



HAL
open science

Système robuste de diagnostic décentralisé à l'aide d'automates cellulaires simples

Nicolas Gauville

► **To cite this version:**

Nicolas Gauville. Système robuste de diagnostic décentralisé à l'aide d'automates cellulaires simples. Automates cellulaires et gaz sur réseau [nlin.CG]. 2018. hal-01894581v1

HAL Id: hal-01894581

<https://inria.hal.science/hal-01894581v1>

Submitted on 12 Oct 2018 (v1), last revised 15 Feb 2019 (v2)

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.



Systeme robuste de diagnostic décentralisé à l'aide d'automates cellulaires simples

Stage effectué au Loria du 05/03/2018 au 31/08/2018

Encadré par Nazim Fatès et Irène Marcovici
Au sein de l'équipe Mocqua

Systeme robuste de diagnostic decentralise a l'aide d'automates cellulaires simples

Nicolas Gauville

Sommaire

Introduction et resume	4
1 Contexte	7
1.1 Systemes complexes	7
1.2 Automates cellulaires	7
1.3 Comportement emergent	8
1.4 Applications des automates cellulaires	8
2 Definition du cadre du sujet	9
2.1 Automate cellulaire isotrope	9
2.2 Diagnostic decentralise	11
2.3 Dynamique des defaillances	11
2.4 Signal de diagnostic	11
2.5 Robustesse	11
2.6 Formalisation du probleme	12
3 Etat de l'art et applications concretes	12
3.1 Les reseaux de capteurs sans fils	13
3.2 Diagnostic a l'aide d'automates cellulaires	13
4 Pistes methodologiques	13
4.1 Diffusion d'une estimation de l'etat de la grille	13
4.2 Modele de Greenberg-Hastings	14
4.3 Probleme de la classification de la densite	14
4.4 Transition de phase	15
5 Transition de phase du modele Greenberg-Hastings	15
5.1 Evolution de la densite	15
5.2 Etude analytique de l'evolution des densites	16
5.3 Etude de l'activite	17
5.3.1 Etude experimentale	17
5.4 Diagnostic a partir de l'activite	17
5.4.1 Modification du modele	17
5.4.2 Propagation d'une alerte	17
5.4.3 Deux etats d'alerte	18
5.5 Robustesse du diagnostic	20
6 Automates RGB	20
6.1 Automate RGB-ab	20
6.2 Variantes de l'automate	22
6.3 Obtenir un diagnostic decentralise avec la regle <i>RGB-a-2</i>	23
6.4 Robustesse du diagnostic	24
6.5 Test de diagnostic avec differentes regles RGB-ab	25
6.6 Observations sur les regles	25
7 Regle semi-chaotique	26
7.1 Regle et voisinage	26
7.2 Test du diagnostic	27

8	Comparaison des méthodes basées sur l'activité	27
8.1	Faible taux de défaillance	27
8.2	Faible taux de défaillance avec asynchronisme	28
8.3	Taux élevé de cellules défaillantes avec asynchronisme	29
8.4	Temps de réaction du diagnostic	29
8.5	Tableau comparatif	29
8.6	Problème de la diffusion d'un état d'alerte	30
9	Diagnostic avec passage à l'échelle	30
10	Bilan et applications concrètes	32
10.1	Diagnostic avec passage à l'échelle	32
10.2	Diagnostic à partir de l'activité sur grille finie	33
10.3	Bilan	34
11	Évolution de Fiatlux	34
11.1	Nouveaux modèles	34
11.2	Interface utilisateur	35
	Conclusion	37
	Bibliographie	38
A	Annexe : autres modèles et expérimentations	41
A.1	Automate isotrope à grille d'états bidimensionnelle	41
A.1.1	Espace des états	41
A.1.2	Fonction de transition locale	41
A.1.3	Diagnostic : estimation du nombre de cellules défaillantes	41
A.1.4	Évaluation de la méthode de diagnostic	41
A.1.5	Intérêt du modèle	42
A.2	Réaction diffusion avec vagues cumulatives	42
A.2.1	Espace des états	42
A.2.2	Règle de transition	43
A.2.3	Résultats	43
A.2.4	Remarques sur le modèle	43
A.3	Règles « Larger than Life »	43
A.3.1	Étude de la densité sur les différents voisinages	44
A.3.2	Variations de la règle	45
A.3.3	Conclusion	45
A.4	Expériences sur le modèle de Greenberg-Hastings	46
A.5	Règle semi-chaotique	47
A.6	Modèle RGB	47
A.7	Comparaison des modèles	50
B	Annexe : autres modèles sur grille infinie	50
B.1	Classification de la densité avec défaillance	51
B.1.1	Voisinage de Toom sans défaillance	51
B.1.2	Diagnostic à deux états	52
B.1.3	Problème de l'asynchronisme	52
B.1.4	Test du diagnostic	53
B.2	Classification de la densité biaisée	54
B.2.1	Règle de transition locale	54
B.2.2	Test de la classification biaisée	54
B.2.3	Résistance à l'asynchronisme	54
B.2.4	Règle des carrés	54
B.2.5	Test du diagnostic	56

