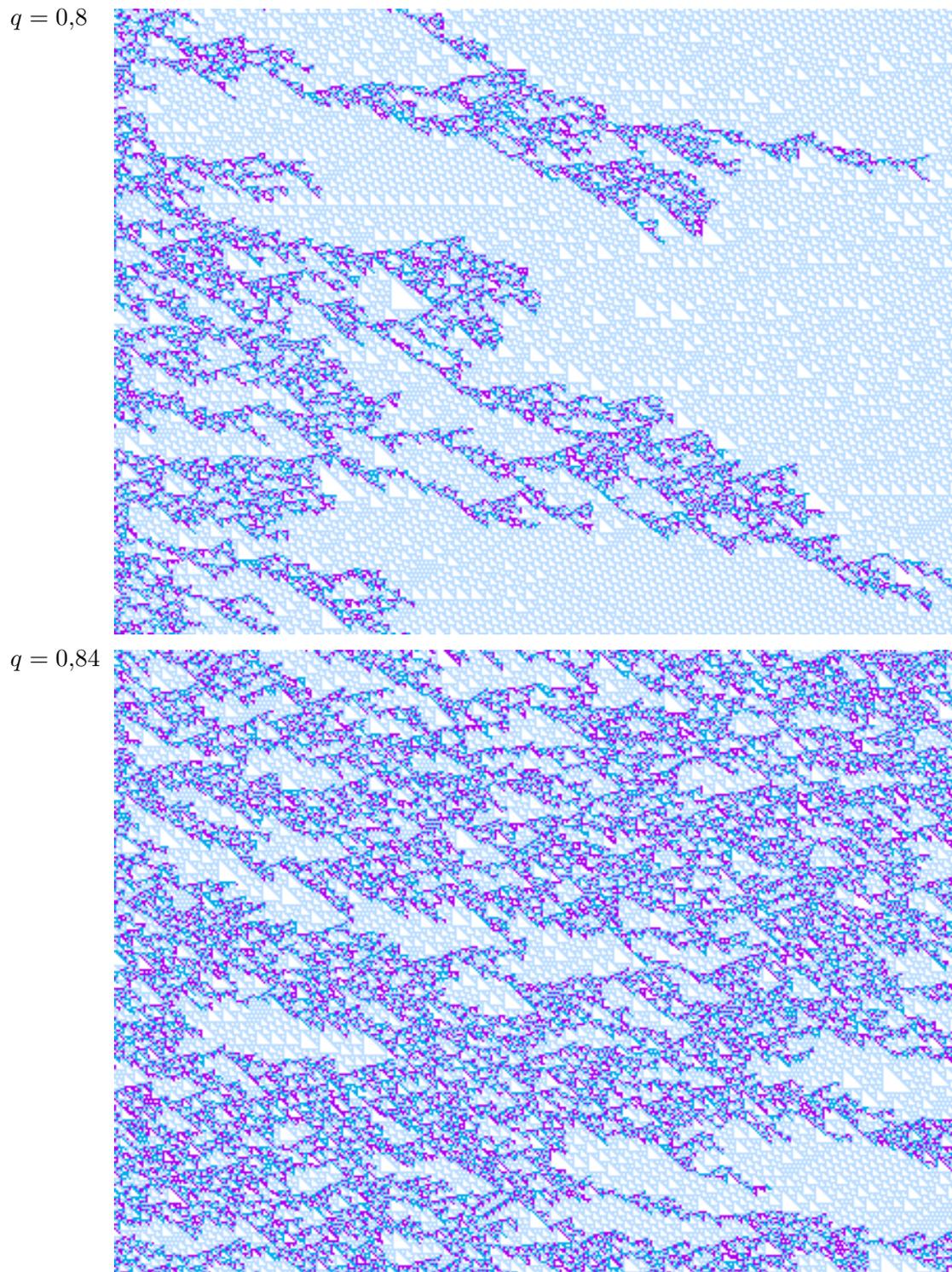
$$\rho(c) := \#\{x \in \mathcal{U} \mid c_1(x) \neq c_2(x)\} / |\mathcal{U}|$$

Fait 5 (Fort couplage). *Pour une configuration finie de taille n , si $q \leq \frac{1}{|V|}$ alors la coalescence apparaît après $O(n)$ étapes en moyenne.*

Démonstration. La densité de cellules en désaccord est, en moyenne, multipliée par q chaque fois que F_q est appliquée. Elle est multipliée par au plus $|V|$ en appliquant δ , car une cellule en désaccord ne peut propager le désaccord qu'à ses voisines. Ainsi, si $q|V| < 1$, l'espérance de la densité décroît. ■

Remarque : cette borne sur q est assez lâche. En effet, la densité de cellules en désaccord n'est multipliée par $|V|$ que si

- le désaccord se propage à tous les voisins, ce qui n'est pas le cas de toutes les règles ;



Le temps va de gauche à droite. Les configurations c^1 et c^2 sont superposées. Les cellules en accord (qui sont dans le même état entre les deux configurations) sont représentées d'une couleur claire (bleu clair ou blanc) ; celles en désaccord sont représentées d'une couleur sombre (bleu foncé ou violet). On peut voir dans les zones claires des dessins caractéristiques de la règle 110.

FIG. 3.1 – Le modèle de forçage appliqué à la règle 110.

- toutes les cellules en désaccord sont isolées. Mais le désaccord se propage de voisine en voisine, les cellules en désaccord tendent donc à être groupées au lieu d’être isolées.

3.1.1 Lien avec un autre modèle

Le modèle étudié ici en a inspiré d’autres, notamment le suivant. Les auteurs de [Sánchez and López-Ruiz (2005)] étudient ce qu’ils appellent « auto-synchronisation », c’est-à-dire qu’ils considèrent une seule configuration, qu’ils tentent de « synchroniser » avec elle-même. Plus précisément, ils étudient la coalescence entre une étape et la suivante. Ces deux configurations sont identiques lorsque la configuration est un point fixe.

Pour cette étude, ils posent $c^2(\cdot, t) := c^1(\cdot, t - 1)$, puis appliquent le modèle que nous venons de décrire. Cela signifie que, à chaque étape et pour chaque cellule indépendamment, la cellule est ramenée à son état de l’étape précédente avec probabilité $\frac{1-q}{2}$. Cela est équivalent à ne mettre à jour la cellule qu’avec probabilité $\frac{q+1}{2}$, et ne rien faire sinon (geler l’état de la cellule).

Ce modèle, consistant à ne mettre à jour qu’une partie des cellules, est celui des automates partiellement asynchrones, que nous utiliserons aux chapitres suivants. Il a déjà été étudié expérimentalement et analytiquement par [Fatès (2007)].

3.2 Étude expérimentale

Dans cette section, nous examinons le comportement de tous les automates élémentaires pour le modèle de forçage introduit ci-dessous. Nous verrons que seuls deux comportements sont possibles : coalescence rapide dès qu’un infime forçage est introduit, ou bien une transition de phase de percolation dirigée quand q varie.

3.2.1 Classification des automates élémentaires

Protocole

Voici le protocole de l’expérience.

Appelons *lancer* l’orbite d’un automate cellulaire lorsque tous les paramètres sont fixés : règle, nombre de cellules n , probabilité q et couple de configurations initiales. Un lancer est stoppé lorsque les deux configurations sont identiques, ou bien lorsqu’un nombre d’étapes fixé à l’avance est atteint.

Voici les paramètres utilisés.

1. Nombre de cellules n . Que l’on s’intéresse à des systèmes finis ou infinis, le point essentiel lors du choix de n est de s’assurer qu’il n’influe pas sur les résultats, en particulier qu’il est assez grand. Certains auteurs, comme [Wuensche (1999)], ont obtenu des résultats intéressants sur de petites configurations (jusqu’à $n = 30$). Notre étude est plus proche des travaux de [Broadbent and Hammersley (1957)] qui suggèrent au contraire qu’une grande taille est nécessaire. Nous prenons comme guide un problème similaire étudié par [Fatès (2007)], qui montre un comportement stable pour $n \geq 200$. Nous fixons $n = 2\,000$ et vérifions que les résultats ne changent pas pour $n = 500$.
2. Nombre d’étapes de calcul. Pour évaluer la densité asymptotique $\rho_\infty(q)$ de cellules en désaccord, on laisse la configuration évoluer pendant 200 000 étapes, puis l’on mesure la densité moyenne sur 10 000 étapes.

Remarque. D'après la théorie de la percolation dirigée, près d'une transition de phase, le système peut nécessiter un nombre arbitrairement grand d'étapes avant de trouver la densité asymptotique. Ainsi, pour quelques choix de q , ces paramètres sont insuffisants pour mesurer $\rho_\infty(q)$. Ils sont cependant suffisants pour *détecter* qu'il y a une transition de phase et décider alors d'étudier plus précisément ce qui se passe à cet endroit.

3. q . Sans connaissance à priori sur les valeurs intéressantes, nous échantillons tout l'intervalle $[0; 1]$. Précisément, pour chacune des 88 règles, 999 lancers sont effectués : un pour chaque valeur de q de 0,001 à 0,999.

On pourrait être tenté d'affiner les résultats en moyennant $\rho_\infty(q)$ sur plusieurs lancers à paramètres identiques (mais à séquence aléatoire différente). Cela n'est pas nécessaire comme le montre la courbe $\rho_\infty(q)$: la régularité de la figure 3.2 page suivante montre que la variabilité entre les lancers est faible.

L'initialisation du générateur aléatoire est distincte pour chaque valeur de q . Le chapitre 4 montre l'importance de ce point.

4. Les configurations initiales sont aléatoires (uniformément distribuées) et distinctes pour chaque valeur de q .

Résultats

En appliquant ce protocole à tous les automates élémentaires, seuls deux comportements apparaissent. La figure 3.2 page ci-contre donne un exemple de chacun des deux cas.

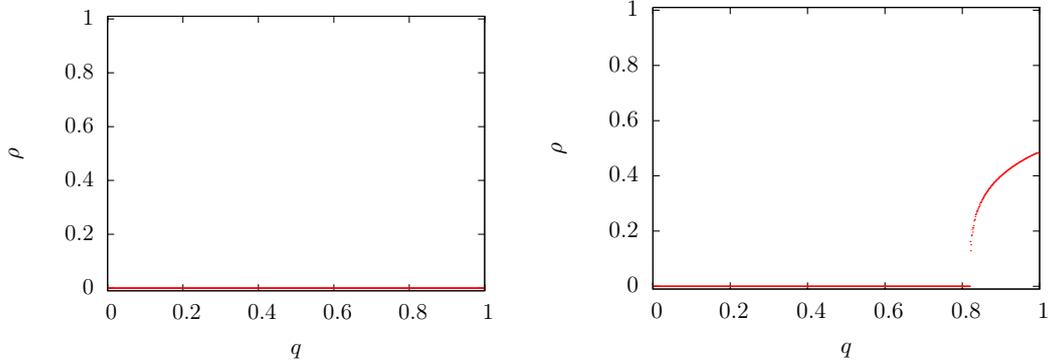
- 68 automates ont un comportement trivial : les deux configurations coalescent toujours pendant le temps imparti, pour toutes les valeurs de q .
- 20 automates subissent une transition de phase : pour un certain q_c (dépendant de la règle),
 - si $q < q_c$, les deux configurations coalescent rapidement,
 - si $q > q_c$, la densité de cellules en désaccord converge vers une valeur non nulle.

Nous avons discuté à la section 1.3.1 le choix d'un système fini ou infini. Nous y annonçons en particulier que nous vérifierions que la taille n'a pas d'influence, ce que nous faisons maintenant.

Un ordinateur ne peut manipuler que des configurations finies et peut donc être sujet à certains artefacts. Un effet notable est le suivant. Considérons une règle ayant une transition de phase, et une valeur de q pour laquelle la densité ρ de cellules en désaccord se stabilise à une valeur non nulle. Dans ce régime sur-critique, c'est-à-dire non rapidement coalescent, il est possible (avec une faible probabilité) que le résultat des tirages aléatoires déterminant quelles cellules forcer se produise comme si l'on avait fixé une valeur de q où ρ tend vers 0. Le système se comporte alors pendant quelques étapes comme dans le régime sous-critique (rapidement coalescent). Ainsi, si un automate est rapidement coalescent pour un certain q (et c'est le cas de tous les automates comme vu à la proposition 5 page 29), alors quel que soit q le système converge toujours vers l'état absorbant. Le véritable régime asymptotique est donc toujours la coalescence. Mais en régime sur-critique il n'apparaît qu'après un temps exponentiel. Cela nous permet de conjecturer que les résultats présentés ici concernant la densité et la transition de phase s'appliquent également à des systèmes infinis.

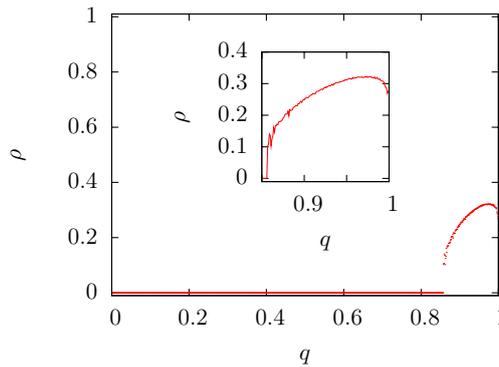
En d'autres termes, quand la densité s'établit à une valeur non nulle (i.e. dans le cas $q > q_c$), elle continue néanmoins d'osciller autour de cette valeur. Les fluctuations

finissent par lui faire toucher 0, qui est absorbant. La densité est alors définitivement nulle. Cependant, des fluctuations d'une taille donnée deviennent de plus en plus rares à mesure que n croît, le temps pour obtenir une fluctuation qui fasse toucher 0 augmente donc.



Une règle pour laquelle les deux configurations coalescent toujours rapidement (26)

Une règle avec transition de phase (110)



La règle 57

FIG. 3.2 – $\rho_\infty(q)$ (densité asymptotique de cellules en désaccord).

3.2.2 Percolation dirigée

La percolation dirigée a été présentée à la section 2.2. Dans notre modèle, les sites actifs sont les cellules en désaccord. Le désaccord peut en effet se propager, mais une fois les deux configurations en accord sur une région, aucune cellule de cette région ne peut devenir en désaccord ailleurs qu'au bord de la région. Les couples de configurations identiques (où toutes les cellules sont en accord) constituent l'ensemble absorbant. La transition entre coalescence rapide ou non apparaît quand on fait varier q , voir figure 3.1 page 30. Noter qu'il n'y a pas a priori de relation simple entre q et le paramètre p de la percolation dirigée. Le but est donc d'identifier β sous l'hypothèse que, pour q proche d'un certain q_c ,

$$\rho_\infty(q) \propto |q - q_c|^\beta \quad (3.1)$$

Comme plusieurs auteurs dont [Grassberger (1995)], nous nous concentrons sur la mesure de β en la considérant suffisante pour tester la présence de percolation dirigée.

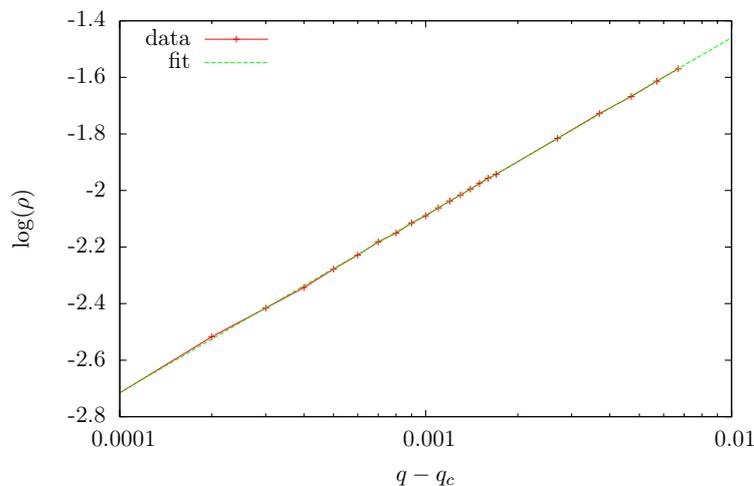


FIG. 3.3 – Mesure de β pour la règle 110.

Mesure de β

Une méthode pour mesurer β est la suivante (nous discutons une autre méthode à la section 4.2.2). On trace $\log \rho_\infty(q)$ en fonction de $\log(q - q_c)$ pour des valeurs de q proches de q_c , puis l'on ajuste q_c pour obtenir une ligne droite.

Une fois que q_c est déterminé, on cherche par régression linéaire la pente de cette ligne, qui est un estimateur de β (cf. équation 3.1 page précédente). Cela est illustré sur la figure 3.3.

Remarque. Il est important d'effectuer une régression linéaire sur $\log \rho_\infty$ (et non la régression d'une loi de puissance sur ρ_∞), afin que les erreurs aient des poids équitables lors de l'ajustement d'une ligne sur le diagramme en échelle doublement logarithmique. Sinon, les erreurs pour ρ_∞ proche de 0 auraient un poids quelques ordres de grandeurs plus faible que les erreurs pour ρ_∞ non négligeable et seraient donc ignorées.

Protocole Voici le détail des opérations effectuées pour estimer β . Le protocole n'est que semi-automatique. Pour une valeur de q , on trace $\rho(q, t)$ jusqu'à ce que cette densité soit stabilisée à une valeur constante (ce qui est vérifié visuellement). On mesure alors la densité moyenne sur au moins une demi-décade. Cela donne un point de mesure $\rho_\infty(q)$, on répète donc le processus pour q variant entre $q_c + 10^{-4}$ et $q_c + 10^{-2}$.

Afin d'obtenir des courbes lisibles, c'est-à-dire de distinguer parmi deux valeurs candidates de q_c celle qui donne une ligne droite, il a été nécessaire de choisir les paramètres jusqu'à $n = 10^6$ cellules et 10^7 étapes.

La courbe $\rho(q, t)$ est d'autant plus bruitée que q est proche de q_c . On essaye donc d'abord avec comme paramètre $n = 10^4$ pour accélérer les calculs. Quand la courbe est trop bruitée, on augmente la valeur de n jusqu'à 10^6 , qui est la limite des capacités de calcul dont nous disposons actuellement. Cela permet de vérifier que, tous les autres paramètres étant fixés,

- le temps transitoire ne dépend pas de n ,
- de même, différentes valeurs de n montrent des fluctuations d'intensité différentes mais autour de la même valeur. Augmenter n permet donc d'améliorer la précision

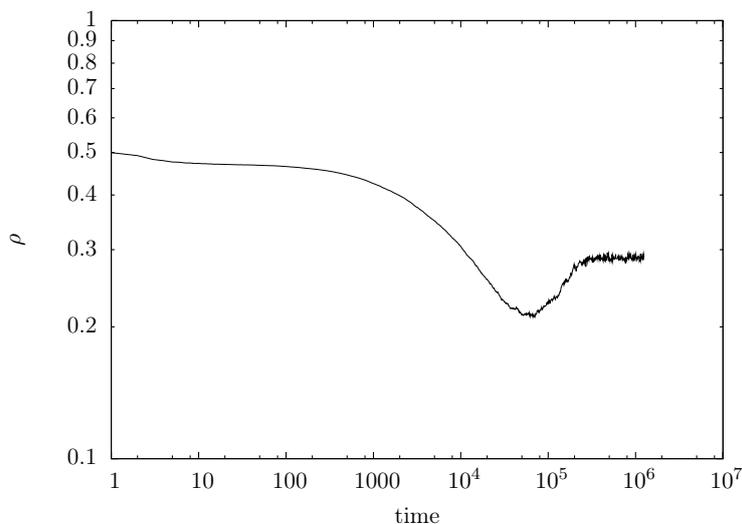


FIG. 3.4 – $\rho(q, t)$ pour la règle 58, en $q = 0,999\ 9$ (échelles logarithmiques).

mais n'affecte pas le résultat.

Noter le cas trompeur de cette courbe $\rho(q, t)$ pour la règle 58 (figure 3.4) : le calcul de 50 000 étapes peut laisser croire que la densité va atteindre 0, puis lors des étapes suivantes la densité croît vers une valeur stable. Ce cas est particulier à la règle 58, pour les autres règles les courbes sont monotones.

Résultats

La régression donne les intervalles de la table 3.1 page suivante. Ces intervalles prennent en compte l'incertitude sur q_c . La valeur expérimentale de β mesurée sur d'autres modèles de percolation dirigée est 0,276. On peut noter que l'on connaît une valeur expérimentale précise de β mais pas de valeur théorique en une dimension. Une valeur exacte des exposants de percolation dirigée est connue à partir de la dimension 4 [Hinrichsen (2000)].

Ce chapitre apporte donc la confirmation suivante : toutes les règles subissant une transition de phase (i.e. les règles qui ne coalescent pas rapidement pour certaines valeurs de q) semblent bien appartenir à la classe d'universalité de la percolation dirigée.

Les seuils de transition q_c pour les règles 58 et 62 sont très proches de 1. Pour ces règles, une toute petite perturbation (environ 1 %) est déjà suffisant pour que l'automate coalesce rapidement, mais il y a néanmoins une valeur de q pour laquelle il ne coalesce pas rapidement.

Quelques règles ont des seuils très proches, et ce n'est pas une coïncidence : on peut trouver des liens entre leurs orbites.

- 18 et 146. La seule différence entre ces règles est le cas de la configuration locale 111 : $\delta(1, 1, 1) = 0$ pour la règle 18 tandis que $\delta(1, 1, 1) = 1$ pour 146. Mais le seul cas où le motif 1^k (avec $k \geq 3$) apparaît pour la règle 146 est quand il y a le motif 1^{k+2} à l'étape précédente. Ainsi, pourvu qu'il y ait au moins un 0 dans la configuration initiale, ces motifs disparaissent rapidement et les orbites de 18 et 146 sont alors identiques.
- 60 et 90. La règle 90 s'interprète par « XOR entre mon voisin de gauche et mon voisin de droite ». 60 signifie « XOR entre mon voisin de gauche et moi ». Notons \oplus

règle	q_c	β
9	0,963 0(2)	0,306 \pm 0,031
18	0,809 2(2)	0,285 \pm 0,021
22	0,772 7(2)	0,268 \pm 0,018
25	0,957 0(2)	0,288 \pm 0,016
30	0,793 5(1)	0,269 \pm 0,016
41	0,795 4(1)	0,277 \pm 0,013
45	0,794 6(1)	0,275 \pm 0,011
54	0,838 7(2)	0,283 \pm 0,012
57	0,854 6(1)	0,295 \pm 0,026
58	0,996 8(2)	0,260 \pm 0,030
60	0,809 4(1)	0,270 \pm 0,015
62	0,985 4(2)	0,291 \pm 0,027
90	0,809 4(2)	0,263 \pm 0,022
105	0,678 9(2)	0,268 \pm 0,010
106	0,849 8(1)	0,275 \pm 0,009
110	0,819 30(1)	0,272 \pm 0,005
122	0,785 0(1)	0,274 \pm 0,011
126	0,789 2(2)	0,269 \pm 0,011
146	0,809 4(2)	0,259 \pm 0,021
150	0,678 9(2)	0,265 \pm 0,013

TAB. 3.1 – q_c et β pour tous les automates élémentaires subissant une transition de phase. Les nombres entre parenthèses donnent la précision : «0,9630(2)» signifie $q \in [0,9628; 0,9632]$. Il a été consacré plus de temps à la règle 110 pour la mesurer avec une plus grande précision que les autres.

le ou exclusif et considérons un diagramme espace-temps $\{c_t(z) \mid t \in \mathbb{N}, z \in \mathcal{U}\}$ de la règle **90**. Alors

$$\forall z \quad \forall t \quad c_{t+1}(z) = \delta(c_t(z-1), c_t(z), c_t(z+1)) = c_t(z-1) \oplus c_t(z+1)$$

Si l'on en extrait le diagramme espace-temps $\{c'_t(z) := c_t(2z-t) \mid t \in \mathbb{N}, z \in \mathcal{U}\}$, il obéit à la règle **60** :

$$\begin{aligned} \forall z \quad \forall t \quad c'_{t+1}(z) &= c_{t+1}(2z-t-1) \\ &= c_t(2z-t-2) \oplus c_t(2z-t) \\ &= c'_t(z-1) \oplus c'_t(z) \end{aligned}$$

Ainsi, un diagramme espace-temps de la règle **90** contient deux diagrammes de largeur moitié de la règle **60**, modulo décalage des cellules. La densité ρ est donc identique entre ces règles.

- **105** et **150**. La règle **150** signifie «XOR entre les états des mes voisins» tandis que **105** s'interprète par «calculer le résultat de **150** et prendre l'opposé». Le fait qu'une cellule soit en accord ou non entre les deux configurations dépend donc uniquement, pour la règle **105** comme pour la règle **150**, de l'état d'accord ou désaccord de ses voisins à l'étape précédente. Il ne dépend pas de l'état des voisins dans chacune des configurations. Cela nous permet de conclure que, si les deux règles sont appliquées à la même paire de configurations initiales, exactement les mêmes cellules seront en accord ou désaccord à chaque étape.

Le cas de la règle 57

Les précédents travaux utilisant ce modèle de forçage n'ont étudié que quelques règles chaotiques, choisies pour leur facilité de simulation ou bien les grandes chances qu'elles avaient de donner des résultats intéressants. Cependant, il n'est pas nécessaire de choisir une règle chaotique. En effet, si nous considérons la règle **57**, on voit que sans perturbation une configuration converge rapidement vers un cycle de période deux (précisément, un damier). Elle n'est donc pas chaotique. Pourtant, lorsque l'on ajoute le forçage entre deux configurations, il apparaît une transition de phase. Cela montre qu'une telle transition de phase n'apparaît pas que pour des règles chaotiques. En revanche, tous les automates élémentaires des classes 3 et 4 de la classification expérimentale de Wolfram (i.e. les règles «chaotiques» ou «complexes») présentent une transition de phase.

Cette règle **57** a un intérêt supplémentaire. Ses diagrammes espace-temps (figure **3.5** page suivante) montrent en effet des particules, et donc des corrélations à longue distance. La règle **110** est également célèbre pour avoir des particules (voir figure **3.1** page **30**) et donc exactement le même type de corrélations à longue distance. Nous avons montré que les deux règles présentent une transition de phase de percolation dirigée. Cela montre que, dans la conjecture de Grassberger (page **22**), le point **1** est utile pour rejeter certains modèles mais est peut-être trop restrictif.

Cette règle nous apprend encore un point. Sur la figure **3.2** page **33.c**, on peut voir que pour q entre 0,98 et 0,999, augmenter le forçage (i.e. q plus proche de 0) implique une densité asymptotique plus élevée. En d'autres termes, augmenter le nombre de cellules que l'on force à être égales à chaque étape rend la densité de cellules en désaccord plus élevée.

Notons qu'appliquer F_q est équivalent à appliquer deux fois $F_{\sqrt{q}}$. Dans ce cas, appliquer $F_{\sqrt{q}}$ une seconde fois à chaque étape augmenterait la densité ρ .

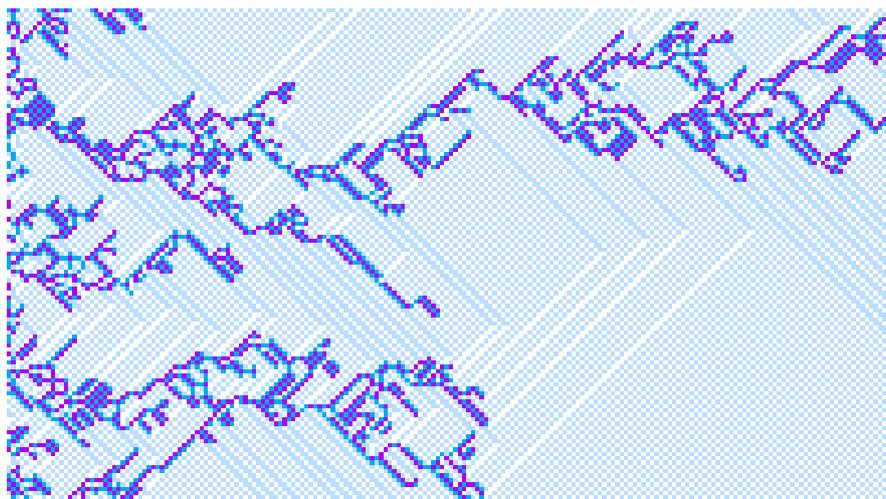


FIG. 3.5 – Un diagramme espace-temps de la règle 57 dans le modèle de forçage.

3.3 Conclusion et perspectives

Ce chapitre a étudié une perturbation portant sur les états des automates cellulaires. Le comportement émergent (c'est-à-dire global) suivant est apparu. Pour 68 des 88 règles élémentaires, la plus petite perturbation permet de synchroniser deux configurations. Pour les 20 restantes, on observe un seuil en deçà duquel les deux configurations restent distinctes (les détails de ce modèle font que pour une perturbation assez forte il y a toujours coalescence). Cette classification montre cependant une corrélation avec la classification empirique de Wolfram [Wolfram (1984)] : les règles subissant une transition sont souvent chaotiques ; mais nous avons montré que ce n'est pas toujours le cas.

Percolation dirigée Les expériences sont nécessairement effectuées sur des configurations finies, mais nous avons vérifié que le nombre de cellules n n'influence pas le résultat. Le seul effet dû à la taille finie rencontré est le fait que la densité 0 est un état absorbant, mais cet effet se manifeste après un temps exponentiel. Augmenter n implique seulement une meilleure précision.

Il a été possible à la section 3.2.1 de détecter les règles ayant un comportement intéressant ; toutes les règles de la seconde classe (subissant une transition) ont été vérifiées en détail à la section 3.2.2.

En étudiant plus en détails le comportement près du seuil, nous avons d'une part confirmé les résultats de la première expérience, et d'autre part montré que la transition de phase appartient à la classe d'universalité de la percolation dirigée. Nous obtenons ainsi 20 nouveaux modèles de percolation dirigée très proches les uns des autres. De même que quelques autres modèles [Grassberger (1999), Morelli and Zanette (1998), Rouquier and Morvan (2006)], la limite du régime sous-critique (quand les états actifs meurent) n'est pas un unique état absorbant ni un ensemble de points fixes, mais une dynamique non figée.

Perspectives Ce chapitre ouvre quelques questions. Il est certainement possible de prouver la transition de phase pour certaines règles, ce qui ouvrirait l'espoir de mieux comprendre cette classe d'universalité.

Ensuite, nos observations suggèrent qu'il y a également une transition pour le temps de coalescence, mais les mesures sont rendues difficiles par le niveau de bruit, plus haut que pour les mesures déjà effectuées.

Il y a dans ce modèle une symétrie entre les deux configurations considérées. Si le but est bien de contrôler l'une des deux, il serait plus naturel d'étudier un schéma « maître / esclave » : quand le tirage au sort indique de rendre les deux états identiques, c'est alors systématiquement l'état du maître qui est copié vers l'esclave. Ce qui revient à briser la symétrie. La configuration maître n'est donc pas perturbée et l'on tente de contrôler la configuration esclave avec un nombre limité de perturbations.

Pour terminer, une généralisation classique est d'étudier le phénomène sur des automates cellulaires avec plus d'états, plus de voisins et en plus grande dimension. Notamment, nous avons vu sur la règle 57 que nous ne pouvons pas espérer que l'étape de synchronisation rapproche toujours de la coalescence. Un exemple plus franc de ce phénomène, et la clef de sa compréhension, se trouvent peut-être dans ces automates cellulaires avec de plus grands paramètres. Enfin, on pourrait imaginer des modèles où plus de deux configurations sont « couplées », ou de mélanger ce couplage avec celui du chapitre suivant.

Chapitre 4

Couplage par les mises à jour

Le chapitre précédent étudiait une perturbation de l'état de la cellule. Ici nous perturbons l'écoulement du temps : les cellules ne sont plus mises à jour de façon synchrone. Nous étudions un effet particulier de cet asynchronisme.

Ce travail s'inscrit dans la continuité des travaux de [Fatès (2004)], qui a étudié notamment des automates cellulaires en dynamique partiellement asynchrone : à chaque étape on tire au sort une partie des cellules, et seules ces cellules sont mises à jour. Il a observé que, pour quelques règles, la configuration après un régime transitoire ne dépendait pas de la configuration initiale. En d'autres termes, en considérant deux configurations initiales aléatoires, puis en appliquant la même séquence de mises à jour (à chaque étape les *mêmes* cellules sont mises à jour dans les deux configurations), on observe pour certaines règles que les deux configurations deviennent identiques : elles sont dans le même état *à la même étape de temps*. Elles sont alors définitivement identiques. Nous appelons ce phénomène *coalescence*.

Cette coalescence apparaît dans des cas triviaux, par exemple si l'automate cellulaire converge toujours vers un unique point fixe : le couplage entre les deux configurations n'a alors pas d'importance, car il suffit d'attendre que les deux configurations aient chacune indépendamment atteint ce point fixe et elles sont alors identiques. Mais ce phénomène apparaît aussi dans des cas non triviaux, où l'orbite asymptotique n'est pas périodique (ou a une période très grande par rapport au nombre de cellules). Le présent chapitre étudie ce phénomène émergent.

Une autre façon de considérer ce travail est de le voir comme une étude de l'influence du générateur aléatoire : pour certains paramètres, c'est l'histoire des mises à jour qui détermine complètement l'évolution du système, indépendamment de la configuration initiale. Nous nous attachons alors à distinguer quels sont ces paramètres : quelles règles, et quelle force de couplage font apparaître ce comportement. Cette idée de synchroniser deux systèmes par une source commune de perturbations aléatoires (i.e. par la même réalisation du bruit) se trouve déjà dans [Pikovskii (1984)], [Maritan and Banavar (1994)] et [Kaulakys, Ivanauskas, and Mekauskas (1999)].

Plan du chapitre Nous commençons par prouver la coalescence rapide pour certaines règles, l'absence de coalescence rapide pour d'autres, et montrons qu'il existe une infinité d'automates cellulaires rapidement coalescents « non triviaux » dans la section 4.1. Dans la section 4.2, nous étudions expérimentalement tous les automates cellulaires élémentaires, ce qui nous amène à proposer une classification en fonction du comportement face à cette perturbation. Enfin, comme au chapitre précédent, il apparaît un phénomène

de transition de phase. À nouveau, la plupart des règles (mais pas toutes) satisfont les conditions de la conjecture de Grassberger (section 2.2) et nous vérifions à la section 4.2.2 que ces règles sont dans la classe d'universalité de la percolation dirigée.

Ce travail a été publié dans [Rouquier and Morvan (2009)].

4.1 Étude formelle

Le modèle Nous commençons par formaliser les observations de Fatès. On considère deux configurations initiales, que l'on fait évoluer avec la *même* histoire $\{M(\cdot, t) \mid t \in \mathbb{N}\}$. En d'autres termes, on considère deux copies de l'automate et, à chaque étape, ce sont les mêmes cellules qui sont mises à jour. Puisque la seule utilisation du hasard se trouve dans le choix des cellules à mettre à jour, ce modèle revient à utiliser la même source d'aléatoire dans les deux copies pour la dynamique.

Nous pouvons maintenant formaliser la notion d'automate rapidement coalescent. Informellement, un automate rapidement coalescent est un automate qui coalesce « spontanément », c'est-à-dire par le faible couplage consistant à faire les mêmes mises à jour sur chacune des configurations, et non par le procédé assez « brutal » du chapitre 3.

Remarque. On peut définir la force de couplage par le nombre de cellules dont on empêche la mise à jour dans les deux configurations. Ainsi, si presque toutes les cellules sont mises à jour dans les deux configurations, il y a peu d'information partagée, le couplage est faible. Si la moitié des cellules sont mises à jour, le couplage est assez fort. Mais il n'y a pas de couplage très fort dans ce modèle, contrairement au chapitre précédent où l'on disposait de toute la gamme, d'un couplage nul à un couplage total (forcer toutes les cellules à être identiques).

Définition 6. Un automate cellulaire asynchrone fini à n cellules est *rapidement coalescent* si, pour tout couple de configurations sauf un nombre négligeable, appliquer la même séquence de mise à jour conduit les deux configurations à être identiques en un nombre d'étapes dont l'espérance est un polynôme en n .

Rappelons que le nombre de couples de configurations « gênantes » (i.e. pour lesquelles l'automate n'est pas rapidement coalescent) est *négligeable* devant le nombre total de couples $(|Q|^n)^2$ si le rapport des deux tend vers 0 en $n \rightarrow \infty$.

Le terme « coalescent » (plutôt que synchronisant) a été choisi pour éviter toute confusion avec le synchronisme des mises à jour.

Un point important de cette définition est que la coalescence intervienne « rapidement », ce qui est formalisé par un temps polynomial. La section suivante (4.1.1) contient l'exemple d'un automate qui ne coalesce pas en temps polynomial, mais néanmoins coalesce toujours si on lui en laisse le temps.

4.1.1 Preuves de non-coalescence rapide

Dans cette section nous prouvons la non-coalescence rapide pour certains automates. Il est évident que l'automate élémentaire 204, qui est la règle identité, n'est pas rapidement coalescent : les cellules en désaccord le sont définitivement. Mais nous allons voir dans cette section deux exemples plus intéressants.

Le premier est celui de la règle 142 qui n'est pas rapidement coalescente non plus, mais pour laquelle la densité évolue.

Proposition 7. *Pour chacune des deux dynamiques asynchrones, la règle 142 n'est pas coalescente.*

Démonstration. Dans la règle 142, une cellule ne peut changer d'état que si la configuration locale est 001 ou 110. Les deux cellules voisines ne peuvent être dans l'une de ces configurations locales et ne peuvent donc changer d'état. Ainsi, le nombre de zones de 1 consécutifs (ou de façon équivalente, le nombre de motifs 01) est constant. Il ne peut y avoir coalescence que si cette quantité est identique dans les deux configurations initiales, situation dont la probabilité tend vers 0 quand n croît. ■

Une règle encore plus intéressante serait une règle qui n'est pas rapidement coalescente à cause du temps trop long qu'il lui faut pour que deux configurations s'accordent, mais qui néanmoins converge toujours vers deux configurations identiques. Dans le reste de cette section 4.1.1, nous montrons que c'est le cas de la règle 60. La preuve utilise des propriétés spécifiques des règles affines, que nous commençons par introduire.

Définition 8. Pour l'ensemble d'état $Q = \{0, 1\}$, une règle est *affine* si elle s'exprime sous la forme

$$\delta(q_1, \dots, q_{|V|}) = \varepsilon + \sum_{i \in I} q_i \pmod{2}$$

où $I \subseteq \llbracket 1, |V| \rrbracket$ et $\varepsilon \in \{0, 1\}$.

Les règles affines ont été introduites par [Martin, Odlyzko, and Wolfram (1984)], et étudiées en détail dans [Chaudhuri, Chowdhury, Nandi, and Chattopadhyay (1997)] par exemple.

Lemme 9. *Pour les dynamiques totalement asynchrone et partiellement asynchrone, les règles affines sont exactement celles pour lesquelles le diagramme espace-temps d'accord est le diagramme espace-temps d'un automate cellulaire.*

Nous appelons la règle régissant ce dernier diagramme la règle *quotient* de la règle d'origine.

Démonstration. Pour montrer le sens direct, considérons une règle affine et deux configurations. Nous calculons l'état d'accord d'une cellule (le fait que les deux configurations soient identiques ou non en cette cellule) résultant d'une transition. Nous montrons alors que cet état peut être calculé directement en appliquant une règle d'automate cellulaire sur la configuration d'accord à l'étape de départ.

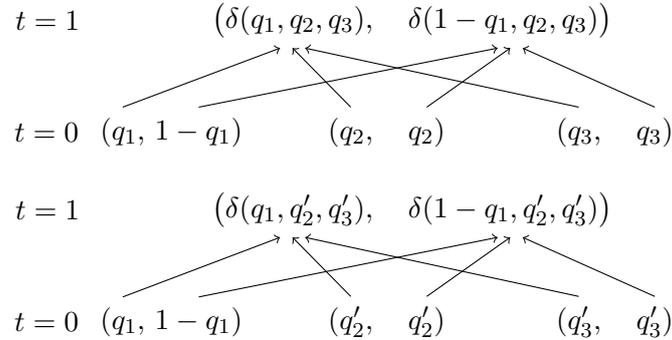
Soient $(q_i)_{i \in \llbracket 1, |V| \rrbracket}$ les états d'une configuration locale et $(q'_i)_{i \in \llbracket 1, |V| \rrbracket}$ ceux de l'autre, l'état d'accord est 1 si $q_i = q'_i$ et 0 sinon, que nous notons $q_i \oplus q'_i$. Dans la configuration de l'étape suivante, l'état d'accord est

$$\delta(q_1, \dots, q_{|V|}) \oplus \delta(q'_1, \dots, q'_{|V|}) = \varepsilon + \sum_{i \in I} q_i + \varepsilon + \sum_{i \in I} q'_i \pmod{2} = \sum_{i \in I} (q_i \oplus q'_i) \pmod{2}$$

ce qui est précisément le résultat de l'application d'une règle d'automate δ' sur les états de la configuration d'accord à l'étape précédente. On lit sur cette équation que la règle quotient δ' est $\delta'(q_1, \dots, q_{|V|}) = \sum_{i \in I} q_i \pmod{2}$, c'est-à-dire la partie linéaire de δ .

Réciproquement, supposons que le diagramme d'accord est celui d'un automate δ . Si la règle δ est constante, $I = \emptyset$ convient. Sinon, il existe une configuration $(q_1, \dots, q_{|V|})$ pour laquelle δ change si un q_i donné change. Par symétrie, on peut supposer qu'il s'agit de q_1 , i.e. $\delta(q_1, q_2, \dots, q_{|V|}) \neq \delta(1 - q_1, q_2, \dots, q_{|V|})$. Supposons par l'absurde qu'il existe

$q'_2, \dots, q'_{|V|}$ tels que $\delta(q_1, q'_2, \dots, q'_{|V|}) = \delta(1 - q_1, q'_2, \dots, q'_{|V|})$. Nous avons alors deux couples de configurations avec la même configuration locale d'accord à l'étape 0, mais un état d'accord différent à l'étape 1. Cela est illustré sur la figure suivante, avec trois états :



C'est une contradiction avec le fait que le diagramme d'accord est toujours le diagramme espace-temps d'un automate. Ainsi, $\delta(q_1, q'_2, \dots, q'_{|V|}) \neq \delta(1 - q_1, q'_2, \dots, q'_{|V|})$ pour tous $q'_2, \dots, q'_{|V|}$. La règle s'écrit donc sous la forme $\delta(q_1, q_2, \dots, q_{|V|}) = \varepsilon + q_1 + f(q_2, \dots, q_{|V|}) \pmod{2}$.

Puisque cela est vrai pour tout voisin pouvant influencer le résultat de δ , le résultat suit. ■

Proposition 10. *Pour chacune des deux dynamiques asynchrones, la règle 6o n'est pas rapidement coalescente.*

Démonstration. La règle 6o signifie « XOR entre moi et mon voisin de gauche ». Elle est affine, on peut donc raisonner sur le diagramme espace-temps d'accord. La règle quotient est 6o.

Distinguons deux cas pour une configuration globale :

- Si toutes les cellules sont en désaccord, la coalescence apparaît si et seulement si toutes les cellules sont mises à jour en même temps. Cela se produit avec probabilité α^n .
- Sinon, il existe une cellule en désaccord dont la voisine de gauche est en accord. Qu'elle soit mise à jour ou non, cette cellule sera toujours en désaccord à l'étape suivante.

La seule possibilité d'atteindre la coalescence est donc de passer par la configuration où toutes les cellules sont en désaccord. L'espérance du nombre de passages par cette configuration est $\frac{1}{\alpha^n}$, l'espérance du temps de coalescence est donc au moins $\frac{1}{\alpha^n}$, ce qui est asymptotiquement plus grand que tout polynôme en n car $0 < \alpha < 1$. ■

La règle 6o n'est pas rapidement coalescente, mais cependant elle converge toujours vers deux configurations identiques. En effet, depuis toute configuration, il est possible (même si cela est peu probable) d'atteindre la configuration où toutes les cellules sont en accord. Pour cela, il suffit de mettre à jour les cellules en accord ayant une voisine gauche en désaccord (ces cellules passent dans l'état de désaccord), et de répéter jusqu'à ce toutes les cellules soit en désaccord. Mettre alors à jour toutes les cellules en même temps fait apparaître la coalescence. Ainsi, à chaque étape, il y a une probabilité non nulle de suivre un chemin vers la coalescence.

4.1.2 Preuves de coalescence rapide

Nous avons montré dans la section précédente que certaines règles ne sont pas rapidement coalescentes. Nous prouvons ici que certaines règles le sont. Un cas trivial de règle rapidement coalescente est le cas des règles nilpotentes. Une règle est *nilpotente* si, quelle que soit la configuration initiale, l'orbite converge vers une configuration dont toutes les cellules sont dans le même état. Un automate nilpotent est bien sûr rapidement coalescent si cette convergence se fait en temps polynomial.

Pour la plupart des automates élémentaires captifs¹, le caractère nilpotent ou non ainsi que le temps de convergence ont été établis dans l'article [Fatès (2004)] (pour la dynamique totalement asynchrone) puis (pour la dynamique partiellement asynchrone) dans [Fatès, Regnault, Schabanel, and Thierry (2006)]. Ces deux références nous permettent donc d'affirmer immédiatement que les règles suivantes sont rapidement coalescentes, pour les deux dynamiques : 128, 130, 136, 138, 152, 160, 162, 168.

Mais le caractère nilpotent de ces règles les rend peu intéressantes du point de vue de la coalescence. Nous allons voir, et c'est ce qui donne de l'intérêt à cette étude, qu'il existe des automates rapidement coalescents non nilpotents.

Automates rapidement coalescents non nilpotents

Dans cette section nous montrons qu'il y a une infinité d'automates rapidement coalescents non nilpotents. Pour cela, nous montrons la coalescence rapide pour deux règles particulières, puis construisons une suite infinie d'automates rapidement coalescents.

Une façon simple de réaliser le second point serait d'augmenter un automate en ajoutant des états qui ont toujours pour image un état de l'automate d'origine, quels que soient les voisins. Cette transformation est cependant artificielle car elle produit un automate qui est en un sens identique à celui d'origine. Pour éviter cela, nous nous intéressons aux automates *état-minimaux* : les automates dans lesquels tout état peut être atteint (mais pas nécessairement toute configuration). Parmi les automates élémentaires, seules les règles 0 et 255 ne sont pas état-minimales.

Nous prouvons donc l'existence de deux automates état-minimaux rapidement coalescents (proposition 11); puis grâce à eux nous déduisons l'existence d'une infinité de tels automates (théorème 12); enfin nous décrivons le comportement de deux autres automates élémentaires vis-à-vis de la coalescence. (proposition 13).

Proposition 11. *Pour la dynamique totalement asynchrone, pour n impair, les automates cellulaires élémentaires 6 et 7 sont rapidement coalescents.*

La preuve de cette proposition est spécifique à ces règles et assez technique, elle peut éventuellement être ignorée en première lecture.

Démonstration. Appelons *nombre de zones* le nombre de motifs 01 dans une configuration. C'est aussi le nombre de blocs, ou zones, de 1 consécutifs. Commençons par considérer une seule configuration (et non un couple).

La table 4.1 page ci-contre montre la table de transition de la règle 6. Puisque les cellules sont mises à jour une par une, et puisque la mise à jour de la cellule centrale des configurations locales 101 ou 010 ne change pas son état, les zones ne peuvent pas fusionner. Le nombre de zones ne peut donc décroître.

¹Un automate cellulaire *captif* est un automate cellulaire où le résultat de la règle de transition est toujours l'un des états de la configuration locale. Pour les automates élémentaires, il s'agit donc des règles δ telles que $\delta(0, 0, 0) = 0$ et $\delta(1, 1, 1) = 1$, parfois appelées « doublement quiescentes ».

TAB. 4.1 – Table de transition de l'automate élémentaire 6.

Voisins	111	110	101	100	011	010	001	000
Nouvel état	0	0	0	0	0	1	1	0

TAB. 4.2 – Une suite de mises à jour intéressantes pour la règle 6.

$$\begin{array}{c}
 \dots 0 \underbrace{001} \dots \\
 \dots \underbrace{001} 1 \dots \\
 \dots 0 \underbrace{111} \dots \\
 \dots 0 1 0 1 \dots
 \end{array}$$

Montrons tout d'abord que le nombre de zones croît jusqu'à ce qu'il n'y ait plus de motifs 000 ou 111. Pour chaque motif 111, la cellule centrale peut être mise à jour avant ses voisines (ce qui mène au motif 101) avec probabilité $\frac{1}{3}$, et en un temps d'espérance n . Pour chaque motif 0001, la séquence de la table 4.2 est possible. Elle se déroule sans autre mise à jour des quatre cellules avec probabilité $1/4^3$ et en temps d'espérance $3n$. Ainsi, tant qu'il y a des motifs 000 ou 111, le nombre de zones croît d'une unité en un temps d'espérance au plus $O(n)$. Puisqu'il y a $O(n)$ zones, la durée totale de cette première phase est $O(n^2)$.

Considérons maintenant la configuration comme une concaténation de mots sur l'alphabet $\{0, 1\}^*$, en plaçant une séparation entre deux mots au centre de chaque motif 00 ou 11. Nous obtenons ainsi une suite de mots qui n'ont jamais deux lettres consécutives identiques, chaque mot étant de longueur au moins deux. (Ce sont donc des mots reconnus par l'expression régulière « $(01)^+0? \mid (10)^+1? \rangle$ ».)

Montrons que les frontières entre ces mots suivent une marche aléatoire unidirectionnelle (vers la droite) et se rencontrent, auquel cas un mot disparaît avec probabilité non nulle. L'automate évolue donc vers une configuration où il n'y a qu'un seul mot. Mettre à jour la cellule centrale du motif 100 ne change pas son état, donc les frontières ne peuvent pas se déplacer vers la gauche de plus d'une cellule. À l'opposé, mettre à jour la cellule centrale des motifs 001 ou 110 déplace la frontière. Une étape de cette marche aléatoire prend un temps d'espérance $O(n)$. La longueur d'un mot suit donc une marche aléatoire non biaisée (contrairement à la précédente qui ne marche que vers la droite). Cette longueur atteint 1 après (en moyenne) $O(n^3)$ étapes, ce qui mène au motif 000 ou 111. Avec une probabilité non nulle, ce motif disparaît comme détaillé dans la phase de croissance du nombre de zones. L'espérance du temps pour que $O(n)$ mots disparaissent est donc $O(n^4)$.

Une fois qu'il n'y a plus qu'un mot, puisque n est impair, les lettres aux deux extrémités du mot sont identiques, c'est-à-dire qu'il existe un unique motif 00 ou 11. Ce motif suit toujours la marche aléatoire. Ce motif change la phase dans la configuration : d'un côté (resp. de l'autre) les cellules de coordonnée paire sont dans l'état 1 (resp. 0), de l'autre ce sont les cellules de coordonnée impaire.

Considérons à nouveau un couple de configurations, qui a atteint ce stade : les deux configurations sont de la forme $(01)^*1$ ou $0(01)^*$. À cause du changement de phase, le motif 00 ou 11 est une frontière entre une région où les deux configurations sont en accord et une où elles sont en désaccord. Le motif 00 ou 11 de l'autre configuration est une seconde frontière.

Étudions la longueur de l'unique région où les deux configurations sont en désaccord.

TAB. 4.3 – Transitions intéressantes sur des motifs de la règle 6.

... 10100101 01010010 01011010 ...
... 01011010 01011010 01001010 ...
a.	b.	c.

Cette longueur suit une marche aléatoire non biaisée déterminée par les mouvements des deux motifs 00 ou 11. Quand cette longueur atteint n , comme dans la table 4.3.a, la seule mise à jour qui change l'état d'une cellule est celle de la quatrième cellule, ce qui diminue la longueur de la région de désaccord. La marche aléatoire ne reste donc pas indéfiniment dans l'état n .

À l'opposé, quand cette longueur atteint 1, une possibilité est celle de la table 4.3.b, où la mise à jour de la cinquième cellule accorde les deux configurations. La seule autre possibilité est la table 4.3.c, où la mise à jour de la cinquième cellule puis la quatrième mène à la configuration de la table 4.3.b. Dans chaque cas, lorsque la longueur de la zone de désaccord atteint 1, la coalescence intervient avec une probabilité constante non nulle. Une étape de cette marche aléatoire prend un temps d'espérance $O(n)$, l'espérance du temps total est donc $O(n^3)$.

Puisque chaque étape prend un temps d'espérance polynomiale, la règle 6 est rapidement coalescente.

La preuve pour la règle 7 est identique, mis à part le fait que la configuration locale 000 mène à 010. Cela n'affecte pas la preuve, simplement la phase de croissance est plus rapide. ■

Notons que nous avons montré que les règles 6 et 7 ne sont pas nilpotentes.

Remarque. Si n est pair, la preuve est valide jusqu'à ce qu'il n'y ait plus qu'un mot. La configuration n'a alors aucun des motifs 00 ou 11. Cette configuration est donc $(01)^{n/2}$ ou $(10)^{n/2}$.

Si les deux configurations ont la même parité, il y a coalescence, sinon les deux sont en parfait désaccord (définitivement). Les deux cas se produisent expérimentalement.

Théorème 12. *Dans la dynamique totalement asynchrone, il existe des automates cellulaires rapidement coalescents non nilpotents état-minimaux avec un nombre d'état arbitrairement grand, donc une infinité d'automates cellulaires rapidement coalescents non nilpotents état-minimaux.*

Démonstration. Commençons par définir le carré d'un automate. Soit donc un automate cellulaire $\mathcal{A} = (Q, d, V, \delta, n, \mu)$; le carré de \mathcal{A} noté \mathcal{A}^2 est l'automate cellulaire $(Q^2, d, V, \delta^2, n, \mu)$ où

$$\delta^2((q_1, q'_1), \dots, (q_{|V|}, q'_{|V|})) := (\delta(q_1, \dots, q_{|V|}), \delta(q'_1, \dots, q'_{|V|}))$$

Intuitivement, \mathcal{A}^2 est l'automate obtenu en superposant deux configurations de \mathcal{A} et en faisant évoluer les deux selon δ , mais avec les mêmes mises à jour $\{M(\cdot, t) \mid t \in \mathbb{N}\}$.

Si \mathcal{A} est état-minimal ou non nilpotent, \mathcal{A}^2 l'est également.

Considérons maintenant \mathcal{A} un automate rapidement coalescent. Alors \mathcal{A}^2 converge en temps d'espérance polynomiale vers une configuration dont tous les états sont dans l'ensemble $\{(q, q) \mid q \in Q\}$. À partir de ce moment, \mathcal{A}^2 « simule » \mathcal{A} , au sens où l'on a le diagramme commutatif suivant :

$$\begin{array}{ccc}
Q^{\mathcal{U}} & \xrightarrow{\delta_{\mathcal{A}}} & Q^{\mathcal{U}} \\
\updownarrow & & \updownarrow \\
\{(q, q) \mid q \in Q\}^{\mathcal{U}} & \xrightarrow{\delta_{\mathcal{A}^2}} & \{(q, q) \mid q \in Q\}^{\mathcal{U}}
\end{array}$$

\mathcal{A}^2 est donc rapidement coalescent, avec un temps d'espérance au plus deux fois plus long. Il en va de même pour $(\mathcal{A}^2)^2$, $((\mathcal{A}^2)^2)^2$, etc, ce qui forme une suite infinie d'automates dont le nombre d'états est strictement croissant. ■

Nous avons donc obtenu comme annoncé une infinité d'automates rapidement coalescents. Le cas des deux règles suivantes mérite d'être remarqué :

Proposition 13. *Pour chacune des deux dynamiques asynchrones, les automates cellulaires élémentaires 15 et 170 soit rapidement coalescent, soit atteignent rapidement une configuration de désaccord total, chaque cas intervenant avec probabilité $\frac{1}{2}$ si la configuration initiale est tirée au hasard uniformément.*

Démonstration. La règle 170 signifie « copier l'état du voisin de droite », c'est donc le décalage ou *shift*. C'est une règle affine et la règle quotient est 170. Il en va de même pour la règle 15 qui signifie « prendre l'état opposé à celui du voisin de droite » : la règle quotient est toujours 170.

Enfin, sous chacune des deux dynamiques asynchrones, la règle 170 converge en temps d'espérance polynomiale vers 0^n (ce qui correspond à la coalescence) ou vers 1^n (désaccord total) [Fatès, Regnault, Schabanel, and Thierry (2006)]. Le second cas a une probabilité proportionnelle à la densité de 1 dans la configuration initiale. Puisque la configuration initiale est aléatoire, le résultat suit. ■

4.2 Étude expérimentale et transition de phase

Dans cette section, nous exposons des résultats expérimentaux obtenus pour la dynamique partiellement asynchrone. Nous verrons qu'environ la moitié des automates élémentaires sont rapidement coalescents (parfois cela dépend du taux de synchronisme α), puis proposons une classification plus fine (section 4.2.1).

L'une de ces classes contient les automates subissant une transition de phase quand α varie. Nous montrons alors expérimentalement que cette transition de phase est dans la classe d'universalité de la percolation dirigée (section 4.2.2).

4.2.1 Classification des automates vis-à-vis de la coalescence

Protocole

Comme au chapitre 3, un *lancer* est l'orbite d'un automate lorsque tous les paramètres sont fixés : règle, nombre de cellules n , probabilité α et configuration initiale. Un lancer est stoppé lorsque les deux configurations sont identiques (ce qui est définitif), ou bien lorsqu'un nombre d'étapes fixé à l'avance est atteint. Voici les paramètres utilisés.

1. Nombre de cellules n . Pas de changement par rapport au point 1 page 31.
2. Nombre d'étapes de calcul : idem. C'est le point 2.
3. Probabilité de mise à jour α . Ici encore nous échantillons tout l'intervalle $[0; 1]$ (et nous allons d'ailleurs trouver des transitions réparties sur cette intervalle). Pour

chacune des 88 règles, on effectue 999 lancers : un pour chaque valeur de α de 0,001 à 0,999.

Ici encore, la régularité des courbes sur les figures 4.1 page suivante et 4.2 page 50 montre qu'il n'est pas nécessaire de moyenner $\rho_\infty(\alpha)$ sur plusieurs lancers à α fixé. L'initialisation du générateur aléatoire est différente pour chaque valeur de α . L'importance de ce point est justement le sujet de cette étude : nous étudions quelle est l'influence de l'aléa dans les mises à jour.

4. La configuration initiale est aléatoire (tirage uniforme parmi toutes les configurations) et différente pour chaque valeur de α .

Résultats

Nous obtenons les classes empiriques suivantes. La table 4.4 donne la composition de chaque classe.

TAB. 4.4 – Classification empirique des automates élémentaires selon le comportement face à la coalescence rapide.

a/ pas de coalescence rapide	4, 5, 12, 13, 25, 28, 29, 33, 36, 37, 41, 44, 45, 51, 54, 60, 72, 73, 76, 77, 78, 90, 94, 104, 105, 108, 122, 132, 140, 142, 150, 156, 164, 172, 200, 204, 232
b/ coalescence triviale (automate nilpotent)	0, 2, 8, 10, 24, 32, 34, 38, 40, 42, 56, 74, 128, 130, 134, 136, 138, 152, 160, 162, 168.
c/ coalescence non triviale	3, 19, 35, 46, 154
d/ combinaison de certains des cas a/, b/ et c/	6, 18, 26, 106, 146, 50, 1, 9, 11, 27, 57, 62, 110, 126, 58
e/ accord ou désaccord total	14, 15, 23, 43, 170, 178, 184
f/ combinaison des cas précédents	22, 30, 7

a/ Certaines règles ne coalescent pas (ou prennent pour cela un temps trop long pour que ce soit observable). On peut être un peu plus précis en traçant la densité asymptotique de cellules en désaccord ρ_∞ en fonction de α . Pas de coalescence signifie que cette courbe est toujours au-dessus de zéro. Dans la plupart des cas cette courbe est horizontale, souvent autour de 0,5, légèrement bruitée (figure 4.1.a). Ce bruit montre la précision de la mesure, en comparaison avec la figure 4.1.b. Cette dernière est la courbe obtenue pour la règle 204, qui est l'identité : pour cette règle, la densité asymptotique est la même que la densité initiale, la courbe montre donc la variance due à la configuration initiale.

On note que la règle 150 (figure 4.1.c) fait tendre la densité vers 0,5 : la courbe est moins bruitée. Notons au passage que ce fait semble difficile à prouver. Les figures 4.1.d et 4.1.e montrent des exemples où la courbe n'est pas une ligne horizontale.

La règle 142 est un cas à part. Nous avons montré à la proposition 7 page 42 qu'elle n'est pas rapidement coalescente. Expérimentalement, il n'y a pas de densité asymptotique : la densité évolue indéfiniment. Il y a cependant des corrélations entre les deux configurations : on observe de larges zones d'accord et de larges zones de désaccord.

b, c/ Certaines règles sont rapidement coalescentes.

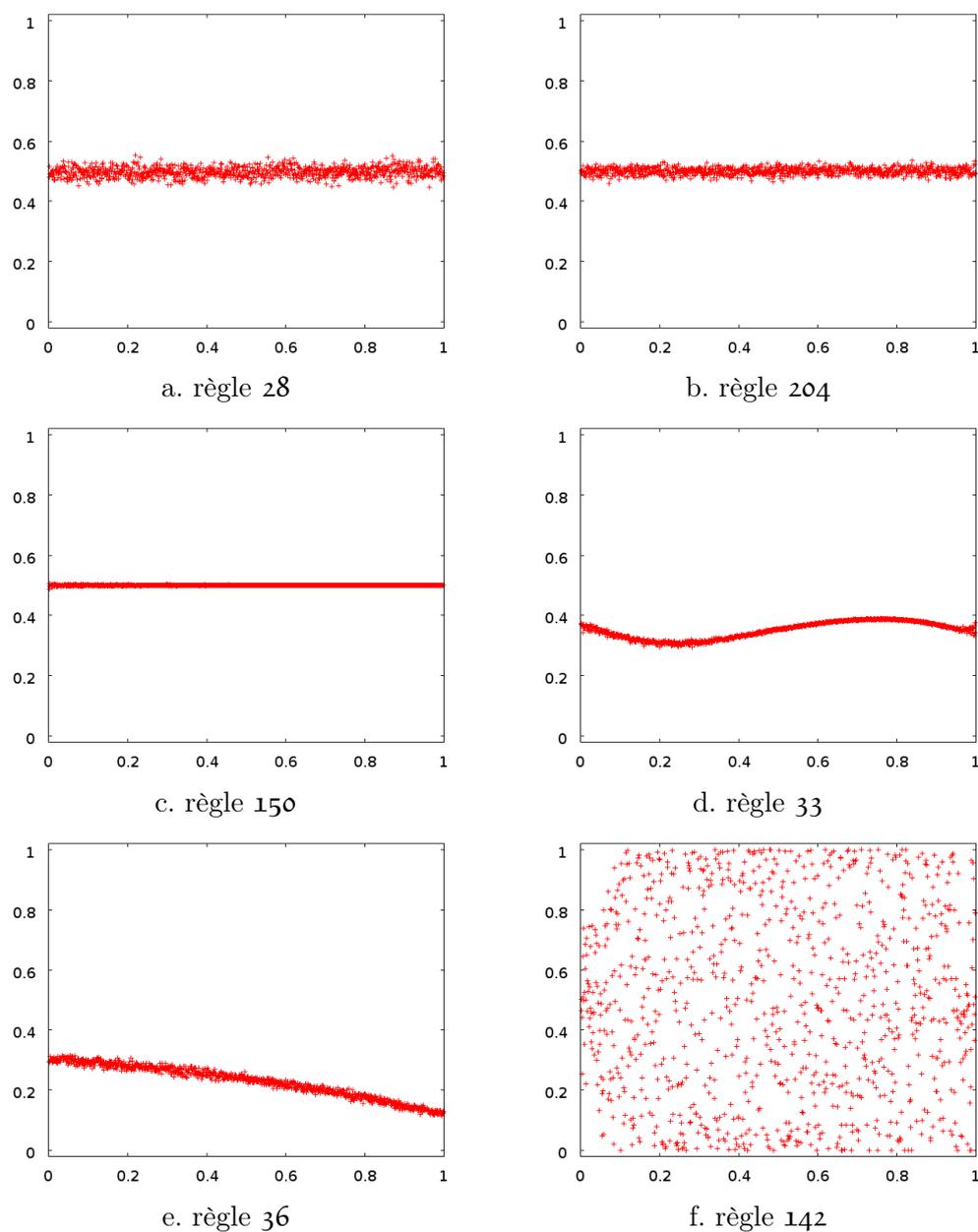


FIG. 4.1 – Densité asymptotique de cellules en désaccord $\rho_\infty(\alpha)$ pour quelques règles expérimentalement non rapidement coalescentes. L'échelle de chacun des axes est $[0; 1]$. Le premier cas est le cas typique.

- b/ Le cas trivial est celui où la règle converge vers un point fixe unique. On peut alors considérer les deux configurations indépendamment et attendre qu'elles aient chacune atteint ce point fixe. Il y a alors coalescence.
- c/ Il y a cinq règles expérimentalement non nilpotentes et coalescentes. Parmi elles, la règle 19 converge particulièrement lentement pour α petit.
- d/ Certaines règles combinent les comportements précédents, selon la valeur de α (figure 4.2). 6, 18, 26, 106, 146 combinent a/ et b/ ; 50 combine b/ et c/ ; 1, 9, 11, 27, 57, 62, 110, 126 combinent a/ et c/ (voir figure 4.4 page 52) ; 58 combine a/, b/ et c/.

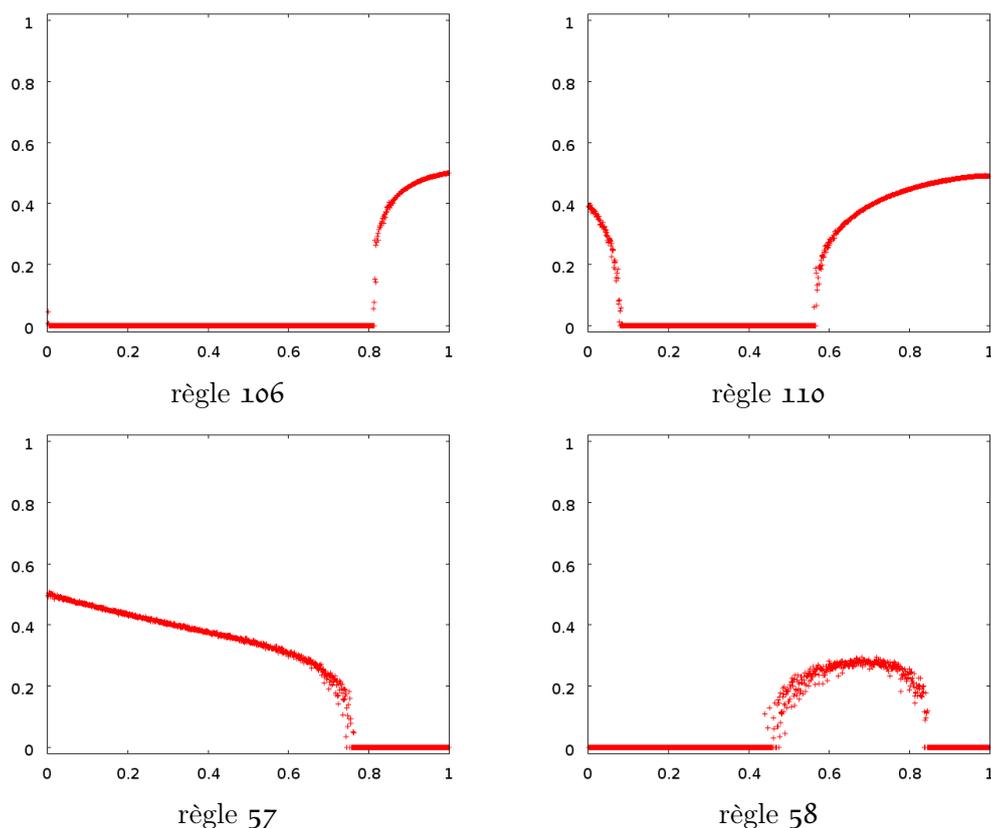


FIG. 4.2 – Densité asymptotique de cellules en désaccord $\rho_\infty(\alpha)$ pour quelques règles rapidement coalescentes ou non selon α . Le premier cas est le cas le plus courant (50 % des règles). Ces règles, qui subissent une transition de phase, sont étudiées plus avant à la section 4.2.2.

- e/ Sept règles convergent vers soit l'accord total entre les deux configurations, soit un désaccord total, selon la configuration initiale et les mises à jour. La courbe $\rho_\infty(\alpha)$ devrait donc être constituée de deux lignes droites, l'une en $\rho_\infty = 0$ et l'autre en $\rho_\infty = 1$. Mais ces règles mettent un temps plus long que les autres à atteindre leur comportement asymptotique, c'est pourquoi la figure est peu précise (figure 4.3 page suivante). Nous avons vérifié que ce comportement décrit est bien celui observé pour quelques valeurs de α .
- Ces règles prennent un temps encore plus long (peut-être même arbitrairement long) à l'approche d'une certaine valeur de α , d'où la figure en « nœud de papillon » (figure 4.3.b). Plus α est proche d'une certaine valeur critique, plus la convergence

à partir de $\rho = 0,5$ est lente. Différentes valeurs critiques de α sont observées : faible α (figure 4.3.a), $\alpha \simeq 0,5$ (figure 4.3.b) ou ailleurs (figure 4.3.c).

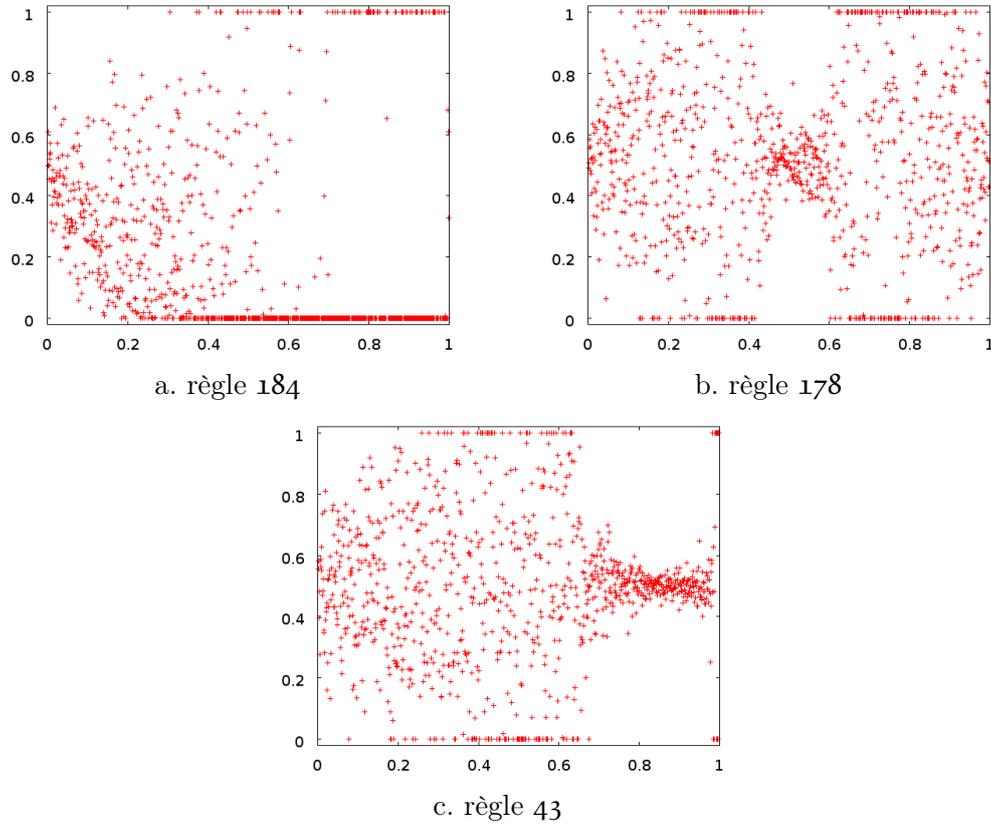


FIG. 4.3 – Densité asymptotique de cellules en désaccord $\rho_\infty(\alpha)$ pour quelques règles convergeant vers accord ou désaccord total, selon α .

f/ Les règles 22 et 30 combinent le point précédent (pour α petit) avec le point a/ ; la règle 7 le combine (pour α petit) avec c/. On peut noter que la règle 30 montre également une densité asymptotique qui varie quand elle effectue une transition entre les points a/ et e/. Mais à cause des difficultés mentionnées au point e/, cette densité n'a pas été mesurée.

4.2.2 Percolation dirigée dans ce modèle

Nous nous concentrons maintenant sur les règles du point d/, qui ont la particularité de coalescer rapidement ou non suivant α . Nous étudions maintenant cette transition de phase.

La percolation dirigée a été présentée à la section 2.2. Les sites actifs sont ici les cellules où les deux configurations sont en désaccord. La densité de tels sites est notée $\rho(\alpha, t)$, ou $\rho_\infty(\alpha)$ pour la densité asymptotique. L'ensemble absorbant est constitué des paires de configurations identiques. La transition (coalescence rapide ou non) apparaît en variant α , voir figure 4.4 page suivante. Le but est donc d'identifier δ puis β sous l'hypothèse $\rho_\infty(\alpha) \propto |\alpha - \alpha_c|^\beta$ pour un certain α_c . Noter qu'il n'y a pas a priori de relation directe entre le taux de synchronisme α et la probabilité qu'un lien soit ouvert p dans le modèle de percolation dirigée.

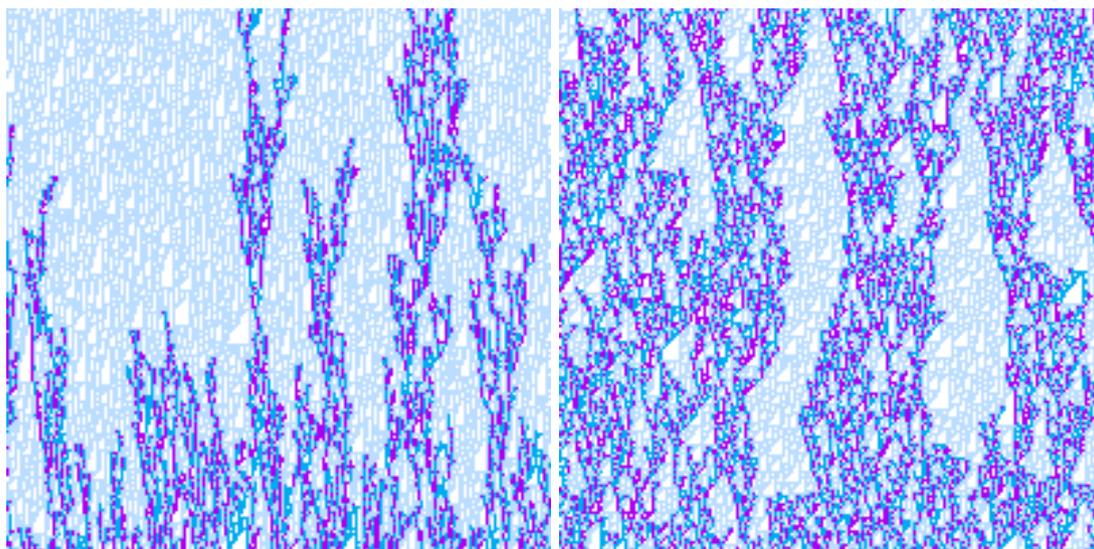


FIG. 4.4 – Règle **110**, $n = 200$. Le temps va de bas en haut, pendant 200 mises à jour. Les sites où les deux configurations sont en désaccord sont sombres, ceux où les deux sont en accord sont clairs. (il y a deux couleurs claires, blanc pour l'état 0, l'autre pour l'état 1). Gauche : régime sous-critique ($\alpha = 0,47 < \alpha_c \simeq 0,566$), les branches meurent. Droite : régime sur-critique ($\alpha = 0,65 > \alpha_c$), les sites actifs se propagent.