Remerciements

Si j'ai eu de nombreuses expériences professionnelles jusqu'ici, mon accueil au Loria s'est avéré être l'un des plus chaleureux que j'ai pu avoir. Je tiens tout particulièrement à remercier *Irène Marcovici*¹ et *Nazim Fatès*², mes deux tuteurs de stage, pour leur gentillesse et leur bienveillance à mon égard. Les nombreuses réunions et discussions que nous avons eu au cours de ce stage, parfois accompagnées d'une infusion de verveine d'Algérie, ont été plus qu'enrichissantes et agréables, tant dans ma découverte du monde de la recherche qu'au niveau plus personnel ; les longues conversations philosophiques que nous avons pu avoir, les nombreux ouvrages qu'ils m'ont prêté ainsi que le temps qu'ils ont pris pour m'aider dans mon travail m'ont permis d'effectuer ce stage dans des conditions plus qu'idéales.

Je tiens évidemment à remercier le *Loria*³ et l'*Inria*⁴ de m'avoir accueilli pour ce stage, et plus particulièrement l'équipe *Mocqua*⁵ et son chef d'équipe *Emmanuel Jeandel*⁶, grâce à qui j'ai eu l'occasion de participer à l'*EJCIM*⁷ et découvrir de nouvelles disciplines en informatique et mathématique. Durant ce stage, j'ai également eu l'occasion de demander de l'aide et des conseils à d'autres chercheurs de différentes équipes, dont Vincent Chevrier qui a pris le temps de m'introduire les réseaux de capteurs, et que je tiens également à remercier.

Enfin, l'ambiance du Loria, les pauses cafés proposées pendant l'EJCIM, la cafétéria ou Caroline nous accueille toujours avec le sourire, m'ont permis de rencontrer de nombreuses personnes de tous les domaines, qu'ils soient stagiaires, thésards ou chercheurs, et ont été autant d'occasions d'avoir des conversations plus qu'intéressantes et permettant de m'ouvrir à de nombreuses disciplines. Ce stage restera pour moi une merveilleuse première expérience dans le monde de la recherche.

1. <http://www.iecl.univ-lorraine.fr/~Irene.Marcovici/>

2. <https://members.loria.fr/nazim.fates/>

3. <http://www.loria.fr/fr/>

4. <https://www.inria.fr>

5. <http://mocqua.loria.fr>

6. <https://members.loria.fr/EJeandel/>

7. <https://ejcim2018.sciencesconf.org>

Introduction

Les systèmes distribués sont aujourd'hui utilisés dans de nombreux domaines, et disposent d'avantages conséquents face aux systèmes centralisés. En effet, de tels systèmes ne font pas intervenir de nœud particulier dont la défaillance entraînerait un dysfonctionnement de l'ensemble du système. Ils bénéficient ainsi d'une robustesse importante malgré une taille potentiellement très grande, et permettent en outre une grande extensibilité. Ces qualités justifient leur présence notable dans le monde de l'informatique : internet, les cartes graphiques ou encore les réseaux de capteurs sans fil sont des exemples significatifs de la mise en œuvre de ce type de système [GDW94].

Malgré cela, les systèmes distribués posent également de nombreuses difficultés : un grand nombre d'algorithmes deviennent beaucoup plus complexes lorsque l'on souhaite distribuer leur exécution sans entité de contrôle global. Par exemple, le problème du tri d'une liste d'éléments dans un ordre donné ne peut généralement pas être envisagé de la même manière s'il doit être effectué par plusieurs machines différentes fonctionnant simultanément. De plus, paralléliser un algorithme s'avère souvent difficile [SHG09]. À l'inverse, des comportements complexes peuvent être observés dans des systèmes distribués alors même que leurs composants semblent pourvus d'un fonctionnement très simple, ce qui caractérise d'ailleurs les systèmes complexes, dont l'évolution peut être difficile à prédire par le calcul.

Ce sujet de stage est né d'un questionnement sur la possibilité d'obtenir des informations sur les caractéristiques topologiques d'un réseau en observant un changement global de comportement [BF11] sur ce dernier. Plus généralement, il s'agit de chercher à diagnostiquer un changement de topologie d'un automate cellulaire, sans établir de contrôle global.

Cette approche peut être particulièrement intéressante dans le cas des systèmes distribués, par exemple pour les réseaux de capteurs : si ces derniers sont robustes aux pannes, l'absence d'autorité globale peut rendre plus complexe la détection d'un problème sur le réseau. De plus, les capacités de calcul et les informations disponibles étant limitées pour chacun des capteurs du réseau, les automates cellulaires, du fait de la simplicité des règles locales utilisées, pourraient s'avérer particulièrement adaptés à ce type de problème [CKM00].