Il nous faut introduire une notation. En effet, certaines règles (**9**, **58**, **110**, **126**) subissent deux transitions de phase, une pour α petit, que nous noterons par un ℓ (pour *low*) en indice comme dans $\mathbf{9}_\ell$, une autre pour α grand, notée par un h (*high*) en indice comme dans $\mathbf{9}_h$.

Mesure de α_c

Pour mesurer β , nous utilisons la méthode décrite dans [Hinrichsen (2000)] qui recommande de mesurer d'abord α_c avant de faire une régression linéaire pour mesurer β , au lieu de faire la régression en ajustant α et β en même temps.

Pour mesurer α_c , il s'agit de tracer $\rho(\alpha, t)$ en fonction de t sur une échelle doublement logarithmique et d'ajuster α pour obtenir une ligne droite (pour $\alpha < \alpha_c$, la densité $\rho(\alpha, t)$ tombe à zéro plus rapidement qu'une ligne droite, pour $\alpha > \alpha_c$, elle rejoint une asymptote horizontale strictement positive.).

La pente de la ligne est δ . Les mesures de δ sont portées dans le même tableau que celles de β (table 4.6 page 54).

Concernant les paramètres, la configuration initiale est aléatoire (uniformément distribuée) et l'on vérifie donc entre deux lancers qu'elle n'a pas d'influence. Pour atteindre la précision actuelle, jusqu'à $n = 10^6$ cellules et 10^7 étapes ont été nécessaires. Les résultats sont dans la table 4.5 page ci-contre. On peut remarquer que α_c pour les règles **1** et $\mathbf{126}_\ell$, de même que pour $\mathbf{9}_\ell$ et $\mathbf{110}_\ell$, sont très proches, et sans doute égaux. Noter aussi la nécessité d'échantillonner α le plus largement possible, en effet le seuil α_c pour les règles **6** et **11** est proche de 0 ou 1.

TAB. 4.5 – Valeurs de α_c pour les règles ayant une transition de phase (rappelons que α_c est spécifique à chaque règle et n'est utilisé que pour mesurer β et δ).

règle	1	6	9 _ℓ	9 _h	11	18	26	27	57
$\alpha_c > \dots$	0,102 5	0,06	0,073 8	0,757 5	0,957 5	0,713 8	0,474 7	0,856	0,749
$\alpha_c < \dots$	0,102 8	0,08	0,074 0	0,757 6	0,958 3	0,714 1	0,475 1	0,858	0,750

règle	58 _ℓ	58 _h	62	106	110 _ℓ	110 _h	126 _ℓ	126 _h	146
$\alpha_c > \dots$	0,474 5	0,840 8	0,598 5	0,814 3	0,074	0,566	0,101 5	0,720 3	0,675 0
$\alpha_c < \dots$	0,474 8	0,841 2	0,599 5	0,814 7	0,075	0,567	0,102	0,720 4	0,675 3

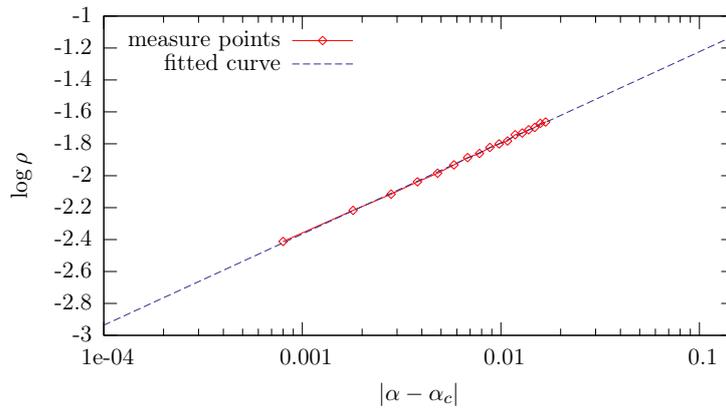


FIG. 4.5 – Mesure de β pour la règle 58_h.

Mesure de β

Pour mesurer β , traçons maintenant $\rho_\infty(\alpha)$ pour α proche de α_c (figure 4.5). On trace en réalité $\log \rho_\infty(\alpha)$ en fonction de $\log(\alpha - \alpha_c)$ puis l'on fait une régression linéaire, la pente de la droite obtenue est une estimation de β . Il est important de faire la régression sur $\log \rho_\infty(\alpha)$ et $\log(|\alpha - \alpha_c|)$, sans quoi (si l'on faisait la régression d'une loi de puissance $\rho_\infty(\alpha) = C|\alpha - \alpha_c|^\beta$) les erreurs pour $\rho_\infty(\alpha)$ petit auraient un poids de quelques ordres de grandeur plus petit et seraient ignorées.

Il est aussi possible d'ajuster α_c pour obtenir une ligne droite. Notre expérience indique que cette méthode a le même rapport temps de calcul/précision que la détermination préalable de α_c .

Le protocole est identique à celui décrit au paragraphe «protocole» de la section 3.2.2.

La régression donne les valeurs de β et δ reportées dans la table table 4.6 page suivante, qui prennent en compte l'imprécision sur α_c et sur les points à conserver pour la régression.

Tous ces chiffres sont compatibles avec la valeur mesurée sur d'autres modèles des exposants critiques de percolation dirigée.

4.2.3 Comparaison à un modèle plus simple

Dans [Fatès (2007)], un problème proche est étudié. Le modèle qui y est étudié est celui, plus classique, où une seule configuration évolue, dans le même régime partielle-

TAB. 4.6 – Valeurs expérimentales des exposants critiques β et δ . Les valeurs expérimentales mesurées sur d'autres systèmes sont $\beta = 0,276$ et $\delta = 0,159$ 5. Les chiffres en police plus petite donnent la précision sur le dernier chiffre.

règle	1	9 _ℓ	9 _h	11	27	57
β	0,272(7)	0,28 ₋₁ ⁺²	0,278(5)	0,29(3)	0,28(2)	0,27 ₋₂ ⁺¹
δ	0,159(1)	0,160(4)	0,160(1)	0,161(3)	0,162(5)	0,159(3)

règle	58 _ℓ	58 _h	62	110 _ℓ	110 _h	126 _ℓ	126 _h
β	0,28 ₋₁ ⁺²	0,26(2)	0,275(6)	0,28(1)	0,276(5)	0,26 ₋₁ ⁺²	0,27(1)
δ	0,160(8)	0,159(2)	0,158(8)	0,160(8)	0,159(8)	0,157(3)	0,160(4)

ment asynchrone. L'article se concentre également sur les automates élémentaires. L'état actif est simplement l'état « 1 », $\{0^n\}$ est l'ensemble absorbant.

Cet article montre expérimentalement que certaines règles (6, 18, 26, 50, 58, 106 et 146) subissent une transition de phase qui appartient à la classe de la percolation dirigée : elle converge vers 0^n si $\alpha < \alpha_c$ pour un certain α_c spécifique à chaque règle, et pour α proche de α_c la densité asymptotique d' de cellules dans l'état 1 est de la forme $C(\alpha - \alpha_c)^\beta$ pour une constante C .

Ces règles sont rapidement coalescentes pour α petit, car les deux configurations convergent indépendamment et en un temps polynomial en n vers 0^n . Ainsi, si l'une de ces règles n'est pas rapidement coalescente pour α grand, elle subira une transition de phase dans notre modèle. Examinons chacune de ces règles :

- La règle 50 coalesce rapidement quel que soit α (parfois car elle est nilpotente, parfois non), elle ne nous intéresse donc pas pour le présent travail. Les autres règles ne coalescent pas rapidement pour α suffisamment grand.
- La règle 58 a un seuil α_c différent dans chacun des deux modèles, elle a été retenue dans l'étude expérimentale de la section 4.2.2 et semble appartenir à la classe de la percolation dirigée.
- La règle 6 a également un seuil α_c différent. Cependant, près de α_c , les deux configurations deviennent rapidement composées de régions périodiques, avec le motif 01 comme période. Ces régions se recouvrent entre les deux configurations. Ces deux configurations sont en accord ou désaccord (selon la parité) sur tout le recouvrement. À l'intérieur d'un tel recouvrement, l'état d'une cellule ne peut pas changer quand on la met à jour. Dès lors, le point 3 page 22 de la conjecture de Grassberger n'est pas rempli : une cellule active isolée ne peut pas toujours devenir inactive. On constate qu'en effet, la courbe $\rho(\alpha, t)$ à α fixé (figure 4.6 page ci-contre) n'est pas une loi de puissance de la forme $t^{-\delta}$ comme on l'attendrait pour la percolation dirigée.
- Enfin, les règles 18, 26, 106 et 146 ont sans doute le même seuil α_c dans les deux modèles (au moins jusqu'à la précision actuelle 3.10^{-4}).

Dans notre modèle à deux configurations, on peut déduire la coalescence rapide si l'on suppose que les deux configurations sont indépendantes à chaque étape².

Comme annoncé dans le paragraphe précédent, soit d' la densité de 1 dans le

²Cette hypothèse est clairement fautive pour les règles coalescentes non nilpotentes, et c'est la raison de l'étude de ce modèle, mais cette hypothèse peut néanmoins être une approximation intéressante pour les règles considérées ici.

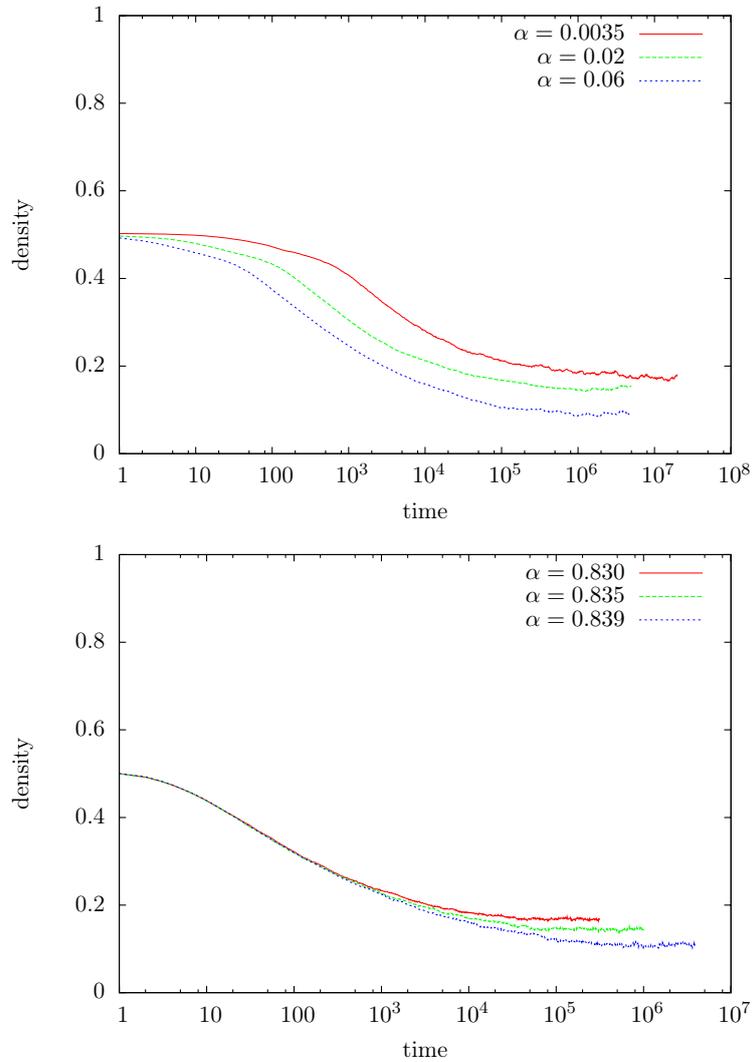


FIG. 4.6 – $\rho(\alpha, t)$ densité de cellules en désaccord en fonction du temps pour quelques valeurs de α . La règle 6 (en haut), qui n'est pas dans la classe de la percolation dirigée, est comparée à une règle typique de cette classe (la règle 58, en bas). Dans le cas de la règle 6, les courbes sont séparées dès les premières valeurs de t . (Il y a néanmoins une transition, la densité asymptotique étant 0 pour α suffisamment grand.)

modèle à une seule configuration. La densité de cellules en désaccord est alors

$$\begin{aligned} 2d'(1-d') &= 2C(\alpha - \alpha_c)^\beta(1 - C(\alpha - \alpha_c)^\beta) \\ &= 2C(\alpha - \alpha_c)^\beta - 2C^2(\alpha - \alpha_c)^{2\beta} \end{aligned}$$

Le second terme est négligeable (il est du deuxième ordre) pour α proche de α_c . Il suit (sous l'hypothèse d'indépendance) que la règle vérifie toujours $\rho_\infty(\alpha) \propto (\alpha - \alpha_c)^\beta$ et appartient donc à la classe de la percolation dirigée.

Notons cependant que cela est alors plus difficile à observer expérimentalement à cause de la plus grande précision nécessaire. Nous n'atteignons pas actuellement cette précision, mais nous observons néanmoins pour ces règles un exposant β plus faible et une plus grande sensibilité à l'intervalle d'échantillonnage proche de α_c .

Ces règles ne satisfont pas le point 4 page 22 de la conjecture de Grassberger (il y a en effet une autre transition de phase qui intervient dans l'automate sous-jacent), il serait intéressant dans de futurs travaux d'étudier si elles font néanmoins partie de la classe de la percolation dirigée.

4.3 Conclusion et perspectives

Nous avons introduit la notion d'automate cellulaire rapidement coalescent pour formaliser un comportement émergent des automates cellulaires asynchrones. Ce comportement est similaire à celui du chapitre 3 : coalescence ou non de deux configurations. Nous avons montré dans une section théorique l'existence d'automates cellulaires non rapidement coalescents et d'une infinité d'automates cellulaires rapidement coalescents. Puis nous avons proposé dans une section expérimentale une classification basée sur le comportement vis-à-vis de la coalescence, et identifié une transition de phase. Nous obtenons alors un nouvel ensemble de quelques modèles de percolation dirigée.

Percolation dirigée De même qu'au chapitre 3 et pour quelques autres modèles, la limite du régime sous-critique n'est pas un attracteur unique ni un ensemble de points fixes mais une configuration qui évolue.

Puisque la règle 6 n'est pas dans la classe d'universalité de la percolation dirigée, sa classification est une question naturelle.

Enfin, il est remarquable que certaines règles subissent deux transitions de phase, une pour α petit et une autre pour α grand.

Simulations Par ailleurs, ce comportement émergent montre que, lors de simulations numériques de systèmes complexes, un soin particulier doit être apporté au choix des cellules mises à jour à chaque étape, car ce choix peut modifier radicalement le comportement du système près du seuil ; par ailleurs, d'un côté du seuil, il efface toute influence de la configuration initiale.

Réciproquement, dans les conditions de coalescence rapide, ce comportement émergent indique que quelle que soit la configuration initiale (sauf pour un nombre négligeable) on atteint rapidement le régime permanent. De plus, simuler une évolution permet de prédire toutes les autres partageant la même histoire de mises à jour.

Systèmes dynamiques Enfin, on peut apporter quelques interprétations en termes de systèmes dynamiques. La coalescence est un exemple de système où deux instances convergent naturellement vers l'ensemble limite, mais de plus finissent par être sur le même point de cet ensemble en même temps.

Ici la source d'aléatoire pour la configuration initiale n'a pas d'influence : cette dynamique aléatoire avec un comportement irrégulier est insensible aux perturbations de la configuration initiale (sensible seulement à l'aléatoire de la dynamique) et n'entre donc pas dans la définition des systèmes chaotiques.

Perspectives Les prochains travaux pourraient porter sur des preuves formelles de coalescence rapide. Plus précisément, un schéma général applicable à plusieurs règles (au lieu de reposer sur les particularités d'une règle comme nous l'avons fait) aiderait sans doute à comprendre le phénomène. Une preuve de transition (coalescence rapide ou non en fonction de α) pourrait ici aussi être une contribution au vieux problème d'une valeur analytique pour β .

Notons enfin que la coalescence apparaît toujours soit pour tout couple de configurations initiales sauf un nombre négligeable, soit pour aucun couple. Il n'y a pas de règles où la coalescence apparaît pour une proportion constante de couples. Cependant, nous avons observé des cas où la coalescence rapide apparaissait une fois sur deux, selon les configurations initiales et l'histoire des mise à jour. Il est possible que pour des règles ayant plus d'états, un plus grand voisinage ou bien en dimension supérieure, le couple de configurations initiales détermine seul s'il y aura coalescence ou non.

Enfin, il nous semble important de généraliser la notion de coalescence à des configurations infinies.

Chapitre 5

Équivalence entre automates déterministes asynchrones et une classe d'automates probabilistes

Nous avons observé expérimentalement que certains automates cellulaires rapidement coalescents ont un comportement de percolation dirigée (chapitres 3 et 4). Il existe de nombreux exemples reconnus de percolation dirigée [Hinrichsen (2000)] et il est remarquable qu'autant de modèles aient les mêmes exposants critiques, comme vu à la section 2.2. Cependant, l'attribution de ces modèles à cette classe est le plus souvent expérimentale : les exposants d'un modèle sont mesurés sur des simulations numériques, et l'accord avec la valeur mesurée sur d'autres systèmes permet d'attribuer ce modèle à cette classe. Les liens entre modèles sont donc d'ordre phénoménologique : on observe expérimentalement les mêmes exposants dans les lois de puissance. Avoir de nombreux modèles que l'on n'a pas réduits les uns aux autres montre la richesse de cette classe et donne autant d'angles d'étude. Cependant, cet état de fait est peu satisfaisant pour l'informaticien habitué aux réductions formelles, telles que celles de NP complétude. Nous avons donc souhaité prouver de façon formelle que les modèles que nous avons étudiés sont également dans cette classe. Nous ne prouverons pas ici que les modèles de couplage appartiennent à la percolation dirigée, nous avons plutôt choisi de commencer par un modèle plus simple, sur lequel historiquement la percolation dirigée a d'abord été observée : les automates asynchrones.

Concernant les recherches d'équivalences entre modèles, [Domany and Kinzel (1984)] établissent une équivalence entre le modèle d'Ising pour un potentiel soigneusement choisi, un automate cellulaire probabiliste à deux paramètres (l'automate de Domany-Kinzel) et la percolation dirigée. Cette équivalence entre certains modèles de spins à $d+1$ dimensions et les automates cellulaires à d dimensions a été par la suite largement étudiée [Goldstein, Kuik, Lebowitz, and Maes (1989), Dai Pra, Louis, and Røelly (2002)].

Nous choisissons donc cet automate de Domany-Kinzel [Kinzel (1983)] comme élément de la classe de la percolation dirigée, pour sa popularité et sa simplicité. Nous allons démontrer un lien analytique entre ce modèle et les automates asynchrones. Nous allons plus précisément prouver qu'une classe d'automates cellulaires probabilistes est équivalente aux automates cellulaires déterministes asynchrones. Puisque cette classe d'automates probabilistes contient l'automate de Domany-Kinzel pour certains choix des paramètres, nous aurons ainsi prouvé que l'on rencontre la percolation dirigée parmi les automates cellulaires asynchrones.

En outre, un apport de ce travail est de définir une notion de simulation préservant la densité pour les automates probabilistes ou asynchrones. La simulation d'un automate par un autre a été largement explorée sur les automates synchrones déterministes en tant que modèles de calcul. Elle a notamment été formalisée par des notions de groupage [Rapaport (1998), Ollinger (2002), Theyssier (2005)], qui introduisent un ordre partiel sur les automates révélant une structure riche. Le groupage fait correspondre un nombre fixe d'étapes de l'automate simulant à un (autre) nombre fixe d'étapes de l'automate simulé. Mais en dynamique asynchrone, cette correspondance n'est plus possible car une cellule de l'automate simulant peut ne pas être mise à jour pendant un nombre d'étapes arbitrairement grand.

D'autres notions de simulation [Golze (1978), Gács (1997)] traitent ce cas. Elles ne conviennent cependant pas à notre but (transporter des résultats de transition de phase, donc préserver la densité d'états « actifs ») car elles traitent de calculs non-déterministes (donc de l'arbre des orbites possibles) sans attacher de probabilité à chaque transition. Nous avons donc été amené à proposer une nouvelle notion de simulation, inspirée de ces dernières mais avec plus de contraintes.

Ce travail exploratoire détaille la construction établissant l'équivalence, établit un lien entre transitions de phase et simulations entre automates, mais laisse de nombreuses pistes ouvertes. Précisément, nous ne montrons la conservation que de l'exposant β , étudier si les autres exposants critiques sont conservés par l'équivalence proposée ici sera nécessaire dans un prochain travail.

5.1 Définitions

Nous présentons dans cette section les automates probabilistes, la sous-classe que nous allons montrer équivalente aux automates asynchrones, et la notion de simulation que nous proposons.

Définition 14. Un automate cellulaire probabiliste est un vecteur (Q, d, V, δ) où

- Q, d et V sont définis comme pour un automate cellulaire :
 - Q est un ensemble fini, l'ensemble des *états*, que l'on identifie aux entiers $\llbracket 1, |Q| \rrbracket$;
 - $d \in \mathbb{N}^*$ est la *dimension* ;
 - $V = \{v_1, \dots, v_{|V|}\}$, le *voisinage*, est un ensemble fini de vecteurs de \mathbb{Z}^d ;
- $\delta : Q^{|V|} \rightarrow \left\{ p_1, \dots, p_{|Q|} \mid \sum_{i=1}^{|Q|} p_i = 1 \right\}$ est la *règle de transition* ;

À une configuration locale $(q_1, \dots, q_{|V|})$, la règle de transition δ associe donc un vecteur $(p_1, \dots, p_{|Q|})$: p_i est la probabilité de passer dans l'état i sachant que le voisinage est $(q_1, \dots, q_{|V|})$. Les orbites sont similaires à celles d'un automate déterministe, mais sont désormais des variables aléatoires : pour une configuration initiale $c(\cdot, 0)$, on tire au sort l'état de chaque cellule de la configuration $c(\cdot, 1)$, puis à partir de cette configuration on tire au sort la configuration $c(\cdot, 2)$, etc. Précisément, l'orbite est définie récursivement comme suit. Soit $t \in \mathbb{N}$, $x \in \mathcal{U}$ et $(p_1, \dots, p_{|Q|}) := \delta(c(x + v_1, t), \dots, c(x + v_{|V|}, t))$, alors $\mathbb{P}(c(x, t + 1) = i) = p_i$.

Remarque (stabilité par composition). Un automate déterministe est une fonction des configurations vers les configurations. Parmi toutes les fonctions $Q^{\mathcal{U}} \rightarrow Q^{\mathcal{U}}$, le classique théorème de Hedlund [Hedlund (1969)] caractérise les fonctions qui sont celles d'un automate cellulaire. Cette classe de fonctions est stable par composition, ce qui veut dire que

pour toutes règles de transition δ et δ' d'automates cellulaires, la règle consistant à appliquer δ (ce qui donne une configuration intermédiaire) puis δ' (sur cette configuration intermédiaire) est une règle de transition d'automate cellulaire.

Les automates probabilistes peuvent également être vus comme une classe de fonctions aléatoires de $Q^{\mathcal{U}}$ vers $Q^{\mathcal{U}}$, mais il est à remarquer que cette classe n'est plus stable par composition de fonction. Définir une classe d'automates probabilistes ayant cette stabilité serait un problème intéressant.

Nous considérons dans ce chapitre une sous-classe des automates probabilistes : une probabilité p est fixée et, à chaque pas de temps, chaque cellule n'a qu'une loi de Bernoulli $\mathcal{B}(p)$ comme source d'aléatoire. Cela correspond à la lecture d'un unique bit. La probabilité p est fixe et identique pour toutes les cellules. Formellement :

Définition 15 (Automate cellulaire probabiliste à un bit). Un automate cellulaire probabiliste (synchrone) à un bit est un vecteur (Q, d, V, δ, p) où

- Q, d et V sont définis comme pour un automate cellulaire,
- $p \in [0; 1]$
- $\delta : \{0; 1\} \times Q^{|V|} \rightarrow Q$ est la règle de transition.

En d'autres termes, δ est une règle déterministe, qui a comme entrées les états de la configuration locale plus un bit aléatoire. Ce bit est tiré selon une loi de Bernoulli de paramètre p . Cela permet par exemple de « simuler » la dynamique asynchrone : le bit est utilisé pour décider de simuler ou non la transition. Si α est le taux d'asynchronisme, on choisit donc $p := \alpha$.

Les orbites sont alors définies comme suit. Soit $c(\cdot, 0)$ une configuration initiale et $(X(x, t))_{x \in \mathcal{U}, t \in \mathbb{N}}$ une famille de variables aléatoires indépendantes identiquement distribuées, de loi $\mathcal{B}(p)$ (ce sera la source d'aléatoire). Alors

$$\forall t \in \mathbb{N} \quad \forall x \in \mathcal{U} \quad c(x, t+1) := \delta(X(x, t), c(x + v_1, t), \dots, c(x + v_{|V|}, t))$$

Fait 16. *Les automates déterministes sont des cas particuliers d'automates probabilistes à un bit (qui ignorent la source d'aléatoire), donc des cas particuliers d'automates probabilistes.*

La classe des automates probabilistes à un bit n'est pas si restrictive qu'elle peut sembler au premier abord. En effet, elle simule les automates probabilistes pour lesquels les p_i de la définition 14 sont tous rationnels.

Montrer cette simulation nous permet de donner un exemple de la notion de simulation que nous allons définir. Soit donc $B = (Q_B, d, V, \delta_B)$ un automate probabiliste dont les p_i sont rationnels, nous allons construire un automate probabiliste à un bit $A = (Q_A, d, V, \delta_A, p_A)$ le simulant.

On choisit $p_A := \frac{1}{2}$ pour la loi de Bernoulli qui est la source d'aléatoire de A . On pose $Q_A := Q_B \times Q$ où Q va servir à stocker des informations intermédiaires.

Considérons une configuration locale $(q_1, \dots, q_{|V|})$ de B et montrons comment A peut calculer le résultat de δ_B . Soit $(p_i)_{i \in [1, |Q|]} := \delta(q_1, \dots, q_{|V|})$ et d le plus petit dénominateur commun des p_i , on peut donc écrire $p_i = \frac{q_i}{d}$. Soit $k := \lceil \log_2 d \rceil$, supposons pour l'instant que d est une puissance de 2. Il suffit alors qu'une cellule fasse k transitions où elle stocke simplement le bit aléatoire. Lorsqu'elle a k bits aléatoire de loi $\mathcal{B}(\frac{1}{2})$, la cellule peut les utiliser pour tirer au sort un entier X de $[1, |Q|]$ de façon que $\mathbb{P}(X = i) = \frac{q_i}{d} = p_i$. Cet entier X est le nouvel état q . Ainsi, une cellule de A peut calculer la règle δ_B sans espace supplémentaire, mais avec du temps (des pas de calcul) en plus.

Nous souhaitons de plus dans notre notion de simulation que le temps pour simuler un pas de calcul soit indépendant de la configuration locale. Dans la construction ci-dessus ce n'est pas le cas car d , donc k , dépendent de la configuration locale. Il suffit pour corriger cela de définir d comme le plus petit dénominateur commun de tous les p_i de l'image de δ_B .

Si d n'est pas une puissance de 2, il suffit d'utiliser la méthode du rejet¹. Le nombre de transitions de A nécessaire à la simulation d'une transition de B est alors une variable aléatoire. Les cellules ne sont donc plus synchronisées. Ce phénomène (le fait que certaines cellules effectuent une transition avant leurs voisines) est inévitable avec des automates asynchrones, il faut donc l'accepter dans notre définition de simulation. Nous imposons cependant que le nombre de transitions simulées par une cellule (son *compteur*) peut être déterminé seulement à partir de la suite des états de cette cellule, et que si l'on considère les cellules chacune quand son compteur a une valeur commune a , on obtient la a^e configuration de B . Dans la présente simulation de B par A , on peut utiliser la construction des *marching soldiers* présentée à la section 5.3.2 pour garantir ce dernier point.

5.1.1 L'automate de Domany-Kinzel

Rappelons la définition de l'automate de Domany-Kinzel. C'est un automate cellulaire probabiliste synchrone, en une dimension. À chaque étape, une cellule regarde son état et celui de sa voisine de droite. Si les deux états sont blancs, elle reste blanche. Si l'un des deux est noir, elle devient noire avec probabilité p_1 et blanche avec probabilité $1 - p_1$. Si les deux sont noirs, elle devient noire avec probabilité p_2 . Cela est résumé sur la figure 5.1. Si l'on choisit $p_1 = p_2$ (cas dans lequel on observe encore la percolation dirigée), c'est un automate probabiliste à un bit.

$$\mathbb{P} \left(\begin{array}{c} \circ \\ \uparrow \swarrow \\ \circ \quad \circ \end{array} \right) := 1, \quad \mathbb{P} \left(\begin{array}{c} \bullet \\ \uparrow \swarrow \\ \bullet \quad \circ \end{array} \right) := \mathbb{P} \left(\begin{array}{c} \bullet \\ \uparrow \swarrow \\ \circ \quad \bullet \end{array} \right) := p_1, \quad \mathbb{P} \left(\begin{array}{c} \bullet \\ \uparrow \swarrow \\ \bullet \quad \bullet \end{array} \right) := p_2$$

FIG. 5.1 – Probabilités de transition de l'automate de Domany-Kinzel.

Dans toute la suite, nous nous restreignons à la dimension $d = 1$.

5.1.2 Simulation

Précisons maintenant la notion de simulation que nous allons utiliser. Cette notion est définie en toute généralité : il n'est pas supposé que l'un des deux automates est probabiliste ni que l'autre est asynchrone. Une première notion de simulation concerne les calculs déterministes : intuitivement, un automate A simule un automate B si tout calcul effectué à l'aide de B peut également être effectué à l'aide de A . Pour une configuration initiale de B , un fonction « simple » (appelons-la « encode ») donne une configuration initiale de A , une autre « décode » donne la traduction du résultat de A vers le résultat de B . Ainsi, pour simuler un calcul de B , on applique *encode*, on fait le calcul grâce à A , on applique *décode*, et l'on obtient le même résultat que si l'on avait fait le calcul sur B (nous n'aborderons pas ici les différentes conventions pour détecter que le calcul de

¹La méthode du rejet consiste dans notre cas à simuler une variable aléatoire X uniforme sur $\llbracket 1, d \rrbracket$ à l'aide d'une variable aléatoire Y uniforme sur $\llbracket 1, 2^k \rrbracket$. Si $Y \leq d$, on prend Y comme une réalisation de X . Sinon ($d < Y \leq 2^k$) on recommence avec un nouveau tirage de Y .

A est terminé.) Voir figure 5.2. Cette notion de simulation s'étend naturellement aux automates probabilistes, il suffit de considérer une distribution de probabilité sur les résultats au lieu d'un résultat déterministe.

Mais les pas de calcul intermédiaires ne sont pas nécessairement reliés : dans cette notion de simulation, seul le résultat compte. Nous utilisons ici une notion plus forte, qui vise à simuler l'orbite, c'est-à-dire l'ensemble des configurations rencontrées. Pour cette notion, un automate A simule un automate B si *chaque pas* du calcul de B peut être déduit du calcul de A . En d'autres termes, l'ensemble des orbites de B est l'image par une fonction « simple » de l'ensemble des orbites de A .

Nous imposons d'avoir une fonction « simple », sans quoi l'on pourrait cacher beaucoup de calculs dans cette fonction. Un exemple extrême : si la fonction connaît toute l'orbite, elle peut extraire n'importe quelle orbite de l'automate trivial « changer d'état au hasard ». Le critère est ici que la fonction soit :

- Locale dans l'espace. Pour connaître l'état d'une cellule de B , il n'y a qu'une cellule de A dont il faut connaître l'état. En particulier, il n'est pas nécessaire de connaître toute la configuration.

Dans cette définition, il y a donc bijection entre les cellules des deux automates. On pourrait au contraire faire correspondre plusieurs états de l'automate simulant à un état de l'automate simulé. Cela aboutit à la notion fructueuse de groupage évoquée en introduction. Le groupage permet notamment de simuler un automate par un autre ayant moins d'états. Mais étudier la généralisation du groupage aux automates asynchrones et probabilistes sort du cadre du présent travail.

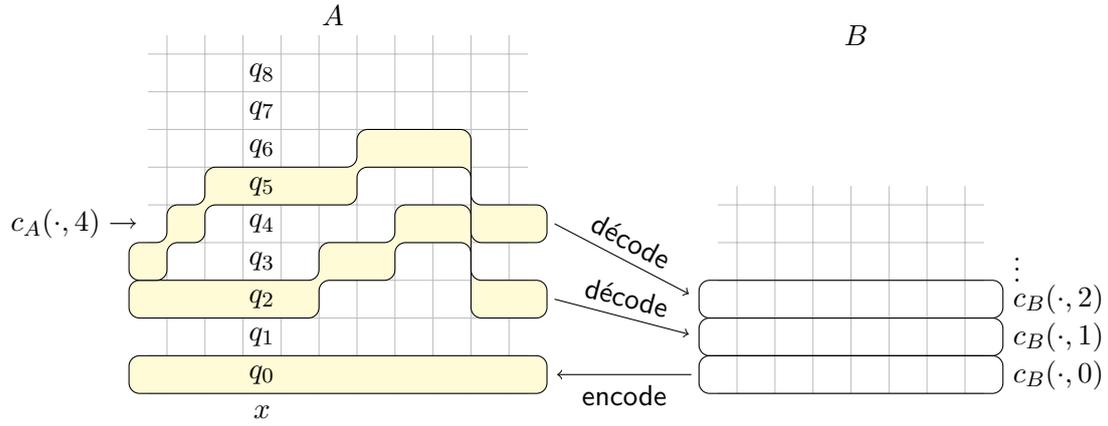
- Locale dans le temps. Ce point est un peu plus délicat car nous voulons autoriser l'automate A à faire un nombre variable (aléatoire et non borné) de pas de calcul pour simuler une transition de B . Nous disons qu'une transition de A est une **transition-étape** si elle correspond à une transition de B par opposition à une suite de transitions de A intermédiaires, qui ne servent qu'à calculer la prochaine transition de B . En d'autres termes, une transition-étape est une transition de q à q' telle que $\text{décode}(q)$ et $\text{décode}(q')$ soient les états d'une cellule de B dans deux configurations successives.

Ici non plus, nous ne pouvons pas autoriser une fonction arbitraire pour déterminer les transitions-étapes. C'est donc l'automate A qui doit « dire » lui-même quand une transition est une transition-étape. C'est le rôle de la fonction $\hat{\text{âge}} : Q_A \rightarrow \mathbb{N}$, (Q_A est l'ensemble des états de A). Pour une cellule dans l'état q , $\hat{\text{âge}}(q)$ devrait être le nombre de transitions-étapes effectuées par la cellule. On peut donc savoir qu'un pas de calcul de B a été effectué lorsque deux états successifs de A n'ont pas la même image par $\hat{\text{âge}}$. Q_A étant fini, $\hat{\text{âge}}(q)$ est plutôt le nombre de transitions-étapes modulo un petit entier. La fonction $\hat{\text{âge}}$ permet ainsi de déterminer localement l'écoulement simulé du temps de B .

Il est important de noter que la fonction $\hat{\text{âge}}$ est entièrement définie par l'état, deux cellules dans le même état ont donc le même âge. Mais rappelons que c'est le nombre de transitions-étapes modulo un entier.

Définition 17 (Simulation). Soient $A = (Q_A, 1, V_A, \delta_A)$ et $B = (Q_B, 1, V_B, \delta_B)$ deux automates cellulaires probabilistes en dynamique partiellement asynchrone (éventuellement avec $\alpha = 1$, c'est-à-dire en fait dans le cas synchrone). Nous disons que A simule B s'il existe trois fonctions

- $\text{encode} : Q_B \rightarrow Q_A$
- $\text{décode} : Q_A \rightarrow Q_B$

FIG. 5.2 – L'automate A simule l'automate B .

Les états q_0 à q_8 sont les états successifs de la cellule x . L'âge de q_0 et q_1 est 0, l'âge de q_2 , q_3 et q_4 est 1, l'âge de q_5 , q_6 , q_7 et q_8 est 2. En particulier, les transitions-étapes de cette cellule x sont les transitions de q_1 à q_2 et de q_4 à q_5 : $\text{étape}(q_1, q_2) = \text{étape}(q_4, q_5) = 1$, tandis que étape vaut 0 sur les autres (q_i, q_{i+1}) .

On impose que les q_i successifs de même âge (par exemple q_2 , q_3 et q_4) aient la même image par la fonction décode .

Noter que la configuration $c_A(\cdot, 4)$ ne correspond à aucune configuration de B : elle contient des états qui seront décodés pour constituer $c_B(\cdot, 1)$, d'autres qui seront décodés pour constituer $c_B(\cdot, 2)$, et encore quelques autres.

– âge : $Q_A \rightarrow \mathbb{N}$

telles que, pour toute configuration initiale $c_B(\cdot, 0)$ d'états de B , en posant

- $c_B(\cdot, \cdot)$ l'orbite de B ayant $c_B(\cdot, 0)$ comme configuration initiale ; c'est une variable aléatoire car B est probabiliste et asynchrone ;
- $c_A(\cdot, 0) := \text{encode}(c_B(\cdot, 0))$ une configuration initiale de A , c'est-à-dire que pour tout x , $c_A(x, 0) := \text{encode}(c_B(x, 0))$;
- $c_A(\cdot, \cdot)$ l'orbite de A ayant $c_A(\cdot, 0)$ comme configuration initiale (c'est aussi une variable aléatoire) ;
- $\text{étape}(q, q') := \begin{cases} 1 & \text{si } \hat{\text{âge}}(q) \neq \hat{\text{âge}}(q') \\ 0 & \text{sinon} \end{cases}$

Intuitivement, la transition de q à q' est une transition-étape si et seulement si $\text{étape}(q, q') = 1$.

– le compteur d'une cellule de A comme le nombre de transitions qu'elle a simulées :

$$\text{compteur}(x, t) := \sum_{i=1}^t \text{étape}(c_A(x, i-1), c_A(x, i))$$

– $\text{compteur}^{-1}(x, t) := \text{argmin}\{u \mid \text{compteur}(x, u) = t\}$, c'est le premier pas de calcul où la cellule x a effectué t transitions-étapes (i.e. a simulé t transitions de B).

– $c'_B : (x, t) \mapsto c_A(x, \text{compteur}^{-1}(x, t))$

alors

1. il existe une fonction $f : \mathbb{N} \rightarrow \mathbb{R}$ telle que

$$\begin{cases} f(m) \underset{m \rightarrow \infty}{=} o(m) \\ \forall t_2 \quad \forall \varepsilon > 0 \quad \exists t_3 \quad \forall m \quad \forall t \geq \max(t_3, f(m)) \\ \mathbb{P}(\forall x \in \llbracket 1, m \rrbracket \quad \text{compteur}(x, t) \geq t_2) \geq 1 - \varepsilon \end{cases}$$

ainsi pour tous (x, t) , $\text{compteur}^{-1}(x, t)$ est presque sûrement fini et $c'_B(\cdot, \cdot)$ est bien définie ;

2. $c_B(\cdot, \cdot)$ et $c'_B(\cdot, \cdot)$ ont la même distribution de probabilité ;
3. $\forall q, q' \in Q_a \quad \hat{\text{age}}(q) = \hat{\text{age}}(q') \Rightarrow \text{décode}(q) = \text{décode}(q')$;
4. les variables aléatoires $c'_B(\cdot, \cdot)$ et $\text{compteur}(\cdot, \cdot)$ sont indépendantes.

La condition principale est la **2** : on a reproduit les orbites de B à l'aide de A .

La condition **1** est utilisée dans la preuve du théorème **18**, elle joue un rôle essentiel dans le découpage de la configuration en blocs de taille m dont les états sont indépendants. Notre construction de simulation utilisera $f(n) := \ln n$.

La condition **3**, qui peut sembler arbitraire, indique que si A n'a pas simulé de transition alors l'état simulé n'a pas changé. Sans cette condition, cette notion de simulation ne serait pas transitive.

Sans la condition **4**, cette notion de simulation ne préserverait pas la proportion asymptotique d'états actifs (nous n'aurions alors pas atteint notre but, car une relation de simulation ne permettrait pas de transporter des résultats sur les transitions de phase.) En effet, la simulation pourrait par exemple privilégier les états actifs en accélérant les transitions qui mènent à un état actif, et en retardant celles qui suppriment un état actif, augmentant ainsi leur densité.

Notations. On abrège « presque sûrement » en « p.s. ». Pour un automate A , Q un sous-ensemble des états de A et c une configuration de A , notons $\rho(c, Q)$ la densité d'états de Q si elle existe :

$$\rho(c, Q) := \lim_{N \rightarrow \infty} \frac{1}{2N + 1} |\{x \mid c(x) \in Q \text{ et } x \in \llbracket -N, N \rrbracket\}|$$

Voici le théorème central de ce chapitre : sur les configurations infinies, la relation de simulation préserve la densité asymptotique de tout sous-ensemble d'états.

Théorème 18. *Soient A et B des automates cellulaires tels que A simule B , Q un sous-ensemble des états de B . Pour toute configuration initiale $c_B(\cdot, 0)$ de B , si les densités $(\rho(c_B(\cdot, t), Q))_{t \in \mathbb{N}}$ et la densité asymptotique $\rho_\infty := \lim_{t \rightarrow \infty} \rho(c_B(\cdot, t), Q)$ existent p.s., alors en posant $c_A(\cdot, 0) := \text{encode}(c_B(\cdot, 0))$, la limite $\lim_{t \rightarrow \infty} \rho(\text{décode}(c_A(\cdot, t)), Q)$ existe p.s. et est égale à ρ_∞ .*

Il est nécessaire d'appliquer la fonction `décode` car on calcule la densité des états appartenant à Q , mais il est intéressant de noter que la densité se calcule sur une configuration de A à t fixé, donc sur des cellules ayant potentiellement des compteurs différents (c'est-à-dire simulant des cellules de B n'ayant pas toutes fait le même nombre de pas de calcul).

Corollaire 19. *La relation de simulation préserve l'exposant critique β d'une transition de phase.*

Démonstration du théorème 18. Pour tout sous-ensemble fini de cellules $U \in \mathcal{U}$, notons $\rho_A(U, t)$ la densité des états de Q dans la configuration $c_A(\cdot, t)$ restreinte à U :

$$\rho_A(U, t) := \frac{1}{|U|} |\{x \in U \mid \text{décode}(c_A(x, t)) \in Q\}|$$

Et de même $\rho_B(U, t) := \frac{1}{|U|} |\{x \in U \mid c_B(x, t) \in Q\}|$.

Admettons pour l'instant que pour $T \in \mathbb{N}$, $\lim_{N \rightarrow \infty} \rho_A(\llbracket -N, N \rrbracket, T)$ existe p.s.,² c'est-à-dire que $\rho(\text{décode}(c_A(\cdot, T)), Q)$ existe. Soit $\varepsilon > 0$, nous allons montrer que presque sûrement

$$\exists t_1 \quad \forall T \geq t_1 \quad |\rho(\text{décode}(c_A(\cdot, T)), Q) - \rho_\infty| \leq 11\varepsilon$$

ce qui établira le résultat.

Soit t_2 tel que

$$\forall t \geq t_2 \quad |\rho(c_B(\cdot, t), Q) - \rho_\infty| \leq \varepsilon \quad (5.1)$$

D'après le point 1 de la définition de simulation, il existe t_3 tel que, pour tout $t \geq t_3$, un bloc de m cellules de l'automate A a simulé au moins t_2 transitions de B avec une probabilité $1 - \varepsilon$:

$$\forall m \quad \forall t \geq \max(t_3, f(m)) \quad \mathbb{P}\left(\forall x \in \llbracket 1, m \rrbracket \quad \text{compteur}(x, t) \geq t_2\right) \geq 1 - \varepsilon \quad (5.2)$$

Soit f la fonction du même point 1. Comme $f(m) \underset{m \rightarrow \infty}{=} o(m)$, il existe t_4 tel que $\forall t \geq t_4 \quad \exists m \in \mathbb{N} \quad t \geq f(m)$ et $\frac{2t}{m+2t} \leq \varepsilon$.

On pose alors $t_1 := \max(t_3, t_4)$ (remarquer que, nécessairement, $t_3 \geq t_2$). Soit $T \geq t_1$, il nous faut donc montrer que $|\rho(\text{décode}(c_A(\cdot, T)), Q) - \rho_\infty| \leq 11\varepsilon$.

D'après l'équation 5.1, la densité de toutes les configurations $c_B(\cdot, t)$ pour $t \in \llbracket t_2, T \rrbracket$ est égale à ρ_∞ à ε près, donc il existe N_1 tel que

$$\forall N \geq N_1 \quad \forall t \in \llbracket t_2, T \rrbracket \quad |\rho_B(\llbracket -N, N \rrbracket, t) - \rho_\infty| \leq 2\varepsilon \quad (5.3)$$

Soit m tel que $T \geq f(m)$ et $\frac{2T}{2T+m} \leq \varepsilon$. Découpons à présent l'ensemble \mathcal{U} des cellules en blocs de m cellules, deux blocs étant séparés par $2T$ cellules :

$$\dots \xrightarrow{m} \xleftrightarrow{2T} \xrightarrow{m} \xleftrightarrow{2T} \xrightarrow{m} \xleftrightarrow{2T} \xrightarrow{m} \xleftrightarrow{2T} \xrightarrow{m} \dots$$

Soit V l'ensemble des cellules appartenant à l'un des blocs. Le premier bloc de m cellules est centré sur la cellule 0, ainsi

$$\forall t \quad \forall N \quad |\rho_A(V \cap \llbracket -N, N \rrbracket, t) - \rho_A(\llbracket -N, N \rrbracket, t)| \leq 2\varepsilon \quad (5.4)$$

Et de même en remplaçant ρ_A par ρ_B .

Soit W l'union des blocs dont toutes les cellules ont simulé au moins t_2 transitions de B (i.e. ont des compteurs valant au moins t_2). W est une variable aléatoire. D'après l'équation 5.2, la probabilité qu'un bloc ne soit pas inclus dans W est majorée par ε . De plus les blocs sont séparés de $2T$ cellules, donc au temps T les états de deux blocs sont entièrement définis à partir de variables aléatoires indépendantes. Les probabilités précédentes sont donc indépendantes. Il s'ensuit qu'il existe p.s. N_2 tel que

$$\forall N \geq N_2 \quad |\rho_A(V \cap \llbracket -N, N \rrbracket, T) - \rho_A(W \cap \llbracket -N, N \rrbracket, T)| \leq 2\varepsilon \quad (5.5)$$

Et de même :

$$\forall t \in \llbracket t_2, T \rrbracket \quad \forall N \geq N_2 \quad |\rho_B(V \cap \llbracket -N, N \rrbracket, t) - \rho_B(W \cap \llbracket -N, N \rrbracket, t)| \leq 2\varepsilon \quad (5.6)$$

En rassemblant les équations 5.4 et 5.5, on obtient

$$\forall N \geq N_2 \quad |\rho_A(W \cap \llbracket -N, N \rrbracket, T) - \rho_A(\llbracket -N, N \rrbracket, T)| \leq 4\varepsilon \quad (5.7)$$

Et de même en remplaçant ρ_A par ρ_B et T par $t \in \llbracket t_2, T \rrbracket$ grâce aux équations 5.4 et 5.6.

Le point 4 de la définition de simulation indique que les variables aléatoires $c'_B(\cdot, \cdot)$ et $\text{compteur}(\cdot, \cdot)$ sont indépendantes. Les états de la configuration $\text{décode}(c_A(W, T))$ sont

²Ce point sera démontré à la fin de la preuve. Comme les schémas de preuve de ce point et du résultat principal (la densité est conservée) sont identiques, il nous a semblé plus intéressant de faire la preuve la plus détaillée sur le résultat principal.

donc tirés au sort parmi les états des configurations $c_B(W, t_2)$ à $c_B(W, T)$, *indépendamment des états de ces dernières configurations*. Les densités sont donc « proches ». C'est-à-dire que, en notant $P_N := \{\rho_B(W \cap \llbracket -N, N \rrbracket, t) \mid t \in \llbracket t_2, T \rrbracket\}$, il existe donc p.s. N_3 tel que

$$\forall N \geq N_3 \quad \rho_A(W \cap \llbracket -N, N \rrbracket, T) \in [\min P_N - \varepsilon; \max P_N + \varepsilon] \quad (5.8)$$

Soit enfin $N_0 := \max(N_1, N_2, N_3)$. D'après (5.7) pour ρ_B et (5.3), pour tout $N \geq N_0$, $P_N \subseteq [\rho_\infty - 6\varepsilon; \rho_\infty + 6\varepsilon]$. En rassemblant avec (5.8), on obtient que

$$\forall N \geq N_0 \quad \rho_A(W \cap \llbracket -N, N \rrbracket, T) \in [\rho_\infty - 7\varepsilon; \rho_\infty + 7\varepsilon]$$

Puis grâce à (5.7), $\forall N \geq N_0 \quad |\rho_A(\llbracket -N, N \rrbracket, T) - \rho_\infty| \leq 11\varepsilon$. En particulier,

$$\rho(\text{décode}(c_A(\cdot, T)), Q) = \lim_{N \rightarrow \infty} \rho_A(\llbracket -N, N \rrbracket, T) \in [\rho_\infty - 11\varepsilon; \rho_\infty + 11\varepsilon]$$

Montrons maintenant comme admis au début de la preuve que, pour $T \in \mathbb{N}$, la limite $\lim_{N \rightarrow \infty} \rho_A(\llbracket -N, N \rrbracket, T)$ existe p.s. Nous utilisons pour cela le théorème suivant, dû à Kolmogorov (voir théorème 2 p. 364 de [Shiryayev (1984)]) :

Si $(X_n)_{n \geq 1}$ est une suite de variables aléatoires indépendantes, de moyennes respectives $(m_n)_{n \geq 1}$ et de variances $(V_n)_{n \geq 1}$ telles que $\sum_{n \geq 1} \frac{V_n}{n^2} < \infty$, alors presque sûrement

$$\frac{1}{n} \left(\sum_1^n X_i - \sum_1^n m_i \right) \xrightarrow{n \rightarrow \infty} 0$$

Il suffit de montrer que la suite des $\rho_A(\llbracket -N, N \rrbracket, T)$ forme p.s. une suite de Cauchy, c'est-à-dire que pour tout $\eta > 0$,

$$\exists N_0 \quad \forall N, M \geq N_0 \quad |\rho_A(\llbracket -N, N \rrbracket, T) - \rho_A(\llbracket -M, M \rrbracket, T)| \leq \eta$$

La preuve utilise le même schéma que pour établir $\rho(\text{décode}(c_A(\cdot, T)), Q) = \rho_\infty$. Soit m tel que $\frac{T}{T+m} \leq \eta$, découpons à nouveau l'ensemble des cellules en blocs $(B_i)_{i \in \mathbb{Z}^*}$ de taille m séparés par $2T$ cellules. Soit V l'union des blocs. Les états de la configuration $\text{décode}(c_A(V, T))$ sont tirés au sort parmi les états des configurations $c_B(V, 0)$ à $c_B(V, T)$, indépendamment des états de ces configurations. La variable aléatoire indiquant quels états sont tirés au sort a donc la même loi sur chaque bloc. De plus, les réalisations sont indépendantes sur chaque bloc car ils sont séparés de $2T$ cellules.

On pose $X_i := \rho_A(B_i, T) + \rho_A(B_{-i}, T)$ la densité de c_A sur les blocs i et $-i$. Puisque les blocs couvrent la configuration c_A sur $\llbracket -N, N \rrbracket$ sauf une proportion η (les espaces entre les blocs), les X_i approximent $\rho_A(\llbracket -N, N \rrbracket, T)$ à η près. D'après le théorème ci-dessus, on peut approximer $\sum_{i=1}^N X_i$ à η près par $\sum_{i=1}^N m_i$ pour N assez grand.³

Or, en notant p_t l'espérance de la proportion d'états de $\text{décode}(c_A(V, T))$ tirés au sort dans $c_B(V, t)$, on a $\sum_{i=1}^N m_i = \sum_{i=1}^N \sum_{t=0}^T p_t \rho_B(B_i \cup B_{-i}, t)$. Cette densité de c_B sur les blocs est, à η près, la densité sur $\llbracket -N, N \rrbracket$. Enfin, grâce à la convergence pour tout t de $\rho_B(\llbracket -N, N \rrbracket, t)$, la somme $\sum_{t=0}^T p_t \rho_B(\llbracket -N, N \rrbracket, t)$ converge quand $N \rightarrow \infty$, c'est donc une suite de Cauchy en fonction de N .

En rassemblant alors ces inégalités par une suite d'inégalités triangulaires, on obtient bien la première inégalité annoncée : $|\rho_A(\llbracket -N, N \rrbracket, T) - \rho_A(\llbracket -M, M \rrbracket, T)| \leq 7\eta$. ■

5.2 Simulation d'un automate déterministe asynchrone par un automate probabiliste synchrone

Ce sens-là est facile : il suffit que l'automate probabiliste tire au sort s'il fait la transition ou bien s'il gèle l'état de la cellule. Pour qu'il puisse dire s'il a simulé une

³Noter que $\sum_{i=1}^N m_i$ ne converge pas nécessairement quand $N \rightarrow \infty$, d'où la nécessité d'utiliser une suite de Cauchy.

transition ou non, on ajoute une composante sur $\{0, 1\}$ à l'ensemble des états.

Soit donc $B = (Q_B, 1, V_B, \delta_B)$ un automate asynchrone à simuler, on définit l'automate probabiliste à un bit $A = (Q_B \times \{0, 1\}, 1, V_B, \delta_A)$ comme suit. On choisit p égal au taux de synchronisme de B pour la loi de Bernoulli. Pour toute configuration locale $q_1, \dots, q_{|Q|}$ dont l'état central s'écrit (q, ε) , on pose

$$\delta_A(b, q_1, \dots, q_{|Q|}) := \begin{cases} (\delta_B(q_1, \dots, q_{|Q|}), 1 - \varepsilon) & \text{si } b = 1 \\ (q, 1 - \varepsilon) & \text{si } b = 0 \end{cases}$$

On pose enfin encode et décode égales à l'identité et $\hat{\text{age}}(q, \varepsilon) := \varepsilon$.

Remarque. Pour éviter d'avoir à ajouter une composante sur $\{0, 1\}$ à l'ensemble des états, on pourrait dans la définition de simulation ajouter comme entrée à la fonction étape le fait que l'histoire des mises à jour ait ou non déclenché une mise à jour à ce pas de temps.

5.3 Simulation d'un automate probabiliste synchrone par un automate déterministe asynchrone

Ce sens-là est plus difficile. Nous proposons tout d'abord (section 5.3.1) une construction permettant à un automate déterministe asynchrone de lire un bit aléatoire dans l'histoire (aléatoire) des mises à jour. L'idée est simplement de mesurer si deux cellules voisines ont été mises à jour en même temps ou bien l'une après l'autre. Nous rappelons ensuite (section 5.3.2) une construction classique permettant de simuler un calcul synchrone dans une dynamique asynchrone. Il reste alors à rassembler le tout (section 5.3.3).

5.3.1 Mesure d'un bit aléatoire

Le principe est de s'aider du voisin de droite. Le bit aléatoire est 0 si les deux cellules ont été mises à jour en même temps, 1 si l'une a été mise à jour avant l'autre. Si α est la probabilité de mise à jour, on obtient donc une loi de Bernoulli de paramètre α^2 . (C'est donc l'automate à simuler qui imposera le choix de α .)

Plus précisément, nous définissons cinq états avec la signification suivante⁴ :

- 0 Attente.
- 1 Je vais mesurer un bit aléatoire.
- 2 J'ai été mis à jour en venant de l'état 1, avant ou en même temps que mon voisin de droite. Je vais lire le bit aléatoire : est-ce que mon voisin de droite a été mis à jour ?
- 3 Prêt à aider mon voisin gauche à faire une mesure.
- 4 J'ai été mis à jour en venant de 3, avant ou en même temps que mon voisin de gauche.

La figure 5.3 page suivante présente la suite de transitions lors de la mesure d'un bit aléatoire. On commence dans la configuration d'attente $b00$, qui mène à $b10$. L'état 1 demande l'aide du voisin de droite pour effectuer une mesure. Lorsque le voisin de droite est prêt à participer (c'est-à-dire qu'il est dans l'état 3), la mesure est effectuée : selon que la cellule centrale, son voisin de droite, ou les deux sont mis à jour, on passe dans la configuration $b23$, $b14$ ou $b24$ respectivement. Lorsque la cellule centrale est mise à jour, elle lit le résultat, indiqué sur la figure par « < », « > » ou « = » respectivement.

⁴Le style à la première personne, qui personnifie les cellules, peut sembler légèrement relâché ; cependant il allège notablement le discours. « Je » désigne ici la cellule qui est mise à jour.

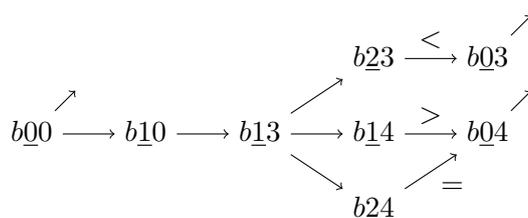


FIG. 5.3 – Graphe des principales configurations locales de l'automate lisant les bits aléatoires. « b » signifie « un état de l'ensemble $\{0, 3, 4\}$ » c'est-à-dire « aidant son voisin ». Depuis la configuration $b13$, selon que la cellule centrale se met à jour avant la cellule de droite, ou le contraire, ou encore que les deux se mettent à jour en même temps, on obtient la configuration $b23$, $b14$ ou $b24$ respectivement. De même, depuis la configuration $b00$, on peut aboutir à plusieurs configurations, qui n'ont pas toutes été représentées : la flèche « \nearrow » indique que, pour la configuration d'où elle part, les transitions n'ont pas toutes été représentées. Les boucles sur un sommet ne sont pas dessinées. Les transitions étiquetées « $<$ », « $>$ » ou « $=$ » produisent un bit aléatoire.

Table de transition

Nous pouvons maintenant spécifier la table de transition complète, c'est la table 5.1 page suivante.

Voici une interprétation de l'utilité des transitions. Les lettres font référence à celles de la table 5.1.

- a Je vais mesurer un bit aléatoire (mon voisin n'est pas occupé).
- b Je ne peux pas mesurer de bit car mon voisin de droite ne peut pas m'aider, donc je reste dans l'état 0.
- c Le voisin gauche souhaite faire une mesure, je suis prêt à l'aider.
- d, e Deux cellules voisines souhaitent faire une mesure en même temps, ce qui est impossible car celle de gauche a besoin de celle de droite pour faire la mesure. Le conflit sera résolu aléatoirement selon la première cellule mise à jour (c'est donc une brisure de symétrie).
- f J'attends que le voisin de droite soit prêt à m'aider.
- g Je note que je suis mis à jour.
- h Je lis un bit aléatoire : le voisin a été mis à jour avant moi. Cela est noté « $<$ ».
- i Je lis un bit aléatoire : j'ai été mis à jour avant (noté « $<<$ »).
- j Je lis un bit aléatoire : les deux cellules ont été mises à jour en même temps (noté « $=$ »).
- k Je note que je suis mis à jour, pour le dire à mon voisin en train d'effectuer une mesure.
- l J'attends que mon voisin lise qu'il a été mis à jour avant moi.
- m Puisqu'il n'est plus dans l'état 1 ou 2, mon voisin a lu que j'avais été mis à jour avant lui. Donc retour à l'état d'attente 0.
- n J'attends que mon voisin lise le bit aléatoire (le résultat de la mesure).
- o Mon voisin a lu le bit, retour à l'état 0.

(étiquette)

Légende :

voisinage
résultat de la transition

Transitions depuis l'état 0

(a)	(b)	(c)
{0,3,4} 0 0	{0,3,4} 0 {1,2,3,4}	1 0 _
1	0	3

(La configuration « 20_ » n'a pas d'antécédent, cf. lemme 21 page suivante.)

Transitions depuis l'état 1

(d)	(e)	(f)	(g)	(h)
1 1 _	_ 1 1	{0,2,3,4} 1 0	{0,2,3,4} 1 3	{0,2,3,4} 1 4
0	0	1	2	0,>

(« _12 » n'a pas d'antécédent)

Transitions depuis l'état 2

(i)	(j)	(« _20 », « _21 » et « _22 » n'ont pas d'antécédent)
_ 2 3	_ 2 4	
0,<	0,=	

Transitions depuis l'état 3

(k)	(l)	(m)
1 3 _	2 3 _	{0,3,4} 3 _
4	3	0

Transitions depuis l'état 4

(n)	(o)
{1,2} 4 _	{0,3,4} 4 _
4	0

TAB. 5.1 – Table de transition de l'automate lisant les bits aléatoires. « _ » signifie « n'importe quel état ».

Définition 20. Nous disons qu'une table est *complète* pour une configuration initiale c_0 si, quelle que soit la réalisation de l'aléa, toutes les configurations locales du diagramme espace-temps apparaissent dans la table.

Cela signifie que la table spécifie complètement l'orbite de l'automate : on ne rencontrera jamais de configuration locale pour laquelle la table ne dit pas quel est l'état suivant. Les tables sont généralement complètes car on spécifie toutes les transitions. Au contraire, la table 5.1 page précédente ne contient pas toutes les configurations locales, ce qui permet de mieux comprendre le fonctionnement de l'automate. En contrepartie, il faut vérifier que la table est complète.

Lemme 21. *La table 5.1 page précédente est complète pour la configuration initiale 0^* .*

Démonstration. Il suffit de vérifier que les configurations locales $20_ , _12, _20, _21$ et $_22$ n'ont pas d'antécédent, car toutes les autres sont dans la table.

C'est le cas pour $20_ :$ supposons par l'absurde qu'il existe une suite s de transitions menant à cette configuration. Le seul état menant à l'état 2 est l'état 1, considérons donc la dernière transition de s où la cellule de gauche passe de l'état 1 à l'état 2. Lors de cette transition, la cellule centrale est, d'après la table, nécessairement dans l'état 3. Mais cette cellule restera dans l'état 3 tant que sa voisine de gauche sera dans l'état 2 ; d'où la contradiction.

Le raisonnement est identique pour les configurations locales $_20, _21$ et $_22$. Pour $_12$, considérons de même une suite s de transitions menant à cette configuration. Le seul état menant à l'état 1 est l'état 0, considérons donc la dernière transition de s où la cellule centrale passe de l'état 0 à l'état 1. Lors de cette transition la cellule de droite est dans l'état 0. Les seuls changements d'état possibles pour la cellule de droite sont alors de passer dans l'état 3 puis 4. Contradiction. ■

Il nous faut maintenant vérifier que :

- Chaque fois que cette règle produit un bit aléatoire, il correspond bien à ce que l'on veut mesurer (mise à jour d'une cellule avant sa voisine ou bien des deux en même temps). Cela est fait au lemme 22.
- Pour toute cellule et à tout pas de calcul, l'espérance du temps pour qu'elle mesure un bit aléatoire est finie. Le lemme 23 garantit cela.

Lemme 22. *Toute suite de transitions menant à la mesure d'un bit aléatoire passe par une configuration locale de la forme $b13$, où $b \in \{0, 3, 4\}$.*

Démonstration. Après la production d'un bit, la cellule centrale n'est pas dans l'état 1. Pour produire un nouveau bit, elle doit passer par l'état 1 ; au moment où elle entre dans cet état, la configuration locale est de la forme $b00$. Tant que la cellule de droite n'est pas passée par l'état 3 (donc pas par l'état 4 non plus), la cellule de gauche reste dans l'un des états de b et la cellule centrale reste dans l'état 1 (ou bien revient à l'état 0 et l'on est ramené au problème précédent). On rencontre donc la configuration locale $b13$ avant de produire un nouveau bit. ■

Lemme 23. *Quelle que soit la configuration locale (accessible depuis la configuration globale 0^*), l'espérance du nombre de pas de calcul avant de revenir dans une configuration de la forme $b00$ est finie.*

Démonstration. Rappelons que, si l'on part de la configuration 0^* , certains motifs sont impossibles, comme vu au lemme 21 page précédente. Traçons (figure 5.4) le graphe ayant toutes les configurations locales possibles comme sommets et les transitions entre deux configurations comme arêtes. La cellule centrale est soulignée sur chaque sommet.

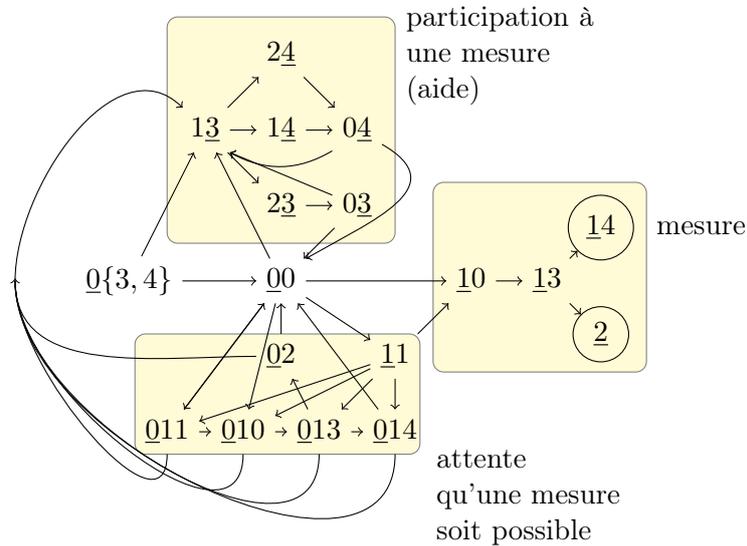


FIG. 5.4 – Transitions possibles de la règle lisant les bits aléatoires. Seules les transitions sortant des configurations entourées (qui ramènent vers l'état 0) n'ont pas été représentées.

On obtient une chaîne de Markov telle que, quel que soit l'état de départ, l'espérance du nombre de pas de calcul pour atteindre l'un des deux états entourés est finie. Depuis l'un de ces états, la règle produit un bit dès que la cellule centrale est mise à jour. ■

Résumons ce que nous venons de construire : il existe un automate asynchrone déterministe dont chaque cellule produit un nombre infini de bits aléatoires indépendants. Formellement :

Théorème 24. *Il existe un automate cellulaire $(Q, 1, V, \delta)$, une fonction $\text{read} : Q^2 \rightarrow \{0, 1, \perp\}$ et une configuration initiale $c(\cdot, 0)$ tels que, dans la dynamique partiellement asynchrone avec probabilité α de mise à jour, si l'on note $B(x, i)$ le i^{e} élément différent de \perp dans la suite $(\text{read}(c(x, t), c(x, t + 1)))_{t \in \mathbb{N}}$, alors ce i^{e} élément existe p.s., et les variables aléatoires $(B(x, i))_{x \in \mathcal{U}, i \in \mathbb{N}}$ sont indépendantes et identiquement distribuées, de loi de Bernoulli $\mathcal{B}(\alpha^2)$.*

5.3.2 Synchronisation

Nous exposons dans cette section la construction des *marching soldiers*, qui a été introduite par [Nakamura (1974)] et est exposée par exemple dans [Gács (1997)].

Cette construction est une méthode pour simuler un automate synchrone à q états par un automate asynchrone à $3q^2$ états. L'idée est de synchroniser à un pas de calcul près les cellules voisines. On ajoute un champ à l'état de chaque cellule pour savoir si

cette cellule est en avance, en retard, ou synchronisée par rapport à ses voisines. Cela permet de garantir qu'aucune cellule n'aura plus d'un pas d'avance sur ses voisines.

Ensuite, chaque cellule stocke son état, ainsi que son état au pas précédent dans le calcul qui serait effectué par un automate synchrone. Les cellules en retard utilisent l'état précédent stocké par leurs voisines pour effectuer leur transition. Les cellules synchronisées utilisent l'état courant. Les cellules en avance ne font pas de transition et attendent leurs voisines.

En d'autres termes, l'ensemble des états est un produit cartésien entre des informations sur l'état de l'automate synchrone et une information de phase. L'évolution de la composante de phase est indépendante de celle de l'état de l'automate d'origine, on peut ainsi étudier le mécanisme de synchronisation séparément de l'automate d'origine.

L'information de phase est un élément de l'ensemble $\{0, 1, 2\}$; elle est incrémentée modulo 3 à chaque transition (pourvu que la cellule ne soit pas en avance sur une voisine).

Formellement :

Théorème 25 (Nakamura, 1974). *Soit $B = (Q_B, d, V, \delta_B)$ un automate cellulaire déterministe synchrone. Alors il existe un automate cellulaire asynchrone $A = (Q_B \times R, d, V, \delta_A)$ qui simule B . De plus, la fonction $\text{décode}(q_B, r)$ de la définition 17 renvoie simplement q_B .*

En d'autres termes, on peut ajouter de l'information aux états de B et ralentir l'automate (car toutes les transitions ne seront pas des transitions-étapes) de manière à garantir que l'asynchronisme ne change pas la suite des états de chaque cellule. La construction aura cette propriété supplémentaire que l'automate n'est pas ralenti si la dynamique est synchrone, i.e. toutes les transitions sont alors des transitions-étapes.

Nous ne pourrions pas utiliser directement ce théorème, c'est pourquoi nous exposons la construction ci-dessous, puis nous adapterons cette construction à la section 5.3.3.

Démonstration. Posons $R := Q_B \times \{0, 1, 2\}$. Pour $q = (q_b, (q'_b, u)) \in Q_b \times R$, on note

- $q.cur := q_b \in Q_B$,
- $q.pre := q'_b \in Q_B$,
- $q.age := u \in \{0, 1, 2\}$.

Le champ $q.age$ sert en effet à stocker l'âge modulo 3; la différence d'âge entre deux voisins ne sera jamais supérieure à 1. On pose naturellement $\hat{age}(q) := q.age$ pour la fonction \hat{age} de la définition de simulation. Le champ $q.pre$ contient l'état avant le dernier changement d'état, c'est la valeur de $q.cur$ pour la précédente valeur de $q.age$.

Pour une cellule dans l'état q , on définit son état suivant $q' = \delta_A(q_1, \dots, q_{|V|})$ comme suit. S'il existe une cellule q_i en retard, c'est-à-dire telle que $q_i.age - q.age \equiv -1 \pmod{3}$, alors la cellule ne change pas d'état : $q' := q$. Ainsi $\hat{age}(q') = \hat{age}(q)$ et la transition n'est pas une transition-étape. Sinon, soit

$$r_i := \begin{cases} q_i.cur & \text{si } q_i.age = q.age \\ q_i.pre & \text{si } q_i.age - q.age = 1 \pmod{3} \text{ (c'est-à-dire si } q_i \text{ est en avance)} \end{cases}$$

On pose alors :

- $q'.cur := \delta_B(r_1, \dots, r_{|V|})$
- $q'.pre := q.cur$
- $q'.age := (q.age + 1) \pmod{3}$

Ainsi $\hat{age}(q') \neq \hat{age}(q)$ et la transition est une transition-étape.

On définit enfin la configuration initiale de A par $\text{encode}(q_b) := (q_b, (\perp, 0))$ (\perp est un état arbitraire de Q_B qui ne sera jamais lu, on peut prendre par exemple q_b).

Il est alors facile de vérifier que la signification donnée à chacun des champs *cur*, *pre* et *age* est respectée à chaque pas de calcul et donc que les orbites $c_B(\cdot, \cdot)$ et $c'_B(\cdot, \cdot)$ (définies comme à la définition 17) sont identiques. ■

Les auteurs de [Berman and Simon (1988)] ont montré le théorème suivant.

Théorème 26 (Berman et Simon, 1986). *Si les temps entre deux mises à jour successives d'une cellule sont indépendants et de loi exponentielle de moyenne μ , alors*

$$\forall b \exists a \forall t \geq \ln n$$

$$\mathbb{P}(\text{à l'étape } a\mu(\ln |V|)t \text{ chaque cellule a simulé au moins } t \text{ transitions}) \geq 1 - e^{-bt}.$$

De plus, a ne dépend pas de $|V|$ ni de n .

Puisque l'on peut majorer la fonction de répartition d'une loi géométrique par la fonction de répartition d'une loi exponentielle, ce théorème s'applique également au cas où les temps entre deux mises à jour successives sont de loi géométrique, c'est-à-dire au cas que nous appelons la dynamique partiellement asynchrone.

Ce théorème indique en particulier que, sur des configurations finies (n est le nombre de cellules), la construction des marching soldiers ne ralentit le calcul que d'un facteur constant.

Pour construire la fonction f de la définition 17 (définition de simulation, condition 1), soit $m \in \mathbb{N}$, soit $T \in \mathbb{N}$ minimal tel que $T \geq \ln(m + 2T)$, posons $f(m) := T$; on a bien $T = o(m)$. Pour établir la condition

$$\forall t_2 \forall \varepsilon > 0 \exists t_3 \forall m \forall t \geq t_3, f(m) \mathbb{P}(\forall x \in \llbracket 1, m \rrbracket \text{compteur}(x, t) \geq t_2) \geq 1 - \varepsilon$$

sur une configuration infinie, il suffit de considérer un bloc de taille $2t + m$ évoluant pendant t étapes et d'appliquer le théorème 26.

5.3.3 Construction complète

Soit $B = (Q_B, 1, V, \delta_B)$ un automate synchrone probabiliste à un bit. Nous allons construire $A = (Q_A, 1, V, \delta_A)$ un automate asynchrone déterministe qui simule B .

Supposons sans perte de généralité que le voisinage V contient comme premiers vecteurs v_1 et v_2 les voisins de droite et gauche (ces voisins sont nécessaires pour l'automate du théorème 24 lisant les bits aléatoires). Soit donc $L = (Q_L, 1, V_L, \delta_L)$ cet automate.

Posons $Q_A := Q_B \times Q_B \times \{0, 1, 2\} \times Q_L \times \{0, 1, \perp\}$. Les coordonnées d'un état $q \in Q_A$ sont notées $(q.cur, q.prec, q.age, q.rand, q.bit)$. Comme leur nom le laisse deviner, les trois premières coordonnées servent à la construction des marching soldiers, tandis que les deux dernières servent respectivement à produire et stocker un bit aléatoire.

Pour une configuration initiale de B , définissons la configuration initiale de A . Il suffit de poser $\text{encode}(q_b) := (q_b, q_b, 0, 0, \perp)$. La deuxième coordonnée de la configuration initiale ne sera jamais utilisée, on pourrait expliciter ce fait en posant $\text{encode}(q_b) := (q_b, \perp, 0, 0, \perp)$. Intuitivement, la configuration initiale signifie : « L'état courant est q_b ; il n'y a pas d'état précédent; l'âge est congru à 0 modulo 3; l'automate L est dans l'état 0 et il n'a pas encore produit de bit aléatoire. ». Toutes les cellules ont le même âge donc il n'y a pas de cellule en retard.

Pour une cellule dans l'état q , nous définissons le résultat de la règle de transition $q' := \delta_A(q_1, \dots, q_{|V|})$ comme suit.

- $q'.rand := \delta_L(q_1.rand, q.rand, q_2.rand)$: l'automate L effectue sa transition indépendamment des autres coordonnées.

- Si $q.bit = \perp$, c'est-à-dire que le bit aléatoire a été utilisé, alors on pose $q'.bit := \text{read}(q.rand, q'.rand)$, c'est-à-dire que l'on interroge l'automate L pour savoir s'il a produit un bit. Puisqu'il n'y a pas de bit aléatoire disponible, on ne peut pas simuler une transition de B , on pose donc $q'.cur := q.cur$, $q'.prec := q.prec$ et $q'.age := q.age$.

Cette transition n'est pas une transition-étape.

- Sinon, $q.bit \neq \perp$. On utilise alors ce bit pour simuler une transition de B , on pose donc $q'.bit := \perp$. Puis on simule une transition de B , en utilisant la construction des marching soldiers, comme nous le rappelons maintenant.

De la même façon que pour le théorème 25, s'il existe une cellule q_i en retard, c'est-à-dire telle que $q_i.age - q.age \equiv -1 \pmod{3}$, alors la cellule ne change pas d'état : $q'.cur := q.cur$, $q'.prec := q.prec$ et $q'.age := q.age$. À nouveau, la transition n'est pas une transition-étape.

Sinon, soit

$$r_i := \begin{cases} q_i.cur & \text{si } q_i.age = q.age \\ q_i.pre & \text{si } q_i.age - q.age = 1 \pmod{3} \text{ (c'est-à-dire si } q_i \text{ est en avance)} \end{cases}$$

On pose alors :

- $q'.cur := \delta_B(q.bit, r_1, \dots, r_{|V|})$ (Noter l'utilisation du bit aléatoire $q.bit$. C'est la seule différence avec la section 5.3.2.)
- $q'.pre := q.cur$
- $q'.age := (q.age + 1) \pmod{3}$ (c'est donc une transition-étape).

Remarque. Il n'est pas possible simplement de créer la source d'aléatoire par l'automate L puis dans un second de temps rendre l'automate synchrone à l'aide du théorème 25. En effet, la construction des marching soldiers perturberait la lecture des bits aléatoires : interdire la mise à jour de certaines cellules introduirait un biais vers le résultat « les deux cellules n'ont pas été mises à jour en même temps ». Dans notre construction, la mise à jour selon la règle δ_L est toujours effectuée.

5.3.4 Autres constructions

Variante Pour mesurer le bit aléatoire, au lieu de la construction présentée ici, on pourrait doubler le nombre de cellules. Une cellule sur deux sert alors à simuler l'automate de départ, tandis que chaque cellule restante est au service de sa voisine pour mesurer les bits aléatoires. Mais les cellules auraient alors des rôles différents, nous avons préféré une construction uniforme. De plus, quitte à augmenter le nombre d'états, nous avons préféré ne pas augmenter également le nombre de cellules.

Optimisation Nous avons cherché une règle raisonnablement simple. Il est certainement possible de concevoir une règle plus économe en nombre d'états, peut-être en fusionnant les états 0 et 3.

Il est aussi possible de concevoir une règle plus efficace en nombre de transitions pour obtenir un bit (mais ayant plus d'états), par exemple à l'aide d'un automate partitionné. Chaque état devient un couple, appartenant à l'ensemble $\{3, 4\} \times \{0, 1, 2\}$ (d'où le nom partitionné). La table de transition est alors la suivante :

Pour la partie gauche (qui aide le voisin de gauche), dont l'état appartient à $\{3, 4\}$:

		voisin de gauche		
		(_, 0)	(_, 1)	(_, 2)
état	(3, _)	3	4	3
	(4, _)	3	4	4

Pour la partie droite (qui effectue la mesure), dont l'état appartient à $\{0, 1, 2\}$:

		voisin de droite	
		(3, _)	(4, _)
état	(_, 0)	1	0
	(_, 1)	2	0,>
	(_, 2)	0,<	0,=

5.4 Conclusion et perspectives

Nous cherchions une validation formelle des observations de percolation dirigée des chapitres précédents. Nous avons prouvé une simulation mutuelle entre les automates asynchrones et une classe d'automates probabilistes, les automates probabilistes à un bit. Nous avons montré que cette classe contient une proportion « raisonnable » des automates probabilistes habituellement utilisés. Notamment l'automate de Domany-Kinzel, reconnu comme modèle de percolation dirigée, fait partie de cette classe (pour certains choix des paramètres). Nous pouvons donc conclure que certains automates asynchrones subissent une transition de phase ayant le même exposant critique β que celui de la percolation dirigée.

Nous avons pour ce résultat été amené à proposer une définition de simulation préservant la densité entre automates cellulaires utilisant une source d'aléatoire (tels que les automates cellulaires asynchrones et les automates cellulaires probabilistes).

Nous avons également construit un automate lisant une source d'aléatoire dans la suite des mises à jour, ce qui est un exemple de programmation d'un système asynchrone. L'idée pour cette mesure est simplement de mesurer si deux cellules voisines ont été mises à jour en même temps ou bien l'une après l'autre. C'est donc l'automate à simuler qui impose le choix de la probabilité de mise à jour α , ce qui peut être considéré comme une faiblesse de la simulation, mais est aussi un moyen de tirer parti de α lorsque l'on peut le choisir.

Nous avons fait une construction en une dimension. La généralisation aux dimensions supérieures ne semble pas poser de problème, il suffit de privilégier une dimension pour définir les termes « voisin de droite » et « voisin de gauche », et d'utiliser des boules au lieu des intervalles dans la preuve du théorème 18.

Perspectives Nous avons prouvé la conservation de l'exposant critique β , mais celle des autres exposants reste à étudier afin de garantir la même classe d'universalité, ici celle de la percolation dirigée. Le présent travail porte sur les automates asynchrones (où l'on considère une seule configuration) et non sur les modèles de couplage des chapitres 3 et 4, ni sur les automates élémentaires, dans lesquels la percolation dirigée est également observée. C'est donc une direction naturelle pour de futurs travaux.

Nous avons également dû introduire une sous-classe des automates probabilistes : les automates probabilistes à un bit. Il semble que cette classe n'ait pas déjà été explicitement définie, même si elle est parfois évoquée [Boccaro, Fuk s, and Geurten (1997)] et si

de nombreux automates en font partie. Une question naturelle est donc : quelle est la puissance de calcul de cette classe par rapport à d'autres classes habituelles ?

Enfin, la notion de simulation introduite gagnerait à être simplifiée (tant que le théorème 18 est valable). Elle est de plus à explorer : correspond-elle à l'idée intuitive d'«équivalence» entre automates ? C'est-à-dire, permet-elle de relier des automates qui nous semblent équivalents ? Quelles propriétés sont préservées et quelles propriétés ne sont pas préservées par cette simulation ? Cette notion permet à l'automate simulant d'effectuer plus de pas de calculs que l'automate simulé, on peut parler de «groupage temporel». On se demande alors si l'on peut étendre cette notion, comme évoqué plus haut, en utilisant la notion plus générale de groupage.

Remerciements Nous souhaitons remercier Nazim Fatès pour les discussions qui ont précisé le but de ce chapitre, et pour avoir relu une version intermédiaire.

Deuxième partie

Perturbations de la topologie

Dans cette partie nous nous intéressons à des modifications de la topologie. La topologie désigne dans notre travail la façon dont est structuré l'espace et donc, dans le cas des automates cellulaires, le graphe d'interaction entre les cellules. La topologie généralement utilisée pour les automate cellulaire est la topologie régulière \mathbb{Z}^d ou $(\mathbb{Z}/n\mathbb{Z})^d$. Une majorité des articles de la littérature concernent la dimension 1, une partie s'intéressent à la dimension 2, plus difficile et dont l'étude suffit souvent à la généralisation à toute dimension. C'est cette topologie que nous modifions.

Nous étudions de plus comment des perturbations de la topologie se combinent avec la perturbation du synchronisme étudiée précédemment. Combiner les deux perturbations nous a semblé plus riche que d'étudier seulement la perturbation de la topologie (même si un travail futur pourra se pencher plus spécifiquement sur ce point). En particulier, cela ouvre des questions telles que « les perturbations se combinent-t-elles de façon linéaire ? », « existe-t-il des automates cellulaires robustes à plusieurs perturbations ? », « une perturbation rend-elle l'automate cellulaire robuste à l'autre perturbation ? ».

Pour valider cette approche, nous commençons par considérer (dans le chapitre 6), l'ensemble des automates cellulaires élémentaires. Nous introduisons quelques défauts dans la topologie $(\mathbb{Z}/n\mathbb{Z})^d$ et restons donc assez proche de la topologie habituelle. Un but de ce chapitre est de répondre aux questions précédentes en étudiant les différents comportements possibles et de découvrir d'éventuels comportements inattendus. Le but principal est de vérifier que combiner ces deux perturbations a du sens, c'est pourquoi il est assez court.

Ce chapitre est purement expérimental.

Au chapitre 7, nous comparons différentes topologies plus exotiques. Quelques résultats simples sur des topologies régulières (cycle, graphe complet) sont énoncés puis nous nous concentrons sur la famille de graphes que sont les arbres : c'est une topologie simple, sans boucles, adaptée à une première étude. Des topologies plus évoluées pourront faire l'objet de prochains travaux.

Pour permettre la comparaison de ces topologies, nous appliquons la même règle à chacune. Cette règle doit à la fois avoir un comportement non trivial et être définie quel que soit le nombre de voisins. La règle minorité remplit ces critères.

Nous étudions alors deux points : le temps transitoire, ou temps pour atteindre l'ensemble limite (aussi appelé *hitting time*), et la structure de ce dernier.

Ce chapitre est purement théorique.

Chapitre 6

Perturbation de la topologie en 1D

L'asynchronisme étudié à la partie **I** est sans doute la perturbation la plus courante sur les automates cellulaires, comme nous l'avons vu à la page **12**. La littérature contient aussi d'autres perturbations, mais le plus souvent une seule perturbation à la fois est étudiée. Cependant, dans [Fatès and Morvan (2004)], les auteurs montrent que combiner deux perturbations (en l'occurrence, introduire de l'asynchronisme et perturber la topologie) fait émerger un comportement intéressant dans le jeu de la vie : le jeu de la vie est sensible à l'asynchronisme, mais cette sensibilité est fortement réduite lorsque l'on perturbe la topologie (précisément, lorsque l'on coupe une petite proportion de liens dans la grille). Le but de ce chapitre est d'étudier exhaustivement la combinaison de ces deux perturbations sur les automates cellulaires élémentaires.

On peut voir le fait de ne pas mettre une cellule à jour comme un défaut de la cellule, ou simplement comme un retard avant la prochaine mise à jour. L'asynchronisme est donc une perturbation des nœuds. Le pendant d'un défaut d'un nœud du graphe (i.e. d'une cellule) est alors un défaut sur un lien : c'est ce que nous appelons une perturbation de la topologie. Ce chapitre étudie donc l'interaction d'une perturbation sur les nœuds et d'une autre sur les liens.

À la différence de [Fatès and Morvan (2004)], tous les défauts introduits sont temporaires : une cellule non mise à jour n'a pas la mémoire de ce défaut et fera un nouveau tirage au sort à l'étape suivante pour décider de se mettre à jour ou non. De même, un lien n'est coupé que le temps d'une étape ; l'état de chaque lien est tiré au sort à chaque étape. Nous utilisons donc des perturbations *dynamiques*. Cela est nécessaire afin de ne pas diviser l'espace en plusieurs composantes isolées. En effet, pour les automates cellulaires élémentaires, un défaut permanent sur une cellule ou sur un lien déterminerait une partie gauche et une partie droite entre lesquelles aucune information ne peut circuler.

Le plan de ce chapitre est simple : à la section **6.1**, nous présentons le modèle et le protocole expérimental ; à la section **6.2** nous présentons les résultats observés et classifions les différents comportements.

Ce travail a été publié dans [Rouquier and Morvan (2008)].

6.1 Protocole expérimental

Comme dans les chapitres précédents, nous voulons étudier les différents comportements possibles face à une perturbation, nous devons donc appliquer ces perturbations

sur un ensemble d'automates, et nous choisissons à nouveau la classe des automates élémentaires.

Nous perturbons ce modèle de deux manières. Premièrement, nous nous plaçons en dynamique partiellement asynchrone, en faisant varier α , qui est la probabilité pour une cellule d'être mise à jour. Le cas $\alpha = 1$ est donc le cas non perturbé (synchrone).

Deuxièmement, nous perturbons le graphe en coupant certains liens. Quand un lien est coupé, le graphe d'interaction n'est plus connexe (rappelons que l'on utilise les deux plus proches voisins). Nous décidons donc que chaque coupure de lien n'est effective que pour une étape de temps. Ainsi, à chaque étape, quand la cellule x demande l'état de ses voisines pour calculer son nouvel état, elle ne reçoit pas de réponse avec probabilité r . Elle suppose alors par défaut que l'état de cette voisine est q . q et r sont des paramètres du modèle, que nous feront varier. Le cas $r = 0$ (aucune coupure de lien) est le cas non perturbé. Entre deux cellules données, il est donc possible que la communication soit coupée dans un sens mais reste fonctionnelle dans l'autre. Du point de vue de la cellule, elle ne « voit » pas toujours ses voisins.

L'évolution est donc :

$$c_{t+1} := \begin{cases} \delta(c'_t(x-1), c_t(x), c'_t(x+1)) & \text{avec probabilité } \alpha \\ c_t(x) & \text{avec probabilité } 1 - \alpha \end{cases} \quad (6.1)$$

$$\text{où } c'_t(y) := \begin{cases} q & \text{avec probabilité } r \\ c_t(y) & \text{avec probabilité } 1 - r \end{cases}$$

Paramètres étudiés

Lors d'un travail sur la classe des automates élémentaires, on restreint généralement l'étude à 88 règles, pour deux raisons :

1. Si deux règles δ et δ' sont identiques modulo une symétrie de la configuration, c'est-à-dire si $\forall q_1, q_2, q_3 \quad \delta(q_1, q_2, q_3) = \delta'(q_3, q_2, q_1)$, alors en connaissant les diagrammes espaces-temps de δ on peut déduire ceux de δ' par une simple symétrie. Il n'est alors pas nécessaire d'étudier δ' . Ce point reste valable dans notre étude.
2. Si deux règles δ et δ' sont identiques modulo un échange des deux états 0 et 1, c'est-à-dire si $\forall q_1, q_2, q_3 \quad \delta(q_1, q_2, q_3) = 1 - \delta'(1 - q_1, 1 - q_2, 1 - q_3)$, alors en connaissant les diagrammes espaces-temps de δ on peut déduire ceux de δ' par un simple échange des deux états. Il n'est alors pas nécessaire d'étudier δ' . Mais cette remarque n'est pas applicable dans notre étude.

En effet, le paramètre q a introduit une asymétrie entre les deux états, l'un des deux est « favorisé ». On ne peut plus déduire les diagrammes espaces-temps d'une règle en connaissant ceux de l'autre, et nous devons donc étudier plus de 88 règles.

Ce paramètre q est le premier que nous faisons varier. Pour chacune des règles restant après le point 1, nous étudions les deux valeurs possibles de q , 0 et 1. Le paramètre q est noté en indice, par exemple 110_0 et 110_1 . La symétrie du point 2 reste valable pour les règles *auto-conjuguées*, i.e. les règles qui sont inchangées lorsque l'on échange les états 0 et 1. Pour ces règles, les diagrammes espaces-temps obtenus pour $q = 1$ se déduisent directement de ceux obtenus pour $q = 0$, par un échange des deux états. Nous n'étudions donc que le cas $q = 0$, noté par exemple 105_a (a pour « auto-conjugué »).

Regardons maintenant les autres paramètres. Pour une règle fixée (y compris le paramètre q), nous échantillons plusieurs valeurs de α et r : nous combinons plusieurs taux de synchronisme et de perturbation de la topologie. Pour α , nous échantillons

tout l'espace des valeurs possibles, y compris le cas synchrone, par pas de 0,01 : $\alpha \in \{0,01 ; 0,02 ; \dots 1,00\}$. Pour r , couper 10% des liens est déjà une forte perturbation [Fatès and Morvan (2004)] et nous ne perturbons pas plus le modèle. Ainsi, r parcourt les valeurs $\{0,00 ; 0,01 ; 0,02 ; \dots 0,10\}$.

Nous faisons un lancer pour chaque combinaison de ces paramètres.

Paramètres fixes

Les paramètres suivant sont identiques pour tous les lancers.

- Le nombre de cellules est $n = 10\,000$. Nous avons vu au chapitre 3 que $n = 2\,000$ semble suffisant sur un problème proche.
- La configuration initiale est distincte pour chaque lancer et aléatoire : chaque cellule est dans l'état 0 avec probabilité $\frac{1}{2}$, indépendamment des autres cellules.
- Pour accélérer les calculs, la densité ρ n'est pas mesurée à chaque étape, mais environ 200 fois par décade (c'est-à-dire environ 200 fois entre les étapes t et $10t$).
- Pour détecter le fait que la densité a atteint une valeur constante, nous faisons une régression linéaire sur l'historique des 200 derniers points de mesure et attendons d'obtenir une ligne horizontale. Nous calculons alors la densité moyenne sur cet historique. Pour être plus précis, nous faisons en réalité 3 régressions linéaires indépendantes sur les trois tiers de cet historique, plus trois régressions linéaires sur respectivement les deux premiers tiers, les deux derniers tiers, et l'historique complet. Nous attendons alors que toutes ces droites aient une pente nulle à 0,01 près. Puisque la densité ρ peut se stabiliser sur des valeurs ayant plusieurs ordres de grandeur de différence (par exemple 0,001 ou 0,1), nous calculons la pente sur un diagramme en échelle doublement logarithmique (c'est-à-dire que nous faisons un régression linéaire de $\log \rho$ en fonction de $\log t$). Ainsi, la précision sur la densité asymptotique est constante entre les mesures.
- Si la densité ne s'est pas stabilisée après 10^6 étapes, nous arrêtons la simulation et ignorons ce point.

Limites du protocole

La densité est sans doute une mesure assez grossière, ce qui nous fait manquer certains phénomènes se produisant dans ce modèle. Cependant, la variété des résultats obtenus montre que cette mesure est capable de détecter au moins certains types de sensibilités aux perturbations.

Une autre limite est que ce protocole ne mesure pas la vraie densité asymptotique, mais seulement une régime suffisamment long où la densité est stable. Ce point a été discuté à la section 1.3.1. Néanmoins, la décision d'arrêter un lancer peut être prise prématurément si la densité décroît exceptionnellement lentement.

Enfin, pour chaque combinaison des paramètres, nous ne faisons qu'un lancer et n'obtenons donc qu'un point de mesure. Mais la régularité de la plupart des surfaces montre que la variance est faible entre les lancers et donc qu'il n'est pas nécessaire de moyenner sur plusieurs lancers pour obtenir un point de mesure plus précis.

6.2 Étude exhaustive des automates élémentaires

Nous appliquons maintenant ce protocole à chaque automate élémentaire et traçons ρ_∞ en fonction de α et r (figure 6.1). Passer en revue les surfaces obtenues suggère la

classification en quatre classes de la table table 6.1, basée sur la sensibilité à chaque perturbation. Nous décrivons maintenant chaque classe.

		Perturbation de la topologie	
		insensible	sensible
Perturbation du synchronisme	insensible	0 ₀ 0 ₁ 2 ₀ 8 ₀ 8 ₁ 10 ₀ 23 _a	4 ₁ 5 ₀ 5 ₁ 12 ₁ 13 ₀ 15 _a 29 ₀ 29 ₁ 32 ₁ 36 ₁ 40 ₁ 44 ₁
		24 ₀ 28 ₀ 32 ₀ 34 ₀ 40 ₀	72 ₀ 72 ₁ 76 ₁ 78 ₁ 104 ₀ 104 ₁ 132 ₁ 140 ₁ 160 ₁
		42 ₀ 51 _a 56 ₀ 60 ₀ 60 ₁	164 ₁ 168 ₁ 170 _a 172 ₁ 184 ₀ 184 ₁ 200 ₀ 232 _a
		74 ₀ 105 _a 128 ₀ 128 ₁	
		130 ₀ 136 ₀ 136 ₁ 138 ₀	
	152 ₀ 154 ₁ 160 ₀ 162 ₀		
	168 ₀ 200 ₁ 204 _a		
	sensible	1 ₀ 4 ₀ 6 ₀ 9 ₀ 11 ₀ 11 ₁	a/ 2 ₁ 6 ₁ 10 ₁ 24 ₁ 34 ₁ 38 ₁ 42 ₁ 45 ₀ 45 ₁ 46 ₁
		12 ₀ 18 ₀ 19 ₀ 19 ₁ 22 ₀	56 ₁ 73 ₀ 73 ₁ 74 ₁ 77 _a 94 ₀ 94 ₁ 130 ₁ 134 ₁
		26 ₀ 33 ₀ 35 ₀ 35 ₁ 36 ₀	138 ₁ 152 ₁ 154 ₀ 162 ₁
38 ₀ 41 ₀ 41 ₁ 44 ₀ 46 ₀		b/ 18 ₁ 26 ₁ 50 ₁ 58 ₁ 106 ₀ 106 ₁ 146 ₁ 178 _a	
50 ₀ 54 ₀ 58 ₀ 76 ₀ 90 ₁		c/ 1 ₁ 3 ₀ 3 ₁ 7 ₀ 7 ₁ 9 ₁ 13 ₁ 14 ₀ 14 ₁ 22 ₁ 25 ₀	
108 ₀ 110 ₀ 110 ₁ 132 ₀		25 ₁ 27 ₀ 27 ₁ 28 ₁ 30 ₀ 30 ₁ 33 ₁ 37 ₀ 37 ₁	
134 ₀ 146 ₀ 164 ₀		43 _a 54 ₁ 57 ₀ 57 ₁ 62 ₀ 62 ₁ 78 ₀ 90 ₀ 108 ₁	
		122 ₀ 122 ₁ 126 ₀ 126 ₁ 140 ₀ 142 _a 150 _a	
		156 ₀ 156 ₁ 172 ₀	

TAB. 6.1 – Classification des automates élémentaires selon leur robustesse à chacune des perturbations. Les sous-classes a/, b/ et c/ sont détaillées à la section 6.2.4.

6.2.1 Règles insensibles aux deux perturbations

La plupart des règles de cette classe ont une surface qui est simplement 0 partout (figure 6.1.a), parfois avec une exception pour le cas non perturbé ($\alpha = 1$, $r = 0$). Il est à noter que quelques règles (environ 9%) dans les autres classes ont également un comportement notablement différent pour le point de mesure du cas non perturbé. Mais comme nous nous intéressons dans ce chapitre à l'interaction de deux perturbations et non aux effets d'une seule, nous laissons ce point pour de futurs travaux.

Puisqu'elles sont par définition insensibles aux deux perturbations, les autres règles de cette classe ont aussi une surface horizontale, dont l'altitude est souvent autour de $\rho_\infty = 0,5$.

6.2.2 Règles sensibles seulement à la perturbation du synchronisme

Certaines règles de cette classe subissent une transition de phase quand α varie, quel que soit r (figure 6.1.b). À la section 2.2 nous avons exposé une conjecture de Grassberger sur les transitions de phase. Les conditions de cette conjectures sont remplies ici, ces transitions devraient donc appartenir à la classe d'universalité de la percolation dirigée. Cependant, vérifier ce point demanderait de lourds calculs pour atteindre la précision nécessaire, d'autant plus qu'il faudrait mesurer les exposants critiques pour plusieurs règles et plusieurs valeurs de r . Nous avons préféré nous concentrer sur les enseignements nouveaux par rapport aux chapitres précédents traitant de percolation dirigée.

Il est intéressant de noter que pour la plupart de ces règles, la perturbation de topologie décale légèrement le seuil (la position de la transition), plus ou moins selon r .

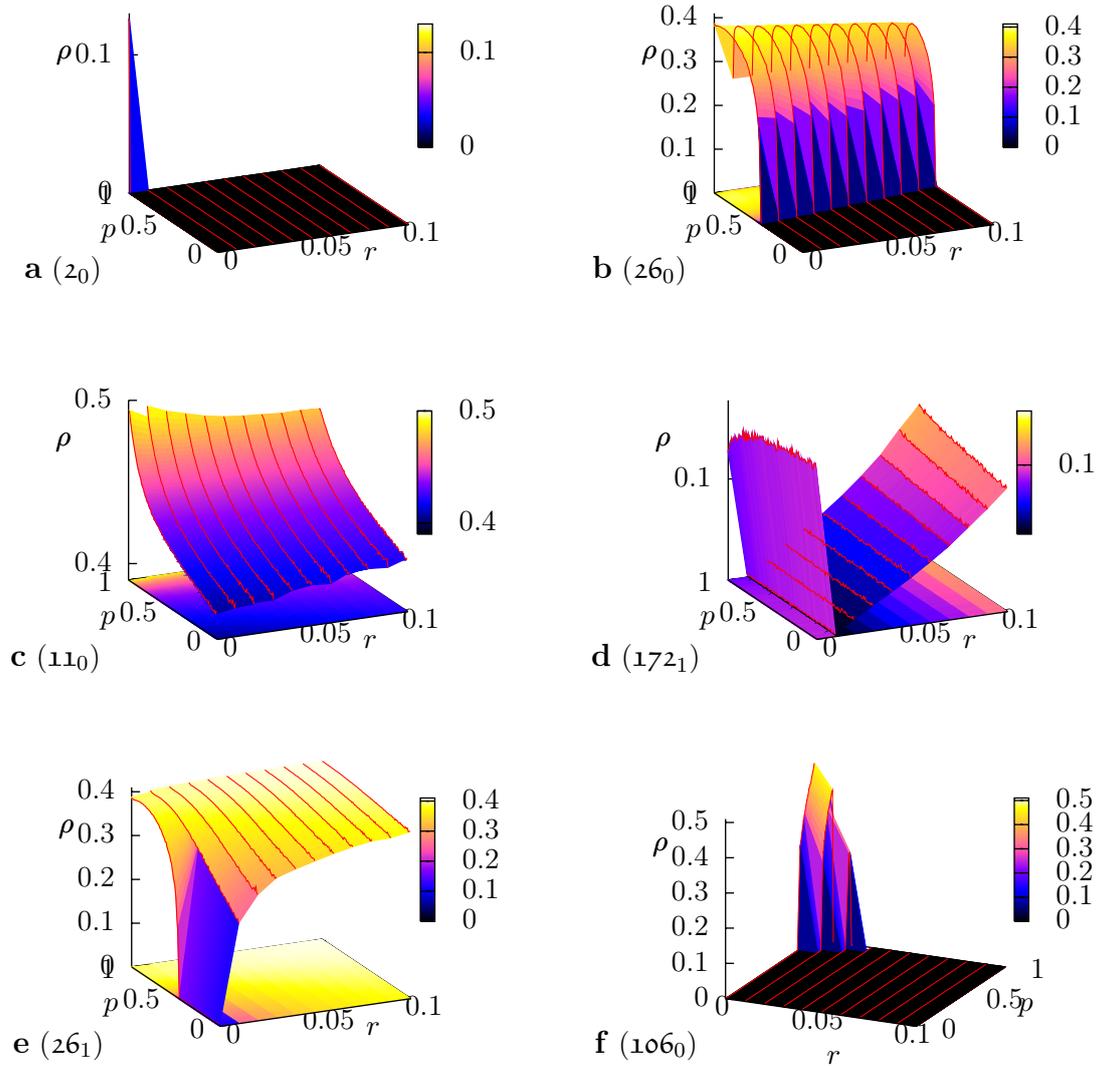


FIG. 6.1 – Quelques surfaces d'échantillonnage. L'axe r représente la perturbation de la topologie, l'axe p (parallèle aux 11 lignes sur la surface) celle du synchronisme l'axe z (en hauteur) la densité asymptotique ρ_∞ . La différence entre les figures 6.1.a et 6.1.f est que dans la première seulement le cas non perturbé ($\alpha = 1, r = 0$) est différent de zéro, tandis que dans la seconde il y a une transition continue entre ρ nul et non nul en fonction de α et r .

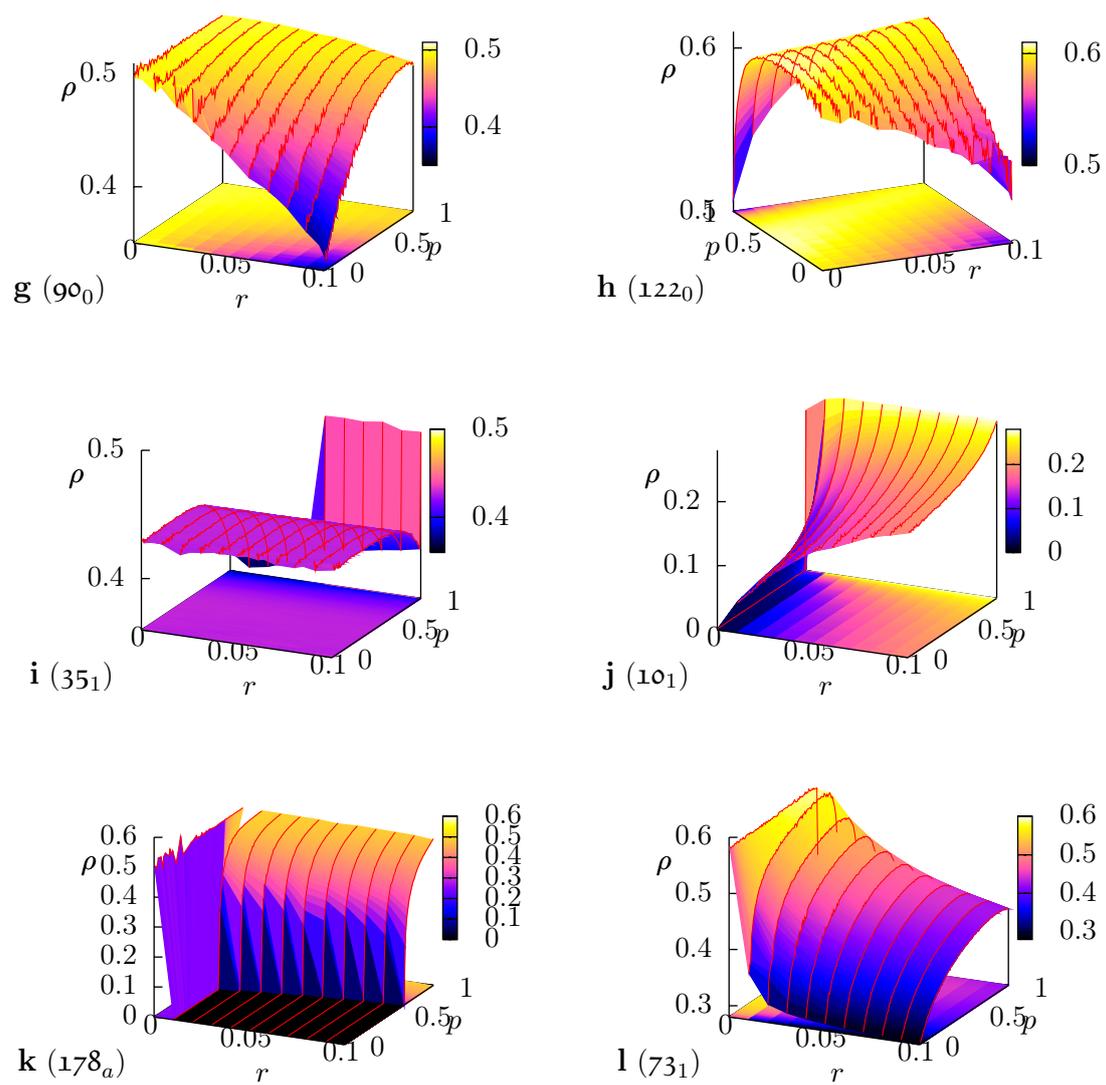


FIG. 6.2 – Suite de la figure 6.1.

Ces règles sont donc en réalité très légèrement sensibles à l'autre perturbation.

Les autres règles de cette classe ont une surface régulière, telle que celle de la figure 6.1.c. Puisque ces règles sont presque insensibles à la perturbation de la topologie, toutes les coupes le long d'une des lignes dessinées sur la surface en rouge (gris pour une version noir et blanc) sont presque identiques.

Enfin, deux règles font exception : pour les règles 35_0 et 35_1 (figure 6.2.i), il y a un saut entre le cas synchrone et le cas asynchrone, quel que soit r . En effet, en $\alpha = 1$, $\rho_\infty = 0,5$ tandis qu'en $\alpha = 0,99$, $\rho_\infty \simeq 0,43$.

6.2.3 Règles sensibles seulement à la perturbation de la topologie

Certaines règles de cette classe sont sensibles seulement à l'introduction de la perturbation de la topologie, i.e. elles montrent un changement brutal de comportement entre $r = 0$ et $r > 0$ (figure 6.1.d). Quelques règles ont en fait une densité qui dépend de α , mais seulement en $r = 0$. Pour ces règles, perturber la topologie a donc annulé la sensibilité à l'asynchronisme.

Les autres surfaces de cette classe sont régulières. Les seules transitions de phase de la présente section 6.2.3 sont donc des transitions de phase de premier ordre (c'est-à-dire que la densité ρ est discontinue) et elles se produisent toutes à l'introduction de la perturbation.

6.2.4 Règles sensibles aux deux perturbations

Cette classe est la plus intéressante, car nous allons voir de nombreux comportements différents. Elle nous permet d'examiner l'interaction des deux perturbations. La richesse de cette classe nous a amené à la découper en trois sous-classes, que voici.

a/ Règles sensibles à l'introduction de la perturbation topologique

Les règles de cette sous-classe montrent un saut entre $r = 0$ et $r > 0$, comme pour la classe 6.2.3. Pour les valeurs de r non nulles, ces règles restent sensibles à l'asynchronisme : la surface n'est pas horizontale (figure 6.2.j).

b/ Transition de phase créée ou détruite

Nous avons vu à la section 6.2.2 que l'une des trois façons pour la densité de varier quand α change est une transition de phase. Pour les règles de la section 6.2.2, cette transition de phase était robuste à la perturbation topologique. Nous voyons ici un exemple du contraire : la transition de phase est détruite quand $r > 0$ (figure 6.1.e). En regardant de plus près les configurations, il semble que couper un lien dans une région où toutes les cellules sont dans l'état 0, qui est l'état absorbant pour cette transition, provoque l'apparition d'un état 1, c'est-à-dire « réensemence » la configuration avec des états 1 qui peuvent se propager. La configuration est donc en permanence réensemencée avec des sites actifs pouvant se propager et 0^n n'est plus un état absorbant. C'est le cas des règles 18_1 , 26_1 , 50_1 , 58_1 , 106_1 et 146_1 .

Le cas de 178_a (figure 6.2.k) est à l'opposé : quand la topologie n'est pas perturbée, cette règle est auto-conjuguée et ne peut avoir un état préféré, donc ne peut avoir une transition de phase de percolation dirigée. Dans [Fatès (2004)], sans perturbation de topologie, il est suggéré que cette règle peut avoir une transition de phase (quand α

varie) de la classe d'universalité \mathbb{Z}_2 symétrique. La plus petite perturbation de topologie introduit un biais vers l'état 0, et 0^n devient le seul état absorbant. La règle satisfait alors aux conditions de la conjecture de Grassberger et est donc sans doute dans la classe d'universalité de la percolation dirigée. Nous pouvons conclure que perturber la topologie a créé une nouvelle transition de phase.

c/ Surface régulière

La dernière sous-classe contient les autres règles, dont la surface est régulière.

Quelques règles remarquables

Nous voudrions pour conclure attirer l'attention sur quelques règles remarquables de la présente section 6.2.4.

La règle 106₀ (figure 6.1.f) subit deux transitions de phases, l'une quand α varie et l'autre quand r varie.

La règle 90₀ (figure 6.2.g) est presque insensible à l'une ou l'autre des perturbations si la perturbations est isolée. Cependant, les deux perturbations combinées provoquent une baisse de la densité notable. (Il est facile, cf. ci-dessous, de démontrer que la règle 90₀ tend vers 0^n et donc une densité nulle pour r assez grand, quel que soit α . Mais le présent travail s'intéresse à de petites perturbations de la topologie.)

Pour montrer que la règle 90₀ tend vers 0^n et donc une densité nulle pour r assez grand, nous utilisons un argument analogue à celui du fait 5 page 29 : nous considérons les états 1 et leur propagation. Si u est le nombre de cellules dans l'état 1 à une étape, les cellules pouvant être dans cet état à l'étape suivante sont :

- les cellules dans l'état 1 et non mises à jour. Il y en a en moyenne $(1 - \alpha)u$;
- les cellules voisines d'une cellule dans l'état 1 et mises à jour. Il y en a en moyenne au plus $\alpha 2u$. Mais si $r^2 > \frac{1}{2}$, chacune de ces cellules a plus d'une chance sur deux de ne voir aucune de ses deux voisines, donc de passer dans l'état 0.

L'espérance du nombre de cellules dans l'état 1 à l'étape suivante est donc au plus $(1 - \alpha)u + \alpha 2u(1 - r^2) < (1 - \varepsilon)u$ pour un certain $\varepsilon > 0$. L'espérance du nombre de cellules dans l'état 1 tend donc vers 0 exponentiellement vite.

La règle 122₀ (figure 6.2.h) a une densité proche de 0,5 dans le cas non perturbé. En appliquant l'une des deux perturbations, la densité tend vers 0,6. Mais en appliquant les deux, la densité tend à nouveau vers 0,5. Un examen des configurations montre que dans ce cas elles sont assez différentes du cas non perturbé. Seule la densité a été restaurée. Mais si l'on ne s'intéresse qu'à la densité, on peut voir une perturbation comme un moyen de rendre l'automate robuste à l'autre perturbation. C'est l'un des rares cas où les deux perturbations ne se combinent pas de manière monotone.

Les règles 73₀ et 73₁ (figure 6.2.i) ont une densité asymptotique insensible à α quand la topologie n'est pas perturbée (sauf en $\alpha = 1$). Avec une petite perturbation de la topologie, ρ parcourt alors l'intervalle $[0,3; 0,5]$. L'effet intéressant est que perturber la topologie a rendu la règle sensible à la perturbation du synchronisme. La même remarque s'applique aux règles 14₀, 14₁, 57₀, 90₀, 150_a, 156₀, 156₁ et à presque toutes les règles de la sous-classe a/.

Le comportement opposé existe également : la règle 45, en l'absence de perturbation topologique, a une densité qui dépend de α . Pour $r > 0$, la densité devient indépendante de α . Perturber la topologie rend donc la règle robuste à la perturbation du synchronisme.

6.3 Conclusion et perspectives

Nous avons donc étudié les interactions des perturbations du synchronisme et de la topologie au moyen de perturbations dynamiques, et proposé un découpage en quatre classes de l'espace des règles selon leur réponse à chacune des deux perturbations. Il ne semble pas exister de relations entre cette classification et d'autres classifications des automates cellulaires. D'ailleurs, une même règle peut être dans deux classes différentes (parmi les quatre introduites) selon le paramètre q .

La mesure macroscopique choisie, la densité, permet déjà d'observer un riche ensemble de comportements, même s'il y a sans doute d'autres choses à découvrir que ce que révèle cet observable. Parmi ces comportements, certains étaient prévisibles : il existe les quatre combinaisons possibles de sensibilité ou robustesse à chacune des deux perturbations : robuste aux deux, sensible à l'une ou l'autre, sensible aux deux. Mais des comportements inattendus sont également apparus, dans la classe des automates cellulaires sensibles aux deux perturbations. Par exemple, la combinaison des deux perturbations crée parfois ou détruit une transition de phase induite par l'une des deux perturbations seule. De plus, la règle 45 montre le même comportement que le jeu de la vie : une perturbation rend le système robuste à l'autre perturbation (la robustesse à l'asynchronisme est augmentée lorsque la topologie n'est plus régulière), ainsi les deux perturbations ne se combinent pas de façon monotone. Dit autrement, combiner les deux perturbations est moins efficace que de n'utiliser qu'une perturbation. Mais ce comportement semble rare.

Perspectives

La perturbation topologique peut être vue comme une perturbation des cellules voisines par opposition à la perturbation du synchronisme qui est une perturbation de la cellule centrale. Il serait intéressant dans de futurs travaux d'étudier isolément cette perturbation, avec la généralisation suivante du paramètre q . Quand une cellule ne connaît pas l'état d'une voisine, elle suppose dans le présent travail qu'elle est dans l'état q . Elle pourrait à la place décider de l'état de sa voisine selon une loi de Bernoulli, de paramètre q . On pourrait alors étudier différentes valeurs des paramètres r et q .

D'autres futurs travaux pourront vérifier que, lorsque la surface contient la courbe caractéristique d'une transition de phase de second ordre (section 6.2.2), cette transition de phase appartient bien à la classe d'universalité de la percolation dirigée conformément à la conjecture de Grassberger.

En conclusion, combiner ces deux perturbations a du sens : les comportements observés ne sont ni triviaux ni trop complexes pour être appréhendés.

Chapitre 7

Perturbations de la topologie pour la règle minorité

En complément de l'étude expérimentale du chapitre précédent, nous souhaitons dans ce chapitre étudier des aspects théoriques de la combinaison des perturbations du synchronisme et de la topologie. Par une étude comparative de plusieurs graphes sous la dynamique asynchrone, nous espérons mieux comprendre l'influence de la topologie. Voici comment nous avons choisi les ingrédients du modèle.

Choix de la règle Pour perturber la topologie, nous devons choisir une règle qui soit à la fois intéressante et adaptable à un nombre de voisins variable. Beaucoup de règles proposées dans la littérature ne peuvent pas être généralisées à une topologie arbitraire : certaines règles définies sur la ligne en 1D ou la grille en 2D font référence à une orientation particulière, par exemple « voisin du haut » ou de « gauche » dans leur définition. Néanmoins, une large classe d'automates cellulaires ont une définition indépendante du graphe : ce sont les automates cellulaires *totalisants*. La règle de mise à jour considère seulement le nombre de voisins dans chacun des états, et éventuellement l'état de la cellule mise à jour. Le classique jeu de la vie en est un exemple. La règle minorité est une règle totalisante.

D'où vient l'idée de la règle minorité ? C'est l'un des automates cellulaires les plus simples avec des interactions de poids négatif. Considérons un problème d'allocation de fréquences simplifié : chaque nœud représente une station radio qui doit choisir entre deux fréquences d'émission notées 0 et 1. Les liens du graphe représentent les interférences possibles entre deux stations si elles émettent sur la même fréquence. Imaginons que chaque cellule mette à jour souvent sa fréquence d'émission en cherchant à minimiser les interférences, selon un algorithme simple : elle choisit la fréquence la moins utilisée parmi ses voisins. C'est la règle minorité et l'évolution des fréquences utilisées est la dynamique de la règle minorité.

Un autre exemple peut être trouvé dans un réseau de régulation génétique, voir par exemple [Demongeot, Aracena, Thuderoz, Baum, and Cohen (2003)]. Dans un réseau de régulation génétique, chaque gène, quand il est *exprimé*, exerce une activation ou une inhibition sur d'autres gènes (qui sont donc ses voisins dans le graphe d'interaction). L'état d'un gène indique s'il est exprimé ou non, ce qui est modélisé par 0 ou 1. Étudions le cas particulier où tous les gènes n'exercent que des inhibitions. Si l'on simplifie en considérant que tous les poids sont identiques et qu'un gène est exprimé dès que moins de la moitié de ses voisins sont exprimés, on obtient à nouveau la règle minorité.

Ces deux modèles sont bien sûr des simplifications extrêmes, mais comme annoncé le but est de comprendre les phénomènes au cœur du systèmes, étape nécessaire avant une compréhension plus spécifique de modèles évolués.

La règle minorité a été bien étudiée sur des graphes arbitraires en dynamique synchrone. [Goles and Martinez (1990)] montrent notamment que pour tout graphe le système converge vers un point fixe ou un cycle de période 2. Mais derrière cette description simple, cette règle présente un comportement complexe lorsque l'on introduit l'asynchronisme. De tels comportements complexes ont été observés et prouvés par [Regnault, Schabanel, and Thierry (2007), Regnault, Schabanel, and Thierry (2008)] où est étudiée la topologie du tore. Certains modèles stochastiques tels que les modèles d'Ising ou les réseaux de Hopfield ont été étudiés en dynamique asynchrone (la dynamique totalement asynchrone correspond dans le modèle d'Ising à la limite quand la température tend vers 0). Ces modèles sont connus pour être plus difficiles sur des graphes arbitraires [Istrail (2000), McCoy and Wu (1973)] ou avec poids négatifs [Rojas (1996)].

Choix des autres paramètres La méthode de mise à jour asynchrone la plus simple est la dynamique totalement asynchrone (à chaque étape, tirer au sort une cellule et la mettre à jour), c'est donc celle que nous retenons. La dynamique partiellement asynchrone est nettement plus difficile et est laissée de côté pour de futurs travaux.

Concernant la topologie, les automates cellulaires ont été principalement étudiés sur la grille \mathbb{Z}^d . C'est ce graphe que nous voulons changer. Pour cette première étude nous nous concentrons sur des graphes simples. Nous comparons les cycles (section 7.2.1), les graphes complets (section 7.2.2), et les arbres (section 7.3). Ce sont les arbres qui constituent la principale difficulté.

Il reste à choisir un ou deux observables. Comme toute chaîne de Markov sur un ensemble fini de configurations, toute trajectoire du modèle étudié ici atteint un ensemble limite, c'est-à-dire une composante fortement connexe du graphe de transitions possibles entre configurations. Les paramètres étudiés sont la structure de l'ensemble limite et le nombre d'étapes pour atteindre cet ensemble limite. Ce nombre d'étapes peut être vu comme le temps que le système passe dans le régime transitoire.

Il s'agit d'un travail en commun avec Damien Regnault et Éric Thierry, non encore publié.

7.1 Le modèle : définitions

7.1.1 Topologie

Comme nous perturbons le graphe, le voisinage n'est plus nécessairement uniforme (il peut changer entre deux cellules). Il nous faut donc introduire de nouvelles notations.

Soit $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ un graphe fini non orienté. \mathcal{U} est l'ensemble des sommets (ou cellules dans la terminologie des automates) et \mathcal{E} l'ensemble des arêtes. $n := |\mathcal{U}|$ est le nombre de cellules comme aux chapitres précédents. Le *voisinage* \mathcal{N}_x d'une cellule x est l'ensemble des sommets y tels qu'il existe une arête (x, y) dans \mathcal{E} , plus la cellule x . Il n'y a pas d'ordre sur les voisines, mais la cellule x est distinguée.

L'ensemble des états est $Q := \{0, 1\}$ (0 sera représenté par du blanc et 1 par du noir). Une *configuration* est une fonction $c : \mathcal{U} \rightarrow Q$.

Une règle δ est une fonction qui à l'ensemble des états des voisines d'une cellule associe un nouvel état. Nous utilisons la **règle minorité**, qui associe l'état le moins présent parmi les états des voisines (pas de changement d'état en cas d'égalité) :

$$c'(x) = \begin{cases} 1 & \text{si } \sum_{y \in \mathcal{N}_x} c(y) < \frac{|\mathcal{N}_x|}{2} & \text{ou} & \sum_{y \in \mathcal{N}_x} c(y) = \frac{|\mathcal{N}_x|}{2} \text{ et } c(x) = 1 \\ 0 & \text{si } \sum_{y \in \mathcal{N}_x} c(y) > \frac{|\mathcal{N}_x|}{2} & \text{ou} & \sum_{y \in \mathcal{N}_x} c(y) = \frac{|\mathcal{N}_x|}{2} \text{ et } c(x) = 0 \end{cases}$$

Dans une configuration, une cellule est **active** si la mettre à jour ferait changer son état. Nous utilisons avec cette règle la dynamique totalement asynchrone.

Dans l'évolution ainsi définie, un ensemble de configurations A est un **attracteur** si pour tout $c, c' \in A$, le temps pour atteindre c' en partant de c est presque sûrement fini. Dans le graphe de transition où les sommets sont les configurations et les arcs sont les couples (c, c') tels que $\mathbb{P}(\delta(c) = c') > 0$, les attracteurs sont donc les composantes fortement connexes. L'union de tous les attracteurs est l'**ensemble limite**, noté \mathcal{A} .

On dit que le système **converge** depuis la configuration initiale $c(\cdot, 0)$ vers un attracteur A (resp. vers l'ensemble limite \mathcal{A}) si la variable aléatoire $T := \min \{t \mid c(\cdot, t) \in A\}$ (resp. $T = \min \{t \mid c(\cdot, t) \in \mathcal{A}\}$) est presque sûrement finie. La variable T est appelée **temps transitoire**. Puisque nous ne considérons que des graphes finis, il y a toujours convergence.

7.1.2 Énergie et Particules

Comme dans le modèle d'Ising [McCoy and Wu (1973)] ou dans les réseaux de Hopfield [Rojas (1996)], nous définissons un paramètre global, que l'on peut voir comme une énergie car il somme les interactions entre cellules voisines. Ce paramètre simple est crucial dans l'étude de ce modèle.

Le **potentiel** v_x d'une cellule x est le nombre de ses voisines (y compris elle-même) dans le même état qu'elle. Par définition, si $v_x \leq \frac{|\mathcal{N}_x|}{2}$ alors la cellule est dans l'état minoritaire et donc inactive. Si au contraire $v_x > \frac{|\mathcal{N}_x|}{2}$ alors la cellule est active. Une configuration c est **stable** si et seulement si $\forall x \in \mathcal{U}, v_x \leq \frac{|\mathcal{N}_x|}{2}$.

L'**énergie** d'une configuration c est $E(c) := \sum_{x \in \mathcal{U}} (v_x - 1)$. L'énergie d'une configuration est toujours positive. Il existe des configurations d'énergie 0 si et seulement si \mathcal{G} est biparti. Ces configurations sont les 2-colorations de \mathcal{G} . Plus précisément, les bornes suivantes sont une conséquence directe des définitions :

Proposition 27 (Bornes sur l'énergie). *L'énergie E satisfait $2|\mathcal{E}| - 2C_{\max} \leq E \leq 2|\mathcal{E}|$, où C_{\max} est le nombre d'arêtes maximum d'une coupe de \mathcal{G} .*

Rappelons qu'une coupe est une partition de \mathcal{U} en \mathcal{U}_1 et \mathcal{U}_2 , le poids d'une coupe est $\#\{xy \mid x \in \mathcal{U}_1 \text{ et } y \in \mathcal{U}_2 \text{ et } xy \in \mathcal{E}\}$

Lemme 28. *L'énergie est une fonction décroissante du temps, et décroît strictement chaque fois qu'une cellule x de potentiel $v_x > \frac{|\mathcal{N}_x|}{2}$ est mise à jour.*

Démonstration. Quand une cellule x de potentiel v_x est mise à jour, son potentiel devient $|\mathcal{N}_x| - v_x + 1$, le potentiel de $v_x - 1$ cellules décroît de 1, et le potentiel de $|\mathcal{N}_x| - 1 - (v_x - 1)$ cellules augmente de 1. La nouvelle énergie est donc $E + 2|\mathcal{N}_x| - 4v_x + 2$. ■

Corollaire 29. Une configuration c appartient à l'ensemble limite si et seulement si il n'y a pas de suite de mises à jour qui fasse décroître l'énergie, i.e. si et seulement si $\forall t \mathbb{P}(E(c(\cdot, t)) > E(c(\cdot, t+1)) \mid c(\cdot, 0) = c) = 0$.

Démonstration. Si l'énergie décroît en mettant à jour c pour obtenir c' , alors c ne peut plus être atteinte (car l'énergie est décroissante). Réciproquement, toute séquence de mises à jour qui garde l'énergie constante est réversible : la cellule mise à jour peut être à nouveau mise à jour pour revenir à la configuration précédente. ■

Puisque mettre à jour une cellule de degré impair fait décroître l'énergie, ces cellules sont inactives dans l'ensemble limite.

Soit c une configuration sur $\mathcal{G} = (\mathcal{U}, \mathcal{E})$; une arête $\{x, y\}$ contient une *particule* si $c(x) = c(y)$. Une configuration est déterminée (à un échange des deux états de Q près) par l'ensemble des positions de ses particules $\mathcal{P} \subseteq \mathcal{E}$. Il est à noter que la proposition réciproque « Tout sous-ensemble $\mathcal{P} \subseteq \mathcal{E}$ correspond à une configuration. » n'est vraie que sur les arbres.

L'énergie d'une configuration est clairement le double de son nombre de particules. Avec le point de vue des particules, lorsque l'on met à jour une cellule x de degré $\deg(x) := |\mathcal{N}_x| - 1$:

- Si le nombre d'arcs adjacents contenant une particule est au moins $\frac{\deg(x)}{2}$, ces particules disparaissent et de nouvelles particules apparaissent sur les arcs adjacents qui n'avaient pas de particules. Cela est illustré sur la figure 7.1).
- Sinon les particules ne bougent pas.

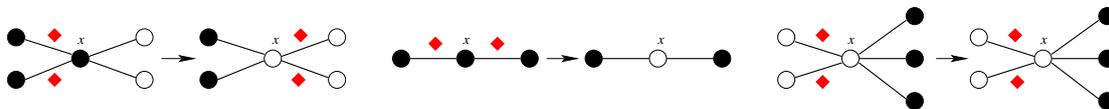


FIG. 7.1 – Transferts de particules (losanges rouges) lors de la mise à jour de la cellule centrale x .

Le point de vue des particules peut simplifier la description des configurations et des orbites, par exemple quand l'énergie est basse et que l'orbite se résume à une marche aléatoire de quelques particules.

7.2 Graphes réguliers

La règle minorité asynchrone a déjà été étudiée sur deux graphes réguliers, précisément deux voisinages différents sur la grille $(\mathbb{Z}/n\mathbb{Z})^2$: le voisinage de Moore (les 8 plus proches voisins) dans [Regnault, Schabanel, and Thierry (2007)], le voisinage de von Neumann (les 4 plus proches voisins) dans [Regnault, Schabanel, and Thierry (2008)]. Dans les deux cas, le temps d'atteinte moyen de l'ensemble limite a été conjecturé polynomial en n . Nous étudions ici deux autres graphes réguliers bien plus simples : les cycles et les graphes complets.

7.2.1 Cycles

Le *cycle* à n sommets est le graphe où l'ensemble des sommets est $\mathcal{U} := \llbracket 1, n \rrbracket$ et l'ensemble des arêtes est $\mathcal{E} := \{(x, x+1 \bmod n) \mid x \in \mathcal{U}\}$.

Théorème 30. *En dynamique totalement asynchrone sur un cycle, l'espérance du temps transitoire est $\Theta(n^3)$. Si n est pair, l'ensemble limite est constitué de deux configurations stables. Si n est impair, l'ensemble limite est l'ensemble des configurations avec une seule particule (configurations d'énergie 2).*

Démonstration. Considérons une configuration finie de taille n avec n pair. La règle minorité signifie ici : « une cellule est active si au moins une de ses deux voisines est dans le même état qu'elle ». Les configurations stables sont donc $(01)^{n/2}$ et $(10)^{n/2}$. Pour une configuration c , définissons la configuration duale $\hat{c} := (10)^{n/2} \oplus c$ (on inverse l'état d'une cellule sur deux). La règle de transition sur la configuration duale devient alors : « une cellule est active si au moins une de ses deux voisines est dans un état différent du sien ». Il s'agit de la règle de l'automate élémentaire 178. En utilisant la preuve de [Fatès, Morvan, Schabanel, and Thierry (2006)], on obtient que l'espérance du temps transitoire est $\Theta(n^3)$.

Si n est impair, alors il n'existe pas de configuration stable. Mais, tant que l'énergie est supérieure à 2, il existe une séquence de mises à jour la faisant décroître. En réutilisant la fonction potentiel de [Fatès, Morvan, Schabanel, and Thierry (2006)] et en la transcrivant dans le primal, il est possible de prouver que l'espérance du temps pour atteindre une telle configuration est $\Theta(n^3)$. ■

7.2.2 Graphe complet

Le **graphe complet** à n sommets est le graphe tel que $\forall x, y \in \mathcal{U}$, $(x, y) \in \mathcal{E}$. Le voisinage d'une cellule est donc \mathcal{U} et toutes les cellules ont le même voisinage.

Toutes les cellules noires (resp. blanches) ont le même potentiel. Si l'on note n_b le nombre de cellules noires, l'énergie est $n_b(n_b - 1) + (n - n_b)(n - n_b - 1)$.

Théorème 31. *En dynamique totalement asynchrone sur un graphe complet, l'espérance du temps transitoire est $\Theta(n)$. Si n est pair, l'ensemble limite est constitué des $C_n^{n/2}$ configurations stables d'énergie $\frac{n^2}{2} - n$, où la moitié des cellules est noire et l'autre blanche. Si n est impair, l'ensemble limite est un unique attracteur constitué des $2C_n^{\frac{n-1}{2}}$ configurations d'énergie $\frac{(n-1)^2}{2}$, où la différence entre le nombre de cellules noires et le nombre de cellules blanches est égale à 1 ou à -1 .*

Démonstration. Établissons tout d'abord la structure des attracteurs. Nous notons toujours n_b le nombre de cellules noires. Comme le voisinage d'une cellule est \mathcal{U} , le potentiel d'une cellule noire est n_b . Le potentiel d'une cellule blanche est $n - n_b$. Si $n_b > \frac{n+1}{2}$ (resp. $n_b < \frac{n-1}{2}$) alors mettre à jour une cellule noire (resp. blanche) fait décroître l'énergie et la configuration n'est pas dans \mathcal{A} . Considérons une configuration de \mathcal{A} , on a donc $\frac{n-1}{2} \leq n_b \leq \frac{n+1}{2}$.

- Si n est pair alors toutes les cellules ont un potentiel $\frac{n}{2}$ et les $C_n^{\frac{n}{2}}$ configurations possibles sont stables.
- Si n est impair, notons C_b (resp. C_w) l'ensemble des configurations où $n_b = \frac{n+1}{2}$ (resp. $n_b = \frac{n-1}{2}$). L'ensemble des configurations d'énergie minimale est $C_w \cup C_b = \mathcal{A}$. Montrons que \mathcal{A} est constitué d'un seul attracteur. Les cellules blanches (resp. noires) des configurations de C_b (resp. C_w) sont inactives et les cellules noires (resp. blanches) sont actives, mettre l'une de ces cellules actives à jour amène à une configuration de C_w (resp. C_b). Toute configuration de C_w peut devenir une configuration de C_b après une mise à jour et vice versa.

Soit $d(c, c') := |\{x \mid c(x) \neq c'(x)\}|$ la distance de Hamming entre deux configurations c et c' . C'est bien une distance, en particulier elle est nulle seulement entre des configurations identiques. Considérons deux configurations distinctes c et c' de C_b . Puisque c et c' ont le même nombre de cellules noires, il existe x et y tels que $c(x) = 1$, $c'(x) = 0$, $c(y) = 0$ et $c'(y) = 1$. Mettre à jour les cellules $c(x)$ puis $c(y)$ dans la configuration c amène à une configuration c'' telle que $d(c'', c') = d(c, c') - 2$. Donc pour toute configurations $c, c' \in C_b \cup C_w$, il existe une séquence de mises à jour qui mène de c à c' . L'ensemble $C_b \cup C_w$ est un attracteur de taille $2C_n^{\frac{n-1}{2}}$.

Pour établir le temps transitoire, considérons une orbite $(c^t)_{t \geq 0}$ et notons n_b^t le nombre de cellules noires au temps t . On peut supposer par symétrie que $n_b^0 \geq \frac{n}{2}$, i.e. qu'il y a au début au moins autant de cellules noires que de cellules blanches. Tant que $n_b^t > \frac{n+1}{2}$, l'automate n'a pas atteint l'ensemble limite et les cellules noires sont actives, tandis que les blanches sont inactives. n_b^t est décrémenté exactement quand une cellule active est mise à jour (le système est donc équivalent au problème du collecteur de coupons), les pires configurations initiales sont donc tout blanc ou tout noir. Dans le pire cas, il suffit que seulement $n/2$ cellules se mettent à jour pour atteindre l'ensemble limite \mathcal{A} . Or à chaque temps t , il y a une probabilité $\frac{n_b^t}{n}$ de mettre à jour une cellule noire. Donc $\mathbb{E}(T) = \sum_{i=0}^{n/2} \frac{n}{n-i}$ et donc $n \leq \mathbb{E}(T) \leq 2n$. L'espérance du pire temps pour atteindre \mathcal{A} est ainsi $\mathbb{E}(T) = \Theta(n)$. ■

Nous avons en passant montré que dans le cas n impair, l'attracteur est un graphe biparti entre C_b et C_w .

7.3 Arbres

Après avoir étudié deux topologies simples dans le but de faire des comparaisons, voici maintenant la topologie principalement étudiée dans ce chapitre : les arbres. Ce sera la topologie utilisée dans tout le reste de ce chapitre.

En tant que graphes bipartis, les arbres ont exactement deux configurations d'énergie 0 (qui est l'énergie minimale), ce sont les deux 2-colorations du graphe. Ce fait est lié à une structure plus profonde de la règle minorité. L'étude de cette règle est facilitée si l'on inverse l'état d'une cellule sur deux, on obtient alors une configuration et une orbite duales, que nous présentons maintenant.

7.3.1 La règle duale

Les règles et orbites duales étant équivalentes à celles de la règle minorité, elles sont introduites seulement pour faciliter l'étude.

Définition 32 (Configuration duale). Étant donné un arbre \mathcal{T} et une racine $r \in \mathcal{T}$, notons $h(x)$ la hauteur d'une cellule x ; c'est la longueur du plus court chemin entre r et x . Pour toute configuration c sur \mathcal{T} , la configuration duale est la configuration \hat{c} définie par

$$\hat{c}(x) := \begin{cases} c(x) & \text{si } h(x) \text{ est pair} \\ 1 - c(x) & \text{si } h(x) \text{ est impair} \end{cases}$$

(voir figure 7.2 page suivante). La fonction $c \mapsto \hat{c}$ est une bijection sur l'ensemble des configurations. En effet, $\hat{\hat{c}} = c$.

Une définition équivalente consiste à faire un XOR avec la 2-coloration de \mathcal{T} où r est blanche. Les configurations duales des configurations d'énergie minimale sont les deux configurations où toutes les cellules sont dans le même état. Dans la configuration duale, les particules sont situées sur les arêtes entre deux cellules de couleurs opposées. Les régions composées de cellules blanches (resp. noires) sont stables, elles vont donc aider à la compréhension d'une orbite dans la suite.

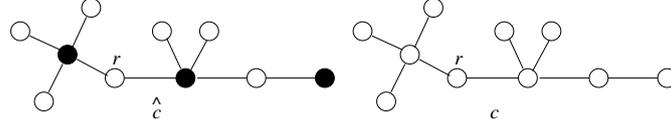


FIG. 7.2 – Une configuration c et sa configuration duale \hat{c} (par rapport à la racine r).

On définit l'énergie dans la configuration duale comme égale à celle de la configuration primale. On peut la définir à partir des états de la configuration duale. Pour cela, soit \hat{v}_x le potentiel dual de la cellule x , défini comme le nombre de ses voisines dans un état différent de celui de x . L'énergie duale est alors la somme des potentiels duaux de toutes les cellules.

Évolution duale

Soit c une configuration et $c' := \delta(c)$. La règle duale $\hat{\delta}$ est la règle qui à \hat{c} associe $\hat{\delta}(\hat{c}) := \widehat{\delta(c)}$. On a donc le diagramme commutatif suivant :

$$\begin{array}{ccc} \text{primal} & & c \xrightarrow{\delta} c' \\ & \uparrow & \uparrow \\ & \downarrow & \downarrow \\ \text{dual} & & \hat{c} \xrightarrow{\hat{\delta}} \hat{c}' \end{array}$$

Fait 33. La règle duale est la règle d'un réseau d'automates en dynamique totalement asynchrone. En effet, $\hat{\delta}$ consiste à choisir une cellule x uniformément au hasard et la mettre à jour selon la règle suivante : « choisir l'état majoritaire parmi les voisines en excluant la cellule mise à jour ; en cas d'égalité changer d'état ». Formellement :

$$\hat{c}'(x) = \begin{cases} 1 & \text{si } \sum_{y \in \mathcal{N}_x \setminus \{x\}} \hat{c}(y) > \frac{|\mathcal{N}_x| - 1}{2} \text{ ou } \hat{c}(x) = 0 \text{ et } \sum_{y \in \mathcal{N}_x \setminus \{x\}} \hat{c}(y) = \frac{|\mathcal{N}_x| - 1}{2} \\ 0 & \text{si } \sum_{y \in \mathcal{N}_x \setminus \{x\}} \hat{c}(y) < \frac{|\mathcal{N}_x| - 1}{2} \text{ ou } \hat{c}(x) = 1 \text{ et } \sum_{y \in \mathcal{N}_x \setminus \{x\}} \hat{c}(y) = \frac{|\mathcal{N}_x| - 1}{2} \end{cases} \quad (7.1)$$

Démonstration. Soit une configuration c et \hat{c} son dual. Une cellule x est active dans c si et seulement si $v_x > \frac{|\mathcal{N}_x|}{2}$. D'après l'équation (7.1), la cellule x est active dans \hat{c} si et seulement si $\hat{v}_x \geq \frac{|\mathcal{N}_x| - 1}{2}$. Or les potentiels sont entiers et $v_x = \hat{v}_x + 1$; les deux conditions sont donc équivalentes. ■

Remarque. La règle duale ainsi définie n'est pas monotone, contrairement aux automates à seuils, dont notamment les réseaux de neurones. En effet, si par exemple une cellule x dans l'état 0 a deux voisines dans l'état 0 et deux dans l'état 1, alors la mettre à jour la fait passer dans l'état 1. Mais dans la nouvelle configuration, mettre à jour la cellule la fait revenir à l'état 0. Nous voyons ainsi deux configurations \hat{c}_0 et \hat{c}_1 telles que $\hat{c}_1 > \hat{c}_0$ pour l'ordre produit, mais où $\hat{\delta}(\hat{c}_1)(x) < \hat{\delta}(\hat{c}_0)(x)$.

En revanche, un argument de monotonie reste possible (et nous en utiliserons un par la suite) si l'on considère toutes les cellules sauf celle mise à jour.

À partir de maintenant nous n'utilisons plus que la règle duale.

Pour alléger les notations, nous ne notons plus les « $\hat{}$ ».

7.4 Structure de l'ensemble limite sur les arbres

Nous caractérisons dans cette section les configurations qui font partie de l'ensemble limite, ce qui permet, pour un arbre donné, de compter le nombre d'attracteurs. Dans le cas particulier des arbres de degré maximum 3, nous verrons à la section 7.5.1 une caractérisation plus simple de l'ensemble limite.

7.4.1 Un algorithme pour l'ensemble limite

Dans cette section nous décrivons un algorithme (algorithme 1) qui détermine si une configuration fait partie de l'ensemble limite et qui sinon donne une suite de mises à jour menant à l'ensemble limite.

Fait 34. *Un attracteur A décompose l'arbre en trois sous-ensembles de cellules :*

1. *les cellules qui sont dans l'état 0 pour toutes les configurations de A ;*
2. *les cellules qui sont dans l'état 1 pour toutes les configurations de A ;*
3. *les cellules qui sont dans l'un ou l'autre état selon la configuration de A . Ces dernières sont dites **oscillantes**.*

Algorithme 1 : Test d'appartenance à l'ensemble limite.

Input : Une configuration c .

1 **while** *Il y a une cellule active x dans l'état 0* **do** Mettre à jour x

2 **while** *Il y a une cellule active x dans l'état 1* **do** Mettre à jour x

3 **while** *Il y a une cellule active x dans l'état 0* **do** Mettre à jour x

Output : c appartient à l'ensemble limite si et seulement si $E(c') = E(c)$.

La terminaison de l'algorithme 1 est claire : la phase 1 accroît le nombre de cellules dans l'état 1 à chaque itération et est donc terminée en $O(n)$ itérations. La même remarque s'applique aux autres phases.

Proposition 35. *La configuration c' à la fin de l'exécution de l'algorithme 1 est dans l'ensemble limite.*

Corollaire 36. *La configuration d'entrée c est dans l'ensemble limite si et seulement si l'énergie n'a pas diminué pendant l'exécution de l'algorithme.*

Preuve de la proposition 35. Montrons d'abord que les cellules dans l'état 0 à la fin de la phase 1 ne peuvent plus passer dans l'état 1 quelle que soit la suite de mises à jour, donc qu'elles sont dans le cas 1 du fait 34. En effet, supposons par l'absurde qu'il existe une cellule x et une suite de configurations c_1, c_2, \dots, c_k telles que

- c_1 est la configuration à la fin de la phase 1 ;
- $c_1(x) = 0$ et $c_k(x) = 1$;

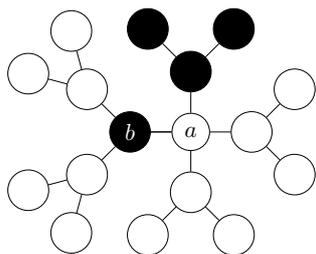
- chaque configuration est le résultat de la mise à jour d'une cellule dans la configuration précédente.

Soit c_ℓ la configuration de cette suite juste avant la première mise à jour d'une cellule dans l'état 0 (une telle cellule existe car au moins x est mise à jour). Soit y cette cellule. Alors y devait déjà être active dans c_1 , car y a dans c_1 au moins autant de voisins dans l'état 1 que dans c_ℓ (c'est un argument de monotonie). L'algorithme ne peut donc pas être sorti de la boucle 1 sur la configuration c_1 , ce qui est une contradiction.

De même, les cellules dans l'état 0 à la fin de la phase 3 le sont définitivement. Par symétrie, les cellules dans l'état 1 à la fin de la phase 2 sont dans le cas 2 du fait 34.

Enfin, les cellules restantes étaient dans l'état 0 à la fin de la phase 2, ce qui signifie que l'on peut les faire passer dans l'état 0 par une suite de mises à jour commençant à la configuration c_1 (la configuration à la fin de la phase 1). Par monotonie, on peut les rendre dans l'état 0 par une suite de mises à jour commençant en toute configuration atteignable à partir de c_1 . Ce qui signifie que la configuration dans laquelle toutes les cellules restantes sont dans l'état 0 est toujours atteignable, elle est donc dans l'attracteur. Cette configuration est la configuration courante à la fin de la phase 2. ■

Trois phases sont nécessaires pour classifier les cellules, comme le montre la figure 7.3. En revanche, comme vu dans la preuve précédente, deux phases sont suffisantes pour atteindre l'ensemble limite.



En partant de la configuration ci-contre, la phase 1 de l'algorithme rend la cellule a noire. Puis la phase 2 rend sa voisine de gauche b blanche, ainsi que a .

b est alors définitivement blanche (cas 1 du fait 34 page précédente), mais n'a pas été identifiée ainsi à la fin de la phase 1. Il faut donc une troisième phase pour classifier b correctement avec cet algorithme.

FIG. 7.3 – Trois phases sont nécessaires pour classifier les cellules.

7.4.2 Structure

Nous avons montré à la section 7.4.1 que, dans l'ensemble limite, certains sommets ont un état fixe tandis que d'autres oscillent entre les deux états de Q . Nous allons maintenant étudier plus précisément la structure de l'ensemble limite.

Nous choisissons tout d'abord un nœud r de degré 1 arbitraire et nous enracinons l'arbre en ce nœud, ce qui définit les relations de « père » et « fils ». Pour éviter toute confusion, le degré d'un nœud est son degré dans le graphe de départ : un nœud de degré 3 est donc un nœud ayant deux fils (à condition qu'il soit différent de la racine).

Nous attribuons une étiquette à chaque cellule pour compter le nombre d'attracteurs et la taille de l'ensemble limite. Nous montrerons que le nombre d'attracteurs est le nombre d'étiquetages respectant certaines conditions, que nous appelons étiquetages acceptables. L'ensemble des étiquettes utilisées est $\{(\square, 0), (\square, 1), (\triangleright, 0), (\triangleright, 1)\}$. Pour la première coordonnée, « \triangleright » signifie intuitivement «oscillant» tandis que « \square » signifie «fixe» (ces symboles sont bien sûr inspirés de ceux des appareils audio : play / stop). La seconde coordonnée est appelée l'état *préférentiel* de la cellule.

Définition 37 (étiquetage acceptable). Un étiquetage est dit acceptable quand pour chaque cellule x , si la cellule a pour étiquette

1. (\square, α) alors elle a strictement plus de $\deg(x)/2$ voisines d'étiquette (\square, α) ;
2. (\triangleright, α) alors
 - (a) si son père a une étiquette de la forme (\square, β) , alors $\alpha = 0$ et x a un fils de plus avec l'étiquette $(\cdot, 1 - \beta)$ que de fils avec l'étiquette (\cdot, β) ;
 - (b) sinon, i.e. si le père a une étiquette de la forme (\triangleright, \cdot) , alors x a un fils de plus étiqueté (\cdot, α) que de fils étiquetés $(\cdot, 1 - \alpha)$.

Un étiquetage est donc une fonction $L : \mathcal{U} \rightarrow \{\square; \triangleright\} \times \{0; 1\}$. Noter que seuls les nœuds de degré pair peuvent avoir une étiquette de la forme (\triangleright, \cdot) .

L'apparente asymétrie du cas 2a (qui impose $\alpha = 0$) ne sert qu'à éviter un double compte dans le théorème 39 page suivante, il serait équivalent de définir les étiquetages acceptables avec $\alpha = 1$.

Le théorème 39 page suivante montre qu'un étiquetage acceptable correspond à un attracteur et le théorème 40 page 103 détaille la signification d'un étiquetage, c'est-à-dire la structure d'un attracteur.

Notation. La configuration $\text{snd}(L)$ correspondant à l'étiquetage L est la projection de la seconde coordonnée : $L(x) = (\cdot, \alpha) \iff \text{snd}(L)(x) = \alpha$. («snd» est une abréviation de «second».)

Notons $\text{paint}(L, \alpha)$ la configuration dont tous les sommets sont dans l'état α , sauf ceux étiquetés $(\square, 1 - \alpha)$:

$$\text{paint}(L, \alpha)(x) := \begin{cases} \beta & \text{si } L(x) = (\square, \beta) \\ \alpha & \text{si } L(x) = (\triangleright, \cdot) \end{cases}$$

Pour une configuration c de l'ensemble limite, notons $\text{attr}(c)$ l'attracteur contenant c .

Remarquer que les fonctions snd et paint sont bien définies sur des étiquetages non acceptables ; attr n'est bien sûr définie que sur l'ensemble limite.

Lemme 38. *Pour tout étiquetage acceptable L , les configurations $\text{snd}(L)$, $\text{paint}(L, 0)$ et $\text{paint}(L, 1)$ sont dans l'ensemble limite et de plus dans le même attracteur.*

Démonstration. Considérons la configuration $\text{paint}(L, 0)$. On examine les nœuds de bas en haut (i.e. un nœud est considéré après tous ses fils, on considère la racine en dernier), et l'on met à jour chaque nœud s'il n'est pas dans son état préférentiel. Mettre à jour un nœud le rend dans son état préférentiel, car son étiquette est acceptable. La configuration est alors $\text{snd}(L)$.

Partant de $\text{snd}(L)$, on considère alors les nœuds de haut en bas et on met à jour ceux qui sont dans l'état 0. Grâce à cet ordre de parcours, quand on considère un nœud, les fils sont dans leur état préférentiel. De plus, si le père d'un nœud a pour étiquette (\triangleright, \cdot) alors ce père est dans l'état 1. À nouveau, d'après la définition d'un étiquetage acceptable, mettre à jour un nœud le rend dans l'état 1. On obtient la configuration $\text{paint}(L, 1)$.

Par symétrie, il existe une suite de mises à jour menant de la configuration $\text{paint}(L, 1)$ à la configuration $\text{paint}(L, 0)$. De plus, il n'y a pas de suite de mises à jour menant de l'une de ces configurations au changement d'état d'une cellule étiquetée $(\square, 1)$: le premier changement d'état d'une telle cellule x serait en contradiction avec le fait qu'elle a plus de $\deg(x)/2$ voisines étiquetées (\square, α) et donc dans l'état α . Quelle que soit la configuration atteinte par des mises à jour depuis $\text{paint}(L, 0)$, il existe donc par monotonie une suite de mises à jours menant à $\text{paint}(L, 1)$. Tout le cycle est donc dans l'ensemble limite.

Puisqu'il existe des mises à jour allant d'une configuration à l'autre, ces configurations sont dans le même attracteur, qui est $\text{attr} \circ \text{snd}(L)$. ■

Théorème 39. *Étant donné un arbre, les étiquetages acceptables et les attracteurs sont en bijection.*

Démonstration. Nous allons définir une fonction f qui à un attracteur A associe un étiquetage acceptable L . Nous montrerons alors que pour tout étiquetage acceptable L , $f(\text{attr} \circ \text{snd}(L)) = L$ et que pour tout attracteur A , $\text{attr} \circ \text{snd}(f(A)) = A$. Cela montrera que f est une bijection et conclura la preuve.

Pour définir f , soit A un attracteur, nous construisons $L = f(A)$. Pour chaque sommet x qui a le même état α dans toutes les configurations de A , on pose $L(x) := (\square, \alpha)$. En particulier, toutes les feuilles sont maintenant étiquetées.

Nous définissons les étiquettes des nœuds restants par induction, de bas en haut. Noter tout d'abord que chacun des nœuds restant oscille entre les deux états dans l'attracteur, il a donc un degré pair, donc un nombre impair de fils. Pour un nœud x dont tous les fils sont étiquetés,

- si le père est déjà étiqueté (et a donc une étiquette de la forme (\square, \cdot)), on pose $L(x) := (\triangleright, 0)$;
- sinon, soit α l'état en majorité parmi les états préférentiels des fils, on pose $L(x) := (\triangleright, \alpha)$. (Puisque x a un nombre impair de fils, l'état majoritaire est bien défini.)

Montrons que cet étiquetage est acceptable. Par le même argument que la preuve de la proposition 35 page 99, la configuration où tous les sommets oscillants de A sont dans l'état 0 (resp. 1) est dans A . Cette configuration est $\text{paint}(L, 0)$ (resp. $\text{paint}(L, 1)$).

Pour montrer que l'étiquetage est acceptable, on distingue les deux cas de la définition.

1. Soit x un nœud étiqueté $(\square, 0)$. Dans la configuration $\text{paint}(L, 1)$, tous les voisins de x non étiquetés $(\square, 0)$ sont dans l'état 1. Puisque cette configuration est dans l'attracteur A , il y a nécessairement plus de $\deg(x)/2$ voisins dans l'état 0 (et donc étiquetés $(\square, 0)$), sinon mettre à jour x ferait décroître l'énergie (cf. proposition 29 page 95).

De même, par symétrie, un nœud x étiqueté $(\square, 1)$ a plus de $\deg(x)/2$ voisins étiquetés $(\square, 1)$. Le point 1 est donc satisfait.

2. Pour montrer le point 2, nous considérons les nœuds de bas en haut et faisons une preuve par induction.

Soit donc x un nœud étiqueté (\triangleright, α) . Montrons tout d'abord qu'il existe une configuration $c \in A$ telle que tous les nœuds du sous-arbre de racine x sont dans leur état préférentiel, sauf x qui est dans l'état $1 - \alpha$. Pour cela, on part de la configuration $\text{paint}(L, 1 - \alpha)$, puis on considère les nœuds de bas en haut et l'on met à jour ceux qui ne sont pas dans leur état préférentiel. Par induction, l'étiquetage de ces nœuds est acceptable, de plus, d'après la définition d'un étiquetage acceptable cette mise à jour les fait changer d'état, ils passent donc dans leur état préférentiel. Nous sommes ainsi partis d'une configuration de A et avons obtenu une configuration c par des mises à jour, donc c est dans l'ensemble limite.

Cela nous permet de montrer que l'étiquetage de x est acceptable :

- (a) Si le père de x a une étiquette de la forme (\square, β) , alors $L(x)$ a été défini comme $(\triangleright, 0)$. De plus, dans la configuration c , le père de x est dans l'état β . Dans cette configuration, mettre à jour x ne peut pas faire décroître l'énergie, donc x a un fils de plus avec l'étiquette $(\cdot, 1 - \beta)$ que de fils avec l'étiquette (\cdot, β) . Ainsi, le point 2a est satisfait.

- (b) Sinon, le père a une étiquette de la forme (\triangleright, β) . Par définition de L , dans la configuration c , x a une majorité de fils dans l'état α . Le mettre à jour le fait donc changer d'état. Donc x a autant de voisins dans l'état α que dans l'état $1 - \alpha$. En particulier, $\beta = 1 - \alpha$ et x a exactement un fils de plus étiqueté (\cdot, α) que de fils étiquetés $(\cdot, 1 - \alpha)$. Le point 2b est satisfait.

Nous avons donc associé par f un étiquetage acceptable à un attracteur. Réciproquement, le lemme 38 associe l'attracteur $\text{attr} \circ \text{snd}(L')$ à un étiquetage acceptable L' .

Montrons maintenant que pour tout étiquetage acceptable L' , $f(\text{attr} \circ \text{snd}(L')) = L'$. Puisque $\text{paint}(L', 0)$ et $\text{paint}(L', 1)$ sont dans l'attracteur $\text{attr} \circ \text{snd}(L')$ (cf. lemme 38), les nœuds étiquetés (\triangleright, \cdot) dans L' sont oscillants dans cet attracteur. Par définition de f , ces nœuds ont une étiquette de la forme (\triangleright, \cdot) dans $f(\text{attr} \circ \text{snd}(L'))$.

Rappelons qu'il n'y a pas de suite de mises à jour menant de $\text{snd}(L')$ au changement d'état d'une cellule étiquetée (\square, α) dans L' : le premier changement d'état d'une telle cellule x serait en contradiction avec le fait qu'elle a plus de $\deg(x)/2$ voisines étiquetées (\square, α) et donc dans l'état α .

On conclut que les étiquetages $f(\text{attr} \circ \text{snd}(L'))$ et L' ont les mêmes cellules étiquetées $(\square, 0)$ et $(\square, 1)$, ils sont donc égaux. En effet, dans la définition de f , l'ensemble des étiquettes de la forme (\triangleright, \cdot) est totalement déterminé par les étiquettes de la forme (\square, \cdot) .

Soit A' un attracteur, montrons enfin que $\text{attr} \circ \text{snd}(f(A')) = A'$. $f(A')$ est un étiquetage acceptable, d'où (lemme 38 page 101), $\text{paint}(f(A'), 1)$ est dans l'attracteur $\text{attr} \circ \text{snd}(f(A'))$. Par ailleurs, nous avons remarqué ci-dessus (par le même argument que la preuve de la proposition 35 page 99) qu'appartient à A' la configuration dans laquelle tous les sommets oscillants de A' sont dans l'état 1. Par définition de f , cette configuration est $\text{paint}(f(A'), 1)$. Les attracteurs $\text{attr} \circ \text{snd}(f(A'))$ et A' ont en commun l'élément $\text{paint}(f(A'), 1)$, ils sont donc le même attracteur.

En conclusion, pour tout attracteur A , $\text{attr} \circ \text{snd}(f(A)) = A$ et pour tout étiquetage acceptable L , $f(\text{attr} \circ \text{snd}(L)) = L$. Par suite, f est une bijection. ■

Théorème 40 (Structure d'un attracteur). *Soit L un étiquetage acceptable. Pour toute configuration c atteignable par des mises à jour à partir de $\text{snd}(L)$, pour tout nœud x :*

1. Si $L(x) = (\square, \alpha)$ alors $c(x) = \alpha$ (c'est pourquoi « \square » signifie intuitivement «fixe»).
2. Si $L(x) = (\triangleright, \alpha)$ (dans ce cas $\deg(x)$ est pair et x n'est donc pas la racine car cette dernière est de degré 1) alors
 - (a) si le père a une étiquette de la forme (\square, \cdot) , x est dans l'état majoritaire parmi ses voisins (il n'y a pas de contrainte en cas d'égalité);
 - (b) sinon,
 - si x est dans son état préférentiel α , ses fils étiquetés (\cdot, α) sont aussi dans leur état préférentiel α ;
 - sinon, tous ses fils qui ne sont pas dans leur état préférentiel sont dans le même état que x (l'état $1 - \alpha$).

Démonstration. Rappelons qu'il n'y a pas de suite de mises à jour menant de $\text{snd}(L)$ au changement d'état d'une cellule étiquetée (\square, α) : le premier changement d'état d'une

telle cellule x serait en contradiction avec le fait qu'elle a plus de $\deg(x)/2$ voisines étiquetées (\square, α) et donc dans l'état α . Cela prouve le point 1.

Puisque $\text{snd}(L)$ et donc c sont dans l'ensemble limite, l'énergie ne peut décroître lors de la mise à jour d'une cellule. Il suit que toute cellule x a toujours au moins $\deg(x)/2$ voisines dans le même état qu'elle. Cela prouve le point 2a.

Pour prouver le dernier point (2b), on remarque que ces conditions sont clairement vérifiées pour $\text{snd}(L)$. Il suffit donc de montrer récursivement que si les conditions sont vérifiées pour une configuration c , alors elles sont vérifiées dans la configuration c' résultant de la mise à jour d'un nœud. Soit x le nœud mis à jour, il suffit de montrer que le point 2b est vérifié dans c' pour x et ses voisins.

Montrons-le d'abord pour x :

- Si x est dans son état préférentiel α dans c , alors ses fils étiquetés (\cdot, α) , qui sont au nombre de $\deg(x)/2$ d'après la définition d'un étiquetage acceptable, sont aussi dans l'état α dans c et donc dans c' . x ne peut donc changer d'état que si tous les autres fils sont dans l'état $1 - \alpha$. Le point 2b reste alors vérifié pour x dans c' .
- Si x n'est pas dans son état préférentiel, soit il est inactif et le point 2b reste vérifié dans c' , soit il est actif.

Dans ce dernier cas, x a autant de voisins dans chacun des deux états (car c est dans l'ensemble limite), tous les fils de x qui ne sont pas dans leur état préférentiel sont dans le même état que x (par hypothèse de récurrence), enfin x a un fils de plus étiqueté (\cdot, α) que de fils étiquetés $(\cdot, 1 - \alpha)$ (d'après la définition d'un étiquetage acceptable). Ces trois contraintes impliquent que tous les fils de x sont en fait dans leur état préférentiel. Par suite, c' satisfait le point 2b.

Montrons-le enfin pour un voisin w de x :

- Si w est un fils de x , rien n'a changé pour les fils de w et le point 2b est toujours vérifié.
- Si w est le père de x , il suffit de considérer le cas où x est dans le même état α que w (il n'y a pas de conditions à satisfaire pour l'autre cas).
 - Si α est l'état préférentiel de x , alors x a au moins $\deg(x)/2 + 1$ voisins dans l'état α : ses fils étiquetés (\cdot, α) et son père w . Donc x est inactif.
 - Sinon, x comme w ne sont pas dans leur état préférentiel et le point 2b reste vérifié, que x change ou non d'état. ■

7.5 Temps transitoire sur les arbres

Nous étudions dans la dernière section de ce chapitre le temps transitoire, ou temps pour atteindre l'ensemble limite. Nous verrons que le degré maximum 4 constitue un seuil : en dessous, la convergence est en temps polynomial ; pour un degré maximum 4 ou plus, la convergence est exponentielle.

7.5.1 Arbres de degré maximum au plus 3

Rappelons que le degré fait référence au graphe de départ : un sommet de degré trois est un nœud de l'arbre ayant deux fils. Si l'on considère des arbres enracinés, nous étudions donc dans cette section des arbres binaires.

On appelle *chemin* dans un arbre une suite de sommets x_1, \dots, x_k telle que x_1 et x_k sont de degré impair et, pour tout i , x_i et x_{i+1} sont voisins et de degré pair.

Le résultat de cette section est le suivant : sur les arbres de degré maximum au plus 3, le système converge en temps polynomial vers une configuration où les cellules

de degré 1 et 3 sont inactives (rappelons que les nœuds de degré impair sont inactifs dans l'ensemble limite) et où il y a éventuellement des particules isolées évoluant sur des chemins disjoints.

Théorème 41. *La règle minorité en dynamique totalement asynchrone sur les arbres de degré maximum au plus 3 converge en un temps d'espérance $O(n^4)$. Les attracteurs d'un arbre \mathcal{T} sont en bijection avec les couplages de l'arbre réduit \mathcal{T}' où chaque chemin de \mathcal{T} a été remplacé par un arc, puis chaque feuille a été supprimée.*

Pour établir ce théorème, nous avons besoin d'un lemme majorant le temps d'atteinte d'une chaîne de Markov sur un rectangle, pour lequel nous avons besoin de quelques rappels sur les chaînes de Markov.

Rappels sur les chaînes de Markov Nous ne rappelons ici que les éléments de la théorie des chaînes de Markov qui sont utiles à notre étude. Pour une introduction plus progressive et pour les preuves, nous renvoyons au chapitre 7 de [Levin et al. (2006)].

Soit $(X_t)_{t \in \mathbb{N}}$ une chaîne de Markov. On note τ_b le **temps d'atteinte** de b , i.e. la première étape où la chaîne de Markov est dans l'état b :

$$\tau_b := \min \{t \mid X_t = b\}$$

Si (X, P) est une chaîne de Markov réversible pour la probabilité π , la **conductance** d'une arête non orientée (x, y) est

$$g(x, y) := \pi(x)\mathbb{P}(x, y) = \pi(y)\mathbb{P}(y, x)$$

Soient a et b deux sommets (qui joueront le rôle de source et de puits). On définit le **potentiel** d'un sommet par

$$V(x) := \mathbb{P}(\tau_a < \tau_b \mid X_0 = x)$$

Clairement, $V(a) = 1$ et $V(b) = 0$. On définit alors l'**intensité** sur une arête orientée par

$$I(x, y) := g(x, y)(V(x) - V(y)) \quad \text{et} \quad \|I\| := \sum_x I(a, x)$$

La **résistance effective** entre a et b est alors

$$\mathcal{R}(a, b) := \frac{V(a) - V(b)}{\|I\|}$$

Théorème 42. $\mathbb{E}(\tau_b \mid X_0 = a) + \mathbb{E}(\tau_a \mid X_0 = b) = g\mathcal{R}(a, b)$

Dans notre cas $g = 1$, donc $\mathcal{R}(a, b)$ est un majorant du temps d'atteinte $\mathbb{E}(\tau_b \mid X_0 = a)$.

Il nous reste à majorer $\mathcal{R}(a, b)$. Un **flot** de a à b est une fonction sur les arêtes orientées qui est antisymétrique : $\theta(x, y) = -\theta(y, x)$ et qui suit la loi de Kirchhoff sur les nœuds (conservation du flot) :

$$\forall x \notin \{a, b\} \quad \sum_y \theta(x, y) = 0$$

La **valeur** du flot est

$$\|\theta\| := \sum_x \theta(a, x)$$

Théorème 43 (Principe de Thomson). *Pour tout graphe fini connexe,*

$$\mathcal{R}(a, b) = \inf \{ \mathcal{E}(\theta) \mid \theta \text{ un flot de valeur 1 de } a \text{ à } b \}$$

$$\text{où } \mathcal{E}(\theta) := \sum_{x, y} \frac{(\theta(x, y))^2}{g(x, y)}.$$

Application à une chaîne de Markov sur un rectangle Nous considérons donc une chaîne de Markov sur un rectangle de taille (ℓ, m) et voulons majorer le temps d'atteinte du coin (ℓ, m) depuis tout point de départ. Précisément, chaque nœud (i, j) a 4 arêtes sortantes, vers les voisins $(i \pm 1, j)$ et $(i, j \pm 1)$ (si un voisin n'existe pas car le nœud est au bord, l'arête pointe sur (i, j)). Chaque arête a la même probabilité $\frac{1}{4}$.

Une probabilité invariante pour cette chaîne de Markov est la probabilité uniforme $\pi : x \mapsto \frac{1}{nm}$. Notre chaîne de Markov est réversible pour cette probabilité. Il existe donc une conductance

$$\forall x, y \quad g(x, y) = \frac{1}{4\ell m}$$

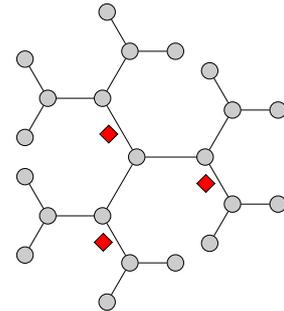
Grâce au principe de Thomson, il suffit de construire un flot de a à b pour obtenir un majorant de $\mathcal{R}(a, b)$. Si $a = (i, j)$ et $b = (\ell, m)$, on considère le flot (largement non optimal) qui consiste en un simple chemin de a à b :

$$\begin{cases} \theta((k, j), (k + 1, j)) := 1 & \text{si } i \leq k < \ell \\ \theta((\ell, k), (\ell, k + 1)) := 1 & \text{si } j \leq k < m \\ \theta(x, y) := 0 & \text{sinon} \end{cases}$$

$\mathcal{E}(\theta) \leq (\ell + m)4\ell m$. On conclut grâce au théorème 42 page précédente que le temps moyen d'atteinte est $O(n^3)$ car ℓ et m sont majorés par le nombre de cellules n .

Preuve du théorème 41. Étudions le mouvement des particules sur l'arbre initial \mathcal{T} . Soit c une configuration de \mathcal{T} appartenant à l'ensemble limite. Il ne peut pas y avoir deux particules sur le même chemin, sinon il existerait une suite de mises à jour qui conduirait à la collision et l'annihilation de ces deux particules, et donc à une diminution de l'énergie. De même, il ne peut pas y avoir deux particules sur deux chemins adjacents \mathcal{P}_1 et \mathcal{P}_2 en x , sinon il existerait une suite de mises à jour positionnant les deux particules sur les arcs de \mathcal{P}_1 et \mathcal{P}_2 adjacents à x , et la mise à jour de x ferait alors disparaître au moins deux particules en en créant au plus une, et ferait donc décroître l'énergie. Enfin, il ne peut y avoir de particule sur un chemin ayant une extrémité x de degré 1, car il existerait une suite de mises à jour menant la particule sur l'arc adjacent à x , mettre à jour x ferait alors disparaître la particule.

Réciproquement, si une configuration ne contient pas deux particules sur un même chemin, ni deux particules sur des chemins adjacents, ni de particule sur un chemin ayant une feuille pour extrémité, alors le nombre de particules sur chaque chemin (0 ou 1) est constant et l'énergie est constante. La configuration appartient donc à l'ensemble limite. Cela établit la bijection avec les couplages de \mathcal{T}' (les arêtes du couplage sont les positions des particules dans \mathcal{T}). Une illustration est donnée ci-contre (le gris est utilisé pour ne pas spécifier l'état noir ou blanc d'une cellule ; les particules sont en rouge à côté des arêtes).

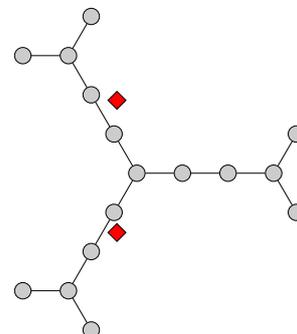


Pour prouver la borne sur l'espérance du temps transitoire, nous établissons une borne sur l'espérance du nombre d'étapes avant qu'une particule ne disparaisse.

Considérons tout d'abord une configuration où il existe au moins deux particules sur le même chemin. Soit ℓ la longueur de ce chemin. Ces deux particules suivent une marche aléatoire pour laquelle on peut supposer qu'il y a une réflexion au bord (si la particule n'est pas réfléchie, c'est qu'une cellule de degré 3 a été mise à jour, donc que l'énergie a diminué) et qu'il n'y a pas de collision avec d'autre particule (sinon à nouveau l'énergie décroît). Une borne sur l'espérance du nombre d'étapes avant la collision de ces deux

particules découle alors d'une analyse classique des marches aléatoires avec réflexion au bord [Grimmett and Stirzaker (2001)] : cette espérance est majorée par $O(\ell^3)$. Cette valeur est donc une borne sur l'espérance du temps avant la disparition d'au moins une particule.

Considérons maintenant une configuration où deux chemins adjacents en x ont chacun une particule, comme illustré ci-contre. Soient ℓ et m la longueur des chemins. Les deux particules suivent une marche aléatoire pour laquelle on peut de même supposer qu'il y a réflexion aux bords différents de x . Le seul moyen pour une de ces particules de disparaître est donc qu'elles soient toutes les deux sur une arête adjacente à x , puis que x soit mise à jour. Les distances respectives de ces deux particules à la cellule x forment une marche aléatoire à deux dimensions, que nous avons analysée à la section 7.5.1.



Cette analyse montre que les deux particules sont adjacentes en même temps à x après un temps d'espérance au plus $O(\max(\ell, m)^3)$. Il y a alors une probabilité $\frac{1}{3}$ de faire disparaître les deux particules plutôt que de les écarter de x .

En conclusion, pour toute configuration n'appartenant pas à l'ensemble limite, au moins une particule disparaît en un temps d'espérance $O(n^3)$. Puisqu'il y a $O(n)$ particules au départ, l'espérance du temps pour atteindre l'ensemble limite est $O(n^4)$. ■

7.5.2 Arbres de degré maximum 4 ou plus

Nous montrons dans cette section que lorsque le degré n'est plus majoré par 3, la convergence vers l'ensemble limite peut prendre un temps exponentiel. Pour cela, nous considérons un arbre et une configuration initiale particuliers et montrons qu'il faut en moyenne $\Omega(na^n)$ étapes, pour une constante $a > 1$, avant d'atteindre l'ensemble limite.

Cet arbre est construit à partir de *gadgets* qui en sont les briques : la *tête*, un nombre variable de *portes*, et la *queue*. Ils sont définis dans la figure 7.4 page suivante. Une porte pourra être au cours de l'orbite dans trois configurations, qui sont définies sur la figure 7.5. Une porte instable oscille aléatoirement entre les configurations verrouillée et déverrouillée. Nous définissons la configuration initiale telle que toutes les portes sont dans dans la configuration instable verrouillée. Il y a donc une particule entre la tête et la porte 1.

Intuitivement, l'orbite avant d'atteindre l'ensemble limite est alors la suivante. Cette particule peut franchir une porte vers la droite (et la rendre stable) seulement si elle est déverrouillée. Elle peut toujours franchir une porte vers la gauche (et la rendre instable)¹. La particule suit donc une marche aléatoire biaisée vers la gauche. Comme l'attracteur est atteint quand la particule a franchi toutes les portes vers la droite, il n'est atteint qu'après un temps exponentiel.

Lemme 44 (Marche aléatoire biaisée). *Soient $0 < a < b < \frac{1}{2}$ et $(X_u)_{u \geq 0}$ une marche aléatoire sur $\llbracket 0, m \rrbracket$ telle que pour tout $u \geq 0$:*

- $\mathbb{P}(X_{u+1} = 1 \mid X_u = 0) = 1$ (barrière réfléchissante en 0),
- $\mathbb{P}(X_{u+1} = m \mid X_u = m) = 1$ (barrière absorbante en m),
- pour tout $0 < i < m$,

$$p_i := \mathbb{P}(X_{u+1} = i + 1 \mid X_u = i) \in [a; b]$$

¹Une image de ce processus est une fermeture éclair : la particule est le chariot et les portes sont les dents.

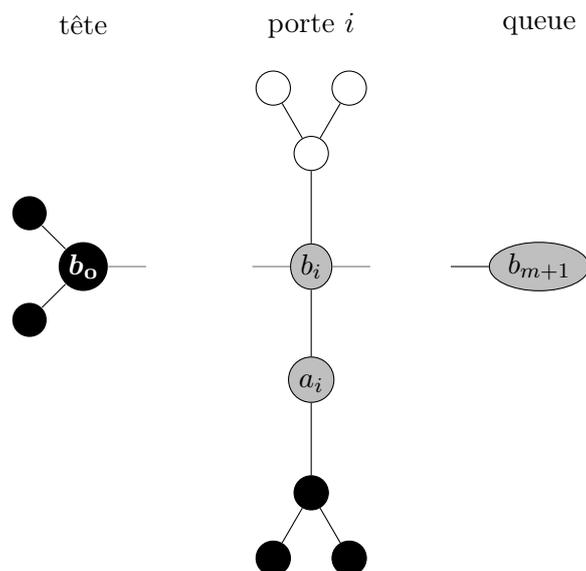


FIG. 7.4 – Les trois gadgets utilisés dans la construction des arbres biaisés. Les cellules grisées peuvent changer d'état entre la configuration initiale et la configuration finale. Les arêtes ayant une seule extrémité serviront à assembler des gadgets pour former un arbre biaisé (cf. figure 7.6).

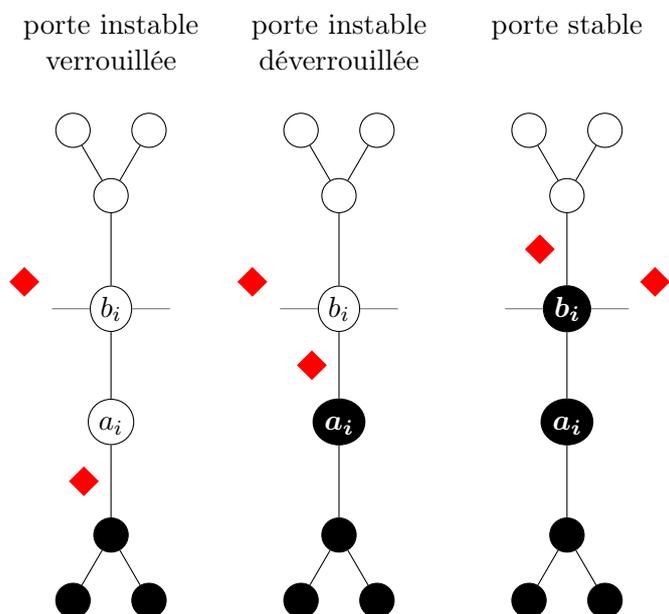


FIG. 7.5 – Les trois configurations d'une porte. Chaque porte contient exactement une particule, sur un lien vertical. Une particule sur un lien horizontal peut passer la porte vers la droite si et seulement si elle n'est pas verrouillée. Le pont b_i change alors de couleur.

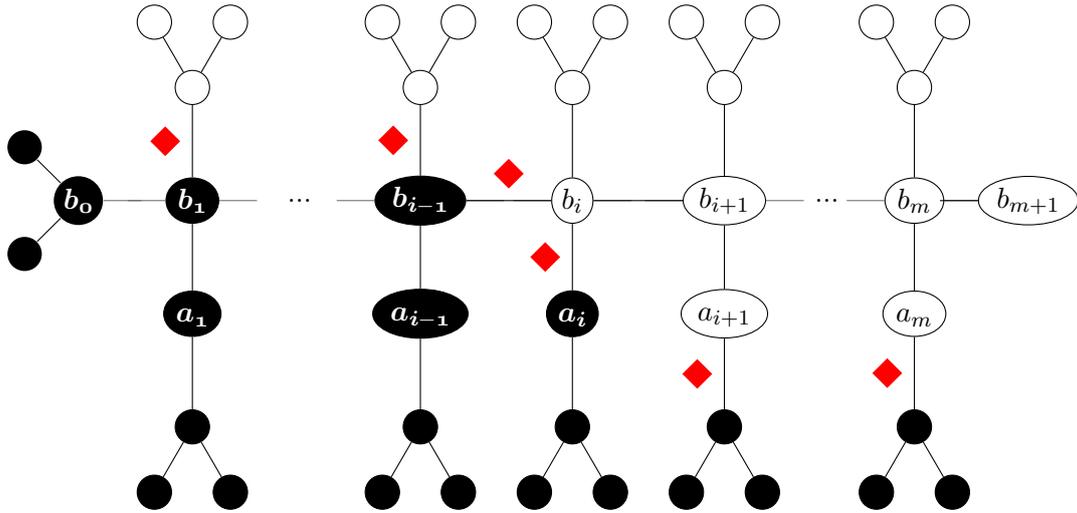


FIG. 7.6 – Un arbre biaisé et une configuration correcte déverrouillée en position $i - 1$. Les portes à gauche de la particule sont stables. Les portes à droite sont instables et peuvent être verrouillées ou déverrouillées. Exactement un lien horizontal porte une particule.

et $\mathbb{P}(X_{u+1} = i - 1 | X_u = i) + \mathbb{P}(X_{u+1} = i + 1 | X_u = i) = 1$
 Soit $T := \min \{u \mid X_u = m\}$.

Alors $\theta(b) \leq \mathbb{E}(T | X_0 = 0) \leq \theta(a)$ où $\theta(p) = \frac{2p(1-p)}{(1-2p)^2} \left(\left(\frac{1-p}{p} \right)^m - 1 \right) - \frac{m}{1-2p}$.

Démonstration. Soit $T_i := \mathbb{E}(T | X_0 = i)$. On a :

$$\begin{cases} T_0 = 1 + T_1 \\ T_m = 0 \\ \forall 0 < i < m \quad T_i = 1 + p_i T_{i+1} + (1 - p_i) T_{i-1} \end{cases} \quad (7.2)$$

Considérons d'abord le cas où tous les p_i sont égaux à p . On vérifie alors qu'une solution de (7.2) est $T_k = \theta_k(p) := \frac{2p(1-p)}{(1-2p)^2} \left(\left(\frac{1-p}{p} \right)^m - \left(\frac{1-p}{p} \right)^k \right) - \frac{m-k}{1-2p}$. Un argument de couplage entre ce cas particulier et le cas général permet d'affirmer que, dans le cas général, $\theta_k(b) \leq T_k \leq \theta_k(a)$ pour tout k . Ce qui implique le résultat en considérant $k = 0$. ■

Nous définissons l'**arbre biaisé** de taille m (cf. figure 7.6) comme l'arbre constitué d'une famille de m portes $(W_i)_{i \in m}$, de W_0 un gadget tête et de W_{m+1} un gadget queue. Les arêtes de cet arbre sont les arêtes des gadgets plus $\bigcup_{i=0}^m b_i b_{i+1}$.

Définition 45 (Configuration correcte). Une configuration c est correcte si les cellules de sa tête sont noires, si celle de sa queue est blanche, et s'il existe j tel que pour tout $1 \leq i \leq j$ la porte i est stable et pour tout $j < k \leq m$ la porte k est instable. La configuration c est alors dite en **position** j , ce qui est noté $Pos(c) = j$. La position est déverrouillée si $j = m$ ou si la porte $j + 1$ est déverrouillée, sinon la position est verrouillée.

Fait 46 (Cellule active). Considérons la configuration correcte c en position j . Les cellules actives de c sont :

- la cellule b_j si $j \neq 0$,
- la cellule b_{j+1} si $j \neq m$ et si la porte W_{j+1} est déverrouillée.
- la cellule a_i si $j < i \leq m$.
- la cellule b_{m+1} si $j = m$.

Démonstration. Considérons une configuration correcte c en position j . Les seules cellules qui peuvent être actives sont les cellules b_i et a_i pour $1 \leq i \leq m$ et la cellule b_{m+1} . La cellule b_{m+1} est active si et seulement si $c(b_m) = 1$ c'est-à-dire que la porte W_m est stable. Pour tout $1 \leq i \leq m$, la cellule a_i est active si et seulement si W_i est instable c'est-à-dire $j < i \leq m$. Pour tout $1 \leq i \leq m$, la cellule b_i est inactive si $c(b_{i-1}) = c(b_i) = c(b_{i+1})$. Donc parmi les cellules $(b_i)_{1 \leq i \leq m}$, seules b_j et b_{j+1} peuvent être actives : la cellule b_j est active et la cellule b_{j+1} est active si la porte W_{j+1} est déverrouillée. ■

Définition 47. La *configuration finale* c_f est la configuration où les cellules de la tête sont noires, la queue est noire et toutes les portes sont stables. On pose $Pos(c_f) := m+1$.

Lemme 48. *La configuration c_f est stable.*

Démonstration. Considérons la configuration correcte c en position m . Selon le fait 46 page précédente, seules les cellules b_m et b_{m+1} sont actives. Si la cellule b_{m+1} se met à jour, ces deux cellules deviennent inactives et aucune autre cellule ne devient active. Mettre à jour la cellule b_{m+1} mène à la configuration c_f . Donc c_f est stable. ■

Corollaire 49. *Considérons une configuration correcte c , alors la configuration $c' = \delta(c)$ est soit correcte, soit c_f . De plus, $|Pos(c') - Pos(c)| \leq 1$.*

Démonstration. Considérons une configuration correcte c en position j et la configuration $c' = \delta(c)$. Si une cellule inactive se met à jour alors $c' = c$. Sinon, distinguons les cas d'après le fait 46.

- Si $j \neq 0$ et la cellule b_j se met à jour, alors la porte W_j devient déverrouillée et c' est une configuration correcte déverrouillée en position $j - 1$.
- Si $j \neq m$ et la cellule b_{j+1} se met à jour, alors la porte W_{j+1} devient stable et c' est une configuration correcte en position $j + 1$.
- Si la cellule a_i se met à jour avec $j < i \leq m$, alors la porte W_i devient déverrouillée (resp. verrouillée) dans c' si elle était verrouillée (resp. déverrouillée) dans c . La configuration c' reste correcte et en position j .
- Si $j = m$ et la cellule b_{m+1} se met à jour : alors $c' = c_f$. ■

Théorème 50. *Sur les arbres biaisés de taille m (i.e. $n = 8m + 4$ cellules), à partir de la configuration initiale où toutes les portes sont instables, la règle minorité converge presque sûrement vers c_f . De plus, il existe $a, b > 1$ tels que le temps transitoire T vérifie $\Theta(na^n) \leq \mathbb{E}(T) \leq \Theta(nb^n)$.*

Démonstration. Considérons un arbre biaisé de taille m , une configuration initiale correcte c^0 en position 0 et $(c^t)_{t \geq 0}$ l'orbite à partir de cette configuration initiale. Soit $(t_i)_{i \geq 0}$ la suite de variables aléatoires correspondant aux changements de position : $t_0 = 0$ et $t_{i+1} := \min \{t > t_i \mid Pos(c^t) \neq Pos(c^{t_i}) \text{ ou } Pos(c^t) = m + 1\}$. Considérons la suite de variables aléatoires $(X_i)_{i \geq 0}$ telle que $X_i = Pos(c^{t_i})$. D'après le corollaire 49, $|X_{i+1} - X_i| = 1$. Nous allons étudier le temps de convergence du processus X_i , qu'il sera ensuite facile de relier au temps de convergence de la règle minorité.

Pour une configuration c verrouillée en position j , mettre à jour la cellule b_j mène à une configuration en position $j - 1$ tandis que mettre à jour la cellule a_j mène à une configuration déverrouillée en position j . Mettre à jour d'autres cellules ne modifie pas la position de la configuration.

Par ailleurs, pour une configuration c déverrouillée en position $m_0 < j < m$, mettre à jour b_j mène à une configuration en position $j - 1$, mettre à jour a_j mène à une configuration verrouillée en position j et mettre à jour b_{j+1} mène à une configuration en position $j+1$. Mettre à jour d'autres cellules ne modifie pas la position de la configuration.

Une cellule a une probabilité $1/n$ de se mettre à jour où $n = 8m + 4$. Donc l'orbite d'une configuration en position $0 < j < m$ peut être résumée par la figure 7.7. On vérifie

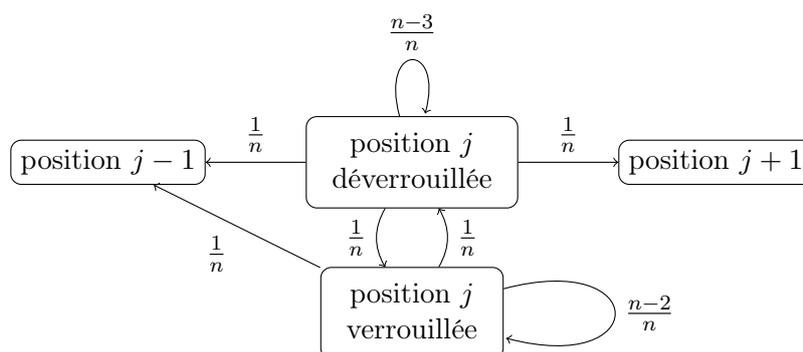


FIG. 7.7 – Évolution d'une position. Les nombres sur les arêtes sont les probabilités de transition. On déduit de ce graphe de chaîne de Markov que la probabilité de passer dans la position $j + 1$ avant la position $j - 1$ à partir de la position j déverrouillée est $\frac{2}{5}$, tandis qu'à partir de la position j verrouillée, la probabilité est $\frac{1}{5}$.

alors que le comportement du processus stochastique $(X_i)_{i \geq 0}$ est celui de la marche aléatoire biaisée du lemme 44 page 107. En effet :

– si $0 < x < m$ alors

$$\mathbb{P}(X_{i+1} = x + 1 | X_i = x) + \mathbb{P}(X_{i+1} = x - 1 | X_i = x) = 1$$

et
$$1/5 \leq \mathbb{P}(x_{i+1} = x + 1 | X_i = x) \leq 2/5$$

– $\mathbb{P}(X_{i+1} = 1 | X_i = 0) = 1$.

Soit $T' := \min\{i \mid X_i = m\}$ la variable aléatoire correspondant au premier temps où toutes les portes sont stables, le lemme 44 nous donne $\Theta\left(\left(\frac{3}{2}\right)^m\right) \leq \mathbb{E}(T') \leq \Theta(4^m)$.

Il nous reste à relier T' et T . Soit c_{f-1} la configuration correcte en position m (*i.e.* toutes les portes sont stables, mais la queue est blanche).

Alors $c^T = c_f$, $c^{T-1} = c_{f-1}$ et $\mathbb{P}(c^{t+1} = c_f \mid c^t = c_{f-1}) = 1/2$

Donc
$$\mathbb{E}(T) = \Theta(\mathbb{E}(t_{T'}))$$

Ce qui nous permet à partir du temps de convergence de X de déduire celui de minorité sur l'arbre biaisé.

Par définition,
$$t_{T'} = \sum_{i=1}^{T'} (t_i - t_{i-1})$$

Puisque qu'il y a au plus deux cellules qui peuvent modifier la position d'une configuration correcte, $\mathbb{E}(t_i - t_{i-1}) = \Theta(n)$.

Donc
$$\mathbb{E}(t_{T'}) = \Theta(n\mathbb{E}(T'))$$

Et finalement,

$$\Theta\left(n\left(\frac{3}{2}\right)^m\right) \leq \mathbb{E}(T) \leq \Theta(n4^m)$$

soit

$$\Theta\left(n\sqrt[4]{\frac{3^n}{2}}\right) \leq \mathbb{E}(T) \leq \Theta\left(n\sqrt{2^n}\right) \quad \blacksquare$$

7.6 Conclusion et perspectives

Ce chapitre a présenté une étude théorique de l'influence de la topologie et en particulier son interaction avec une autre perturbation, l'asynchronisme. Pour cela, nous avons fixé une règle (la règle minorité) et une dynamique asynchrone (la dynamique totalement asynchrone), et avons comparé différentes topologies. Le choix de la règle, de la dynamique et des topologies a été discuté en introduction. Nous avons alors établi la structure de l'ensemble limite et le temps de convergence vers ce dernier pour chaque topologie. Si l'étude des deux premières topologies, très simples, est aisée, elle ne le reste pas dès que l'on change de topologie. En effet, l'étude du temps transitoire comme de la structure de l'ensemble limite se révèle difficile sur les arbres, comme cela avait déjà été observé sur la grille 2D. Ce chapitre est donc un nouvel exemple de problème sur les arbres dont la résolution n'est pas évidente.

La table 7.1 résume les espérances de temps transitoire et les compare à ces temps transitoires pour la dynamique synchrone.

	Totalement asynchrone	Synchrone
Ligne et cycle	Poly	
Arbres	degré max ≤ 3	Exp ¹
	degré max ≥ 4	
Tore, voisinage de Moore ou de von Neumann	Poly ? [Regnault, Schabanel, and Thierry (2007)]	
Graphe complet	Poly	Poly ¹

¹[Goles and Martinez (1990)]

TAB. 7.1 – Temps transitoires en dynamiques synchrone et asynchrone pour la règle minorité sur quelques graphes simples : polynomial (Poly) ou exponentiel (Exp) en le nombre de sommets du graphe, pour la pire configuration initiale.

On observe sur les arbres une transition entre temps de convergence polynomial et exponentiel en fonction du degré maximum. Y a-t-il un paramètre du graphe qui permet de prédire si le temps transitoire est polynomial ou exponentiel ? La question reste ouverte, mais notre étude a déjà écarté le degré maximum et le degré moyen. En remarquant (table 7.2) que le temps de convergence diminue en passant des arbres généraux à la grille 2D, puis de la grille au graphe complet, on peut formuler l'hypothèse qu'ajouter des liens accélère la convergence, sous certaines conditions à préciser.

Par ailleurs, nous avons observé des structures complexes de l'ensemble limite même pour une topologie fréquemment rencontrée et une règle simple. La table 7.2 résume les structures observées.

Perspectives La caractérisation des attracteurs devrait permettre de concevoir un algorithme efficace pour dénombrer le nombre d'attracteurs. De même, la caractérisation des configurations de l'ensemble limite devrait permettre de dénombrer le nombre de configurations d'un attracteur, puis la taille de l'ensemble limite.

	Temps transitoire	Ensemble limite
Arbres	degré max ≤ 3	$O(n^4)$
	degré max ≥ 4	Exp
Ligne et cycle	$O(n^3)$	même état partout
Tore, voisinage de von Neumann	Poly? [Regnault, Schabanel, and Thierry (2007)]	complexe
Graphe complet	$O(n \log n)$	même état partout

TAB. 7.2 – Temps transitoires et structure de l’ensemble limite pour la règle minorité sur quelques graphes simples. n est le nombre de sommets du graphe.

La suite immédiate de cette étude serait alors la structure des bassins d’attraction de chaque attracteur.

Nous avons étudié un paramètre, la durée du régime transitoire, mais d’autres seraient également pertinents. Notamment, les recherches pourraient se poursuivre sur le temps de mélange², souvent plus difficile à étudier que le temps d’atteinte, mais néanmoins utile et souvent étudié sur les chaîne de Markov.

Une piste intéressante serait ensuite de reprendre l’étude en variant la définition du modèle. Notamment, nous avons choisi comme seuil pour changer d’état la moitié des voisins dans un certain état. Mais on peut également introduire une préférence vers l’un des deux états (voire ajouter des états) en décalant ce seuil. Cela revient à étudier la robustesse au seuil. Il serait intéressant d’étudier de même la robustesse à l’asynchronisme : a-t-on besoin de la dynamique totalement asynchrone ou bien peut-on se contenter de propriétés plus faibles, par exemple simplement d’une forme d’équité dans les mises à jour des cellules ?

Enfin, ce travail s’est concentré sur des topologies simples, la suite naturelle est donc d’étudier d’autres topologies plus évoluées.

²Un temps de mélange est le temps après lequel la distribution de probabilité de la chaîne de Markov est proche (pour une distance donnée) de la distribution limite. C’est donc le temps nécessaire pour effacer l’influence de la configuration initiale.

Chapitre 8

Conclusion

Notre objectif initial, qui consistait à étudier différentes perturbations des systèmes complexes, nous a amené tout d'abord à identifier des questions concernant la perturbation et la robustesse à ces perturbations, puis à choisir un modèle (celui des automates cellulaires, et souvent une restriction à l'espace des automates élémentaires). Nous avons alors construit des protocoles de simulation pour tenter d'y répondre avec rigueur. Les simulations menées ont permis de distinguer de nombreux comportements : transitions de phase, sensibilités à diverses perturbations, interactions de perturbations, etc.

Dans une première partie, qui porte sur deux modèles de couplage (i.e. la comparaison de deux instances du même système), nous avons formalisé la coalescence et démontré les premières propriétés, puis focalisé l'étude sur les transitions de phase pour identifier la classe d'universalité. Nous avons établi qu'il s'agit ici de la percolation dirigée, ce qui fournit donc un ensemble de nouveaux modèles de cette classe avec des variantes inhabituelles.

Cette approche expérimentale a mis en évidence dans une seconde partie les interactions de deux perturbations : asynchronisme et modifications dynamiques de la topologie. Nous avons tout d'abord montré expérimentalement la richesse de comportements qui émerge de l'interaction de ces deux perturbations, puis nous nous sommes concentrés sur une règle, la règle minorité, pour comparer de façon analytique différentes topologies. Il apparaît que même dans le cas des arbres, la structure de l'ensemble limite et le temps de convergence vers cet ensemble limite ne sont pas triviaux.

Une préoccupation transversale a été la simulation formelle entre différents modèles. En particulier, le chapitre 5 est consacré à l'équivalence entre deux classes d'automates, ce qui nous a amené à proposer une définition de simulation entre automates cellulaires utilisant une source d'aléatoire, généralisant la notion classique de simulation entre automates cellulaires déterministes. Cette notion de simulation reste à explorer : transfert de propriétés, extension au groupage, etc. Par ailleurs, nous n'avons utilisé que deux dynamiques asynchrones, alors qu'il en existe d'autres (par exemple des délais de communication, un balayage systématique des cellules, etc.). Il serait donc utile d'étudier, en fonction du problème, quelles sont les dynamiques équivalentes.

En étudiant les perturbations, nous avons montré que de petites modifications de certains paramètres peuvent avoir une grande influence sur l'évolution du système (c'est le cas du taux de synchronisme) ou au contraire être rapidement « oubliées » (c'est le cas de la configuration initiale quand il y a coalescence rapide). La richesse des phénomènes étudiés nous permet alors d'affirmer l'intérêt de l'étude de la robustesse des automates cellulaires, et plus généralement de la robustesse des systèmes complexes. Une direction

pour de futur travaux est donc de prouver formellement certains des comportements étudiés expérimentalement.

Les techniques que nous avons utilisées doivent maintenant être appliquées sur des types d'automates cellulaires potentiellement « plus complexes », notamment des automates cellulaires modélisant des systèmes réels, ou inspirés par ces derniers. L'enjeu de ces études serait, à terme, de comprendre comment obtenir des systèmes ayant un comportement global prédéfini, ou bien comment rendre robuste à certaines perturbations un système complexe donné.

Bibliographie

- [Bagnoli (1998)] Franco Bagnoli. Cellular automata, 1998. [arXiv: cond-mat/9810012v1](https://arxiv.org/abs/cond-mat/9810012v1). 11
- [Bandini, Mauri, and Serra (2001)] Stefania Bandini, Giancarlo Mauri, and R. Serra. Cellular automata: from a theoretical parallel computational model to its application to complex systems. *Parallel Computing*, 27(5):539–553, 2001. DOI: [10.1016/S0167-8191\(00\)00076-4](https://doi.org/10.1016/S0167-8191(00)00076-4). 12
- [Berman and Simon (1988)] Piotr Berman and Janos Simon. Investigations of fault-tolerant networks of computers. In *STOC: Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 66–77, New York, NY, USA, 1988. ACM. ISBN 0-89791-264-0. DOI: [10.1145/62212.62219](https://doi.org/10.1145/62212.62219). 73
- [Bersini and Detours (1994)] H. Bersini and V. Detours. Asynchrony induces stability in cellular automata based models. In *Proceedings of Artificial Life IV*, pages 382–387, Cambridge, 1994. MIT Press. 12
- [Blanchard, Kůrka, and Maass (1997)] F. Blanchard, P. Kůrka, and A. Maass. Topological and measure-theoretic properties of one-dimensional cellular automata. *Physica D: Nonlinear Phenomena*, 103(1-4):86–99, 1997. 11
- [Boccaro, Fukś, and Geurten (1997)] N. Boccaro, H. Fukś, and S. Geurten. A new class of automata networks. *Physica D: Nonlinear Phenomena*, 103(1-4):145–154, 1997. DOI: [10.1016/S0167-2789\(97\)00257-1](https://doi.org/10.1016/S0167-2789(97)00257-1). 75
- [Broadbent and Hammersley (1957)] Simon R. Broadbent and John M. Hammersley. Percolation processes in crystals and mazes. In *Proceedings of the Cambridge philosophical society*, pages 629–641, 1957. 31
- [Chaudhuri, Chowdhury, Nandi, and Chattopadhyay (1997)] Parimal Pal Chaudhuri, Dipanwita Roy Chowdhury, Sukumar Nandi, and Santanu Chattopadhyay. Additive cellular automata: theory and applications. *IEEE computer Society*, 1997. 42
- [Dai Pra, Louis, and Rœlly (2002)] Paolo Dai Pra, Pierre-Yves Louis, and Sylvie Rœlly. Stationary measures and phase transition for a class of probabilistic cellular automata. *ESAIM Probability and Statistics*, 6:89–104, 2002. DOI: [10.1051/ps:2002004](https://doi.org/10.1051/ps:2002004). <http://www.esaim-ps.org/articles/ps/abs/2002/01/psVol6-4/psVol6-4.html>. 58
- [Demongeot, Aracena, Thuderoz, Baum, and Cohen (2003)] Jacques Demongeot, Julio Aracena, F. Thuderoz, T.-P. Baum, and O. Cohen. Genetic regulation networks: circuits, regulons and attractors. *C.R. Biologies*, 326:171–188, 2003. 92
- [Domany and Kinzel (1984)] Eytan Domany and Wolfgang Kinzel. Equivalence of cellular automata to Ising models and directed percolation. *Physical Review Letters*, 53(4):311–314, 1984. 26, 58

- [Fatès (2003)] Nazim Fatès. Experimental study of elementary cellular automata dynamics using the density parameter. In *Discrete models for complex systems, DMCS '03 (Lyon)*, Discrete Mathematics Theoretical Computer Science Proceedings, AB, pages 155–165. Assoc. Discrete Math. Theor. Comput. Sci., Nancy, 2003. 29
- [Fatès (2004)] Nazim Fatès. *Robustesse de la dynamique des systèmes discrets : le cas de l'asynchronisme dans les automates cellulaires*. PhD thesis, ENS Lyon, December 2004. <http://nazim.fates.free.fr/These.htm>. 12, 26, 40, 44, 88
- [Fatès and Morvan (2004)] Nazim Fatès and Michel Morvan. Perturbing the topology of the game of life increases its robustness to asynchrony. In Peter M. A. Sloot, Bastien Chopard, and Alfons G. Hoekstra, editors, *6th International Conference on Cellular Automata for Research and Industry Proceedings (ACRI)*, volume 3305 of *LNCS*, pages 111–120. Springer, Oct. 2004. 13, 82, 84
- [Fatès, Morvan, Schabanel, and Thierry (2006)] Nazim Fatès, Michel Morvan, Nicolas Schabanel, and Eric Thierry. Fully asynchronous behavior of double-quiescent elementary cellular automata. *Theoretical Computer Science*, 362:1–16, 2006. DOI: [10.1016/j.tcs.2006.05.036](https://doi.org/10.1016/j.tcs.2006.05.036). 96
- [Fatès, Regnault, Schabanel, and Thierry (2006)] Nazim Fatès, Damien Regnault, Nicolas Schabanel, and Eric Thierry. Asynchronous behavior of double-quiescent elementary cellular automata. In José R. Correa, Alejandro Hevia, and Marcos A. Kiwi, editors, *LATIN Proceedings*, volume 3887 of *Lecture Notes in Computer Science*, pages 455–466. Springer, 2006. DOI: [10.1007/11682462_43](https://doi.org/10.1007/11682462_43). 12, 13, 44, 47
- [Fatès (2007)] Nazim A. Fatès. Asynchronism induces second order phase transitions in elementary cellular automata. *Journal of Cellular Automata*, march 2007. arXiv: [nlin/0703044](https://arxiv.org/abs/nlin/0703044). <http://hal.inria.fr/inria-00138051/>. 19, 21, 31, 53
- [Fukás (2004)] H. Fukás. Probabilistic cellular automata with conserved quantities. *Non-linearity*, 17(1):159–173, 2004. 12
- [Gács (2001)] Peter Gács. Reliable cellular automata with self-organization. *Journal of Statistical Physics*, 103(1/2):45–267, 2001. 12
- [Gács (1986)] Peter Gács. Reliable computation with cellular automata. *Journal of Computer and System Sciences*, 32(1):15–78, 1986. 12
- [Gács (1997)] Peter Gács. Deterministic computations whose history is independent of the order of asynchronous updating. Technical report, Boston Univ., Computer Science Dept., 1997. <http://www.cs.bu.edu/~gacs/papers/commut.pdf>. 59, 71
- [Gács and Reif (1988)] Peter Gács and J. Reif. A simple three-dimensional real-time reliable cellular array. *Journal of Computer and System Sciences*, 36(2):125–147, 1988. 12
- [Ganguly, Sikdar, Deutsch, Canright, and Chaudhuri (2003)] Niloy Ganguly, Biplab K Sikdar, Andreas Deutsch, Geoffrey Canright, and P Pal Chaudhuri. A survey on cellular automata. Technical report, Centre for High Performance Computing, Dresden University of Technology, December 2003. <http://www.cs.unibo.it/bison/publications/CAsurvey.pdf>. 12
- [Goldstein, Kuik, Lebowitz, and Maes (1989)] S. Goldstein, R. Kuik, J. L. Lebowitz, and C. Maes. From PCA's to equilibrium systems and back. *Communications in Mathematical Physics*, 125:71–79, March 1989. DOI: [10.1007/BF01217769](https://doi.org/10.1007/BF01217769). 58
- [Goles and Martinez (1990)] Eric Goles and S. Martinez. *Neural and automata networks, dynamical behavior and applications*, volume 58 of *Maths and Applications*. Kluwer Academic Publishers, 1990. 93, 112

- [Golze (1978)] U. Golze. (A-)Synchronous (Non-)Deterministic cell spaces simulating each other. *Journal of Computer and System Sciences*, 17(2):176–193, October 1978. DOI: [10.1016/0022-0000\(78\)90003-X](https://doi.org/10.1016/0022-0000(78)90003-X). 59
- [Grassberger (1999)] Peter Grassberger. Synchronization of coupled systems with spatiotemporal chaos. *Phys. Rev. E*, 59(3):R2520–R2522, Mar 1999. DOI: [10.1103/PhysRevE.59.R2520](https://doi.org/10.1103/PhysRevE.59.R2520). 38
- [Grassberger (1995)] Peter Grassberger. Are damage spreading transitions generically in the universality class of directed percolation? *J Stat Phys*, 79:13–23, September 1995. 22, 33
- [Grimmett and Stirzaker (2001)] G. Grimmett and D. Stirzaker. *Probability and random process*. Oxford University Press, 3rd edition, 2001. 107
- [Hedlund (1969)] G.A. Hedlund. Endomorphism and automorphism of the shift dynamical system. *Mathematical System Theory*, 3:320–375, 1969. 59
- [Hinrichsen (2000)] Haye Hinrichsen. Nonequilibrium critical phenomena and phase transitions into absorbing states. *Advances in Physics*, 7:815–958, Nov 2000. DOI: [10.1080/00018730050198152](https://doi.org/10.1080/00018730050198152). <http://arxiv.org/abs/cond-mat/0001070>. 21, 22, 35, 52, 58
- [Ingerson and Buvel (1984)] T. E. Ingerson and R. L. Buvel. Structure in asynchronous cellular automata. *Physica D Nonlinear Phenomena*, 10:59–68, January 1984. DOI: [10.1016/0167-2789\(84\)90249-5](https://doi.org/10.1016/0167-2789(84)90249-5). <http://adsabs.harvard.edu/abs/1984PhyD...10...59I>. 12
- [Istrail (2000)] Sorin Istrail. Statistical mechanics, three-dimensionality and NP-completeness: I. universality of intractability for the partition function of the ising model across non-planar lattices. In *Proceedings of STOC*, pages 87–96. ACM Press, 2000. 93
- [Kanada (1994)] Y. Kanada. Asynchronous 1D cellular automata and the effects of fluctuation and randomness. In *Proceedings of the fourth conference on artificial life (A-Life IV)*. MIT Press, 1994. 12
- [Kari (2005)] Jarkko Kari. Theory of cellular automata: A survey. *Theoretical Computer Science*, 334:3–33, april 2005. DOI: [10.1016/j.tcs.2004.11.021](https://doi.org/10.1016/j.tcs.2004.11.021). 11
- [Kauffman (1984)] Stuart A. Kauffman. Emergent properties in random complex automata. *Physica D Nonlinear Phenomena*, 10:145–156, Jan 1984. DOI: [10.1016/0167-2789\(84\)90257-4](https://doi.org/10.1016/0167-2789(84)90257-4). 22
- [Kaulakys, Ivanauskas, and Mekauskas (1999)] B. Kaulakys, F. Ivanauskas, and T. Mekauskas. Synchronization of chaotic systems driven by identical noise. *International Journal of Bifurcation and Chaos*, 9(3):533–539, 1999. 40
- [Kinzel (1983)] Wolfgang Kinzel. Directed percolation. In R. Zallen G. Deutscher and J. Adler, editors, *Percolation Structures and Processes*, page 425. Adam Hilger Pub. Co., Bristol, 1983. 58
- [Lehn (1999)] Jean-Marie Lehn. Dynamic combinatorial chemistry and virtual combinatorial libraries. *Chemistry European Journal (Weinheim)*, 5(9):2455–2463, 1999. DOI: [10.1002/\(SICI\)1521-3765\(19990903\)5:9%3C2455::AID-CHEM2455%3E3.3.CO%3B2-8](https://doi.org/10.1002/(SICI)1521-3765(19990903)5:9%3C2455::AID-CHEM2455%3E3.3.CO%3B2-8). 8
- [Lehn (2000)] Jean-Marie Lehn. Programmed chemical systems: Multiple subprograms and multiple processing/expression of molecular information. *Chemistry European*

- Journal (Weinheim)*, 6(12):2097–2102, 2000. DOI: [10.1002/1521-3765\(20000616\)6:12<2097::AID-CHEM2097>3.0.CO;2-T](https://doi.org/10.1002/1521-3765(20000616)6:12<2097::AID-CHEM2097>3.0.CO;2-T). 8
- [Lehn (2002)] Jean-Marie Lehn. Toward complex matter: Supramolecular chemistry and self-organization. *Proceedings of the National Academy of Sciences*, 99(8):4763–4768, 2002. ISSN 0027-8424. DOI: [10.1073/pnas.072065599](https://doi.org/10.1073/pnas.072065599). <http://www.pnas.org/content/99/8/4763.short>. 8
- [Levin et al. (2006)] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. University of California, Berkeley, 2006. <http://www.oberlin.edu/markov/>. 105
- [Mamei, Roli, and Zambonelli (2005)] Marco Mamei, Andrea Roli, and Franco Zambonelli. Emergence and control of macro-spatial structures in perturbed cellular automata, and implications for pervasive computing systems. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, 35(3):337–348, May 2005. ISSN 1083-4427. DOI: [10.1109/TSMCA.2005.846379](https://doi.org/10.1109/TSMCA.2005.846379). 12
- [Maritan and Banavar (1994)] Amos Maritan and Jayanth R. Banavar. Chaos, noise, and synchronization. *Physical Review Letters*, 72:1451–1454, March 1994. DOI: [10.1103/PhysRevLett.72.1451](https://doi.org/10.1103/PhysRevLett.72.1451). 40
- [Martin, Odlyzko, and Wolfram (1984)] Olivier Martin, Andrew M. Odlyzko, and Stephen Wolfram. Algebraic properties of cellular automata. *Communications in mathematical physics*, page 219, 1984. 42
- [McCoy and Wu (1973)] Barry M. McCoy and Tai Tun Wu. *The Two-dimensional Ising model*. Harvard University Press, 1973. 22, 93, 94
- [Morelli and Zanette (1998)] Luis G. Morelli and Damián H. Zanette. Synchronization of stochastically coupled cellular automata. *Phys. Rev. E*, 58(1):R8, Jul 1998. DOI: [10.1103/PhysRevE.58.R8](https://doi.org/10.1103/PhysRevE.58.R8). 28, 29, 38
- [Nakamura (1974)] K. Nakamura. Asynchronous cellular automata and their computational ability. *Systems, Computers, Controls*, 5(5):58–66, 1974. 71
- [Ollinger (2002)] Nicolas Ollinger. *Automates cellulaires: structures*. PhD thesis, ENS Lyon, 2002. <ftp://ftp.ens-lyon.fr/pub/LIP/Rapports/PhD/PhD2002/PhD2002-05.pdf>. 59
- [Pikovskii (1984)] Arkady S. Pikovskii. Synchronization and stochastization of array of self-excited oscillators by external noise. *Radiophysics and Quantum Electronics*, 27:390–395, May 1984. DOI: [10.1007/BF01044784](https://doi.org/10.1007/BF01044784). 40
- [Rapaport (1998)] Ivan Rapaport. *Inducing an order on cellular automata by a grouping operation*. PhD thesis, ENS Lyon, 1998. 59
- [Regnault, Schabanel, and Thierry (2007)] Damien Regnault, Nicolas Schabanel, and Éric Thierry. Progresses in the analysis of stochastic 2D cellular automata: a study of asynchronous 2D minority. In *Proceedings of MFCS*, volume LNCS 4708, pages 320–332. Springer, 2007. 93, 95, 112, 113
- [Regnault, Schabanel, and Thierry (2008)] Damien Regnault, Nicolas Schabanel, and Éric Thierry. On the analysis of "simple" 2D stochastic cellular automata. In *Proceedings of LATA*, 2008. To appear. 93, 95
- [Rojas (1996)] R. Rojas. *Neural networks: a systematic introduction*. Springer, 1996. Chap. 13 - The Hopfield Model. 93, 94

- [Rouquier (2008)] Jean-Baptiste Rouquier. An exhaustive experimental study of synchronization by forcing on elementary cellular automata. In Bruno Durand, editor, *Proceedings of the First Symposium on Cellular Automata*, pages 250–261. MCCME Publishing House, Moscow, April 2008. ISBN 978-5-94057-377-7. <http://hal.archives-ouvertes.fr/hal-00274006/en/>. 28
- [Rouquier and Morvan (2006)] Jean-Baptiste Rouquier and Michel Morvan. Coalescing cellular automata. In Vassil N. Alexandrov, G. Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, *Proceedings of the 6th International Conference on Computational Science (ICCS)*, volume 3993 of *Lectures Notes in Computer Science*, pages 321–328. Springer Berlin, mai 2006. ISBN 3-540-34383-0. DOI: 10.1007/11758532_44. <http://fr.arxiv.org/abs/nlin.CG/0610009>. 38
- [Rouquier and Morvan (2009)] Jean-Baptiste Rouquier and Michel Morvan. Coalescing cellular automata: Synchronization by common random source for asynchronous updating. *Journal of Cellular Automata*, 4:55–78, 2009. arXiv: 0712.1992. <http://oldcitypublishing.com/JCA/JCA.html>. 41
- [Rouquier and Morvan (2008)] Jean-Baptiste Rouquier and Michel Morvan. Combined effect of topology and synchronism perturbation on cellular automata: Preliminary results. In Hiroshi Umeo, Shin Morishita, and Katsuhiko Nishinari, editors, *ACRI*, volume 5191 of *LNCS*. Springer Berlin, 2008. 82
- [Sánchez and López-Ruiz (2005)] Juan R. Sánchez and Ricardo López-Ruiz. Self-synchronization of cellular automata: An attempt to control patterns. In Vassil N. Alexandrov, G. Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, *International Conference on Computational Science (3)*, volume 3993 of *Lecture Notes in Computer Science*, pages 353–359. Springer, nov 2005. ISBN 3-540-34383-0. DOI: 10.1007/11758532_48. arXiv: nlin.CG/0511061. 31
- [Schönfisch and de Roos (1999)] Birgitt Schönfisch and André de Roos. Synchronous and asynchronous updating in cellular automata. *Biosystems*, 51(3):123–143, 1999. ISSN 0303-2647. 12
- [Serra and Villani (2002)] Roberto Serra and Marco Villani. Perturbing the regular topology of cellular automata: implications for the dynamics. In *ACRI*, volume 2493 of *LNCS*, pages 168–177. Springer, 2002. 12
- [Shiryayev (1984)] Albert Nicolaevich Shiryayev. *Probability*, volume 95 of *Graduate Texts in Mathematics*. Springer Berlin, 1984. ISBN 3-540-90898-6. 66
- [Theyssier (2005)] Guillaume Theyssier. *Automates cellulaires: un modèle de complexités*. PhD thesis, ENS Lyon, 2005. <http://tel.archives-ouvertes.fr/tel-00166295/fr/>. 59
- [Toom (1974)] A. Toom. Nonergodic multidimensional systems of automata. *Problems of Information Transmission*, 10:239–246, 1974. 12
- [Toom (1980)] A. Toom. Stable and attractive trajectories in multicomponent systems. *Advances in Probability*, 6:549–575, 1980. 12
- [Toom et al. (1990)] A. Toom, N. Vasilyev, O. Stavskaya, L. Mityushin, G. Kurdyumov, and S. Pirogov. *Stochastic cellular systems: ergodicity, memory, morphogenesis*, chapter Discrete local Markov systems, pages 1–182. Manchester University Press, 1990. 12
- [Wolfram (1984)] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10:1–35, January 1984. DOI: 10.1016/0167-2789(84)90245-8. 11, 38

-
- [Wolfram (1983)] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644, 1983. 18
- [Wuensche (1999)] Andrew Wuensche. Classifying cellular automata automatically: finding gliders, filtering, and relating space-time patterns, attractor basins, and the Z parameter. *Complexity*, 4(3):47–66, 1999. 31
- [Zanette and Morelli (2003)] Damián H. Zanette and Luis G. Morelli. Synchronization of coupled extended dynamical systems: a short review. *International Journal of Bifurcation and Chaos*, 13(4):781–796, 2003. DOI: [10.1142/S0218127403007114](https://doi.org/10.1142/S0218127403007114). 28

Notations principales

Voici les principales notations communes à plusieurs chapitres de cette thèse.

- $\llbracket a, b \rrbracket$ Ensemble des entiers compris entre a et b .
- $a \propto b$ a est proportionnel à b , i.e. a est égal à b à une constante multiplicative près.
- 1234567890 Règle n°1234567890 dans la numérotation de Wolfram des automates cellulaires élémentaires.
- α Taux de synchronisme, i.e. probabilité pour une cellule à une étape d'être mise à jour.
- $\mathcal{B}(p)$ Loi de Bernoulli de paramètre p .
- β Un des exposants critiques d'une transition de phase, voir section 2.2.
- c Configuration. $c(x, t)$ est l'état de la cellule x à l'instant t . $c(\cdot, \cdot)$ est donc un diagramme espace-temps (fonction de deux variables).
- δ Il y a malheureusement télescopage sur cette lettre, mais les deux notations sont tellement habituelles que nous avons préféré ne pas les modifier. Pour un automate cellulaire, δ est la règle de transition. Dans une transition de phase, δ est un exposant critique (i.e. l'exposant d'une des lois de puissance caractéristiques de la classe d'universalité). Grâce au contexte et au type, il est toujours facile de savoir duquel des deux il s'agit.
- $\mathbb{E}(X)$ Espérance de la variable aléatoire X .
- n Nombre de cellules de l'automate cellulaire. Lorsque le graphe est perturbé, c'est le nombre de sommets du graphe.
- $\mathbb{P}(X)$ Probabilité de l'événement X .
- Q Ensemble des états de l'automate cellulaire.
- ρ Densité de cellules en désaccord (utilisé dans les modèles de couplages).
- ρ_∞ Densité asymptotique de cellules en désaccord.
- t Temps, i.e. nombre d'étapes de calcul.
- x Cellule (sommets du graphe d'interaction).

Table des figures

2.1	Exemple de diagramme espace-temps.	17
2.2	Diagramme espace-temps d'accord de la règle 128.	18
2.3	Diagramme espace-temps de la règle 128 en dynamique partiellement asynchrone.	20
2.4	Diagramme espace-temps d'accord de la règle 170.	20
2.5	Percolations isotrope et dirigée.	22
3.1	Le modèle de forçage appliqué à la règle 110.	30
3.2	$\rho_\infty(q)$ (densité asymptotique de cellules en désaccord).	33
3.3	Mesure de β pour la règle 110.	34
3.4	$\rho(q, t)$ pour la règle 58.	35
3.5	Un diagramme espace-temps de la règle 57 dans le modèle de forçage.	38
4.1	$\rho_\infty(\alpha)$ pour quelques règles expérimentalement non rapidement coalescentes.	49
4.2	$\rho_\infty(\alpha)$ pour quelques règles rapidement coalescentes ou non selon α	50
4.3	$\rho_\infty(\alpha)$ pour quelques règles convergeant vers accord ou désaccord total, selon α	51
4.4	Régimes sous et sur-critiques de la règle 110.	52
4.5	Mesure de β pour la règle 58 _h	53
4.6	$\rho(\alpha, t)$ (densité de cellules en désaccord en fonction du temps) pour la règle 6.	55
5.1	Probabilités de transition de l'automate de Domany-Kinzel.	61
5.2	A simule B	63
5.3	Graphe des transitions utiles de l'automate lisant les bits aléatoires.	68
5.4	Transitions possibles de la règle lisant les bits aléatoires.	71
6.1	Quelques surfaces d'échantillonnage pour la combinaison des deux perturbations.	86
6.2	Suite de la figure 6.1.	87
7.1	Transferts de particules.	95
7.2	Une configuration c et sa configuration duale \hat{c}	98
7.3	Trois phases sont nécessaires pour classifier les cellules.	100
7.4	Les trois gadgets utilisés dans la construction des arbres biaisés.	108
7.5	Les trois configurations d'une porte.	108
7.6	Un arbre biaisé.	109
7.7	Évolution d'une position de l'arbre biaisé.	111

Liste des tableaux

1.1	Quelques exemples de systèmes complexes	7
2.1	Table de transition de la règle 23.	18
3.1	q_c et β pour tous les automates élémentaires subissant une transition de phase.	36
4.1	Table de transition de l'automate élémentaire 6.	45
4.2	Une suite de mises à jour intéressantes pour la règle 6.	45
4.3	Transitions intéressantes sur des motifs de la règle 6.	46
4.4	Classification empirique des automates élémentaires selon le comportement face à la coalescence rapide.	48
4.5	Valeurs de α_c pour les règles ayant une transition de phase.	53
4.6	Valeurs expérimentales des exposants critiques β et δ	54
5.1	Table de transition de l'automate lisant les bits aléatoires.	69
6.1	Classification des automates élémentaires selon leur robustesse à chacune des perturbations.	85
7.1	Temps transitoires en dynamiques synchrone et asynchrone pour la règle minorité sur quelques graphes simples.	112
7.2	Temps transitoires et structure de l'ensemble limite pour la règle minorité sur quelques graphes simples.	113

Résumé

L'objet de ce travail est de mieux comprendre ce qui se produit lorsque l'on perturbe un système complexe, en utilisant les automates cellulaires comme modèle. Nous nous intéressons principalement à deux classes de perturbations. La première concerne la dynamique, c'est-à-dire les règles régissant l'évolution du système pour passer d'une étape de temps à la suivante. Par exemple, à chaque étape de temps, on met à jour seulement une partie des cellules (au lieu de mettre à jour toutes les cellules de façon synchrone). La seconde concerne la topologie : on modifie le graphe des connexions entre cellules (le graphe des interactions), par exemple en coupant quelques liens.

Dans une première partie nous utilisons des *couplages*, ce qui consiste à faire évoluer deux copies d'un système selon la même règle, pour le même nombre d'étapes de calcul, mais avec des conditions initiales différentes.

Au premier chapitre de cette partie (chapitre 3), le couplage consiste à modifier certaines cellules pour qu'elles soient dans le même état entre les deux copies. Le couplage étudié au chapitre 4 se place dans le cadre de la dynamique asynchrone : à chaque pas de temps, une partie seulement des cellules est mise à jour. Le couplage consiste alors à mettre à jour le même ensemble de cellules dans les deux copies. Autrement dit, les deux copies partagent la même source d'aléatoire pour décider où appliquer la règle locale.

Dans ces deux modèles, nous observons que, pour certains paramètres, les deux copies deviennent identiques. Nous appelons ce phénomène *coalescence* et en proposons quelques interprétations en termes de modélisation et en termes de systèmes dynamiques.

Pour certains automates cellulaires élémentaires, nous observons une transition de phase, qui appartient dans presque tous les cas à la classe d'universalité de la percolation dirigée. Au chapitre 5, nous commençons une validation formelle de cela. Précisément, nous démontrons que certains automates cellulaires déterministes asynchrones font apparaître une transition de phase de percolation dirigée. Pour cela, nous proposons une définition de simulation entre automates cellulaires utilisant une source d'aléatoire ; notion qui conserve la densité asymptotique donc l'exposant critique associé. Nous montrons alors l'équivalence entre une classe d'automates cellulaires probabilistes (où l'on rencontre la percolation dirigée) et les automates cellulaires déterministes asynchrones.

Dans une seconde partie, nous étudions l'interaction de deux perturbations : asynchronisme et modification de la topologie.

Au chapitre 6, nous vérifions l'intérêt de combiner les deux perturbations. Nous observons que pour certains automates, la robustesse à l'asynchronisme est augmentée lorsque la topologie n'est plus régulière. Pour d'autres, les deux perturbations se combinent de façon non linéaire voire non monotone : les deux perturbations appliquées simultanément ont un effet opposé à l'application d'une seule des deux perturbations.

Au chapitre 7, nous poursuivons l'étude de l'influence de la topologie en comparant plusieurs graphes et en appliquant la même règle d'évolution (la règle minorité) à chacun. Pour chaque graphe, nous déterminons le temps transitoire, ou temps pour atteindre l'ensemble limite (*hitting time*), et décrivons la structure de cet ensemble limite, ou attracteur. On note une distinction sur le temps transitoire (polynomial ou exponentiel) entre les arbres de degré maximum 3 et ceux de degré maximum 4 ou plus.

Nous avons mené à la fois des études expérimentales (classifications des automates cellulaires, mesure de transitions de phase) et des études théoriques (simulations entre modèles, comparaison de topologies pour la règle minorité). L'enjeu de ces études serait, à terme, de savoir obtenir des systèmes ayant un comportement global prédéfini, ou bien de rendre robuste à certaines perturbations un système complexe donné.