Les réseaux de capteurs ont de nombreuses applications, telles que le monitoring industriel, la surveillance d'indicateurs environnementaux, la sécurité, le contrôle des transports ou encore la médecine. Un système robuste de diagnostic décentralisé pourrait ainsi permettre de détecter, par exemple, la défaillance d'un pourcentage critique de capteurs du réseau. Avec l'approche par automate cellulaire, nous pouvons respecter les contraintes imposées par un réseau de capteurs sans fils : l'accès limité aux informations locales et les capacités de calcul réduites des différents capteurs.

Nous commencerons par établir le contexte du sujet avant de redéfinir plus formellement le cadre et les limites des problèmes étudiés, pour enfin proposer plusieurs pistes pouvant répondre à la problématique proposée.

Résumé

Nous nous intéressons ici à un réseau d'éléments tous identiques possédant chacun un état discret, pouvant chacun interagir avec les autres éléments suffisamment proches. Notre objectif est de détecter, de manière décentralisée, un seuil critique d'éléments défaillants, en panne, ne donnant plus de réponse, dans un réseau. L'idée est de provoquer un changement de comportement global brutal lorsque le seuil de cellules défaillantes est atteint ou dépassé, par exemple la diffusion d'un état d'alerte spécifique. Les cellules défaillantes peuvent être présentes dès l'initialisation ou apparaître progressivement.

Le cas du diagnostic d'un seuil critique de cellules défaillantes à l'aide d'un automate cellulaire simple est une question difficile : en effet, nous ne pouvons pas compter facilement le nombre de cellules défaillantes de manière décentralisée. De plus, le système doit se diagnostiquer lui-même tout en subissant la détérioration : les cellules défaillantes ne peuvent avoir d'état ou transmettre d'information, et des zones de la grille peuvent également être isolées par des cellules défaillantes. Utiliser un automate cellulaire simple impose également d'utiliser une mémoire restreinte : une cellule peut seulement adopter un petit nombre d'états. Le défi est donc de trouver une règle de fonctionnement de l'automate cellulaire robuste à l'asynchronisme et aux changements de topologie, présentant un comportement particulier par rapport à l'apparition de cellules défaillantes.

Plus précisément, nous souhaitons obtenir une règle pour laquelle il n'existe que des cellules dans un état normal lorsqu'il n'y a aucune cellule défaillante, et uniquement des cellules dans un état d'alerte lorsque le taux de cellules défaillantes est supérieur ou égal au seuil critique choisi. Pour cela, nous souhaitons obtenir une règle de transition prenant en entrée le nombre de cellules voisines dans chacun des états (normal, en alerte ou défaillante), et provoquant un changement brutal de comportement au

niveau du seuil critique de cellules défaillantes à détecter. Enfin, il faut également que nous puissions régler ce seuil critique de cellules défaillantes à détecter, à l'aide d'un paramètre de la règle de transition locale définissant le modèle.

Une première approche consiste à introduire un automate cellulaire ayant la propriété suivante : les cellules changent d'état d'autant moins souvent qu'il y a de cellules défaillantes. Plus précisément, s'il y a peu de cellules défaillantes, les cellules resteront *actives* : leur état changera souvent. À l'inverse, un taux élevé de cellules défaillantes provoquera des zones inactives sur la grille, et une cellule restée inactive depuis un trop grand nombre d'itérations pourra alors déclencher un état spécifique d'alerte, se propageant à son tour de proche en proche sur toutes les autres cellules. Pour mettre en place ce signal d'alerte, nous utiliserons un automate cellulaire spécifique où chaque cellule connaît, en plus de son état, le nombre d'itérations écoulées depuis son dernier changement d'état. Cette méthode possède plusieurs avantages : l'état d'alerte est totalement absent en dessous du seuil, et se propage sur toute la grille lorsqu'il apparaît, un *consensus global* est donc assuré en dehors de la période de propagation de l'alerte. De plus, cette méthode permet d'utiliser de nombreuses règles de transition, pourvu que l'activité dépende du nombre de cellules défaillantes.

Cependant, une partie de la grille peut rester inactive suffisamment longtemps pour déclencher l'état d'alerte même en dessous du taux de cellules défaillantes à détecter, par exemple à cause d'un motif particulier de cellules défaillantes, ou d'une zone où les cellules défaillantes sont très proches. De plus, la probabilité que ce cas de figure arrive augmente avec la taille de la grille. Nous allons donc explorer d'autres solutions ne présentant pas ce problème du passage à l'échelle.

Pour nous assurer de trouver une solution ne dépendant pas de la taille de la grille, nous pouvons nous intéresser au cas des grilles infinies : une solution fonctionnant sur une grille infinie sera robuste au problème de propagation d'alerte de la solution précédente. Le problème est formalisé dans la partie 2.6. Puisque nous n'utilisons plus d'état d'alerte se propageant automatiquement sur toute la grille, celui-ci peut alors apparaître et disparaître localement. En dessous du seuil critique, l'état d'alerte pourra tout de même être présent : le *consensus global* est donc moins strict que pour la première solution. Il envahira totalement la grille à partir du seuil critique de cellules défaillantes détecté. La figure 1 présente des configurations à un instant donné pour différentes propositions de modèles.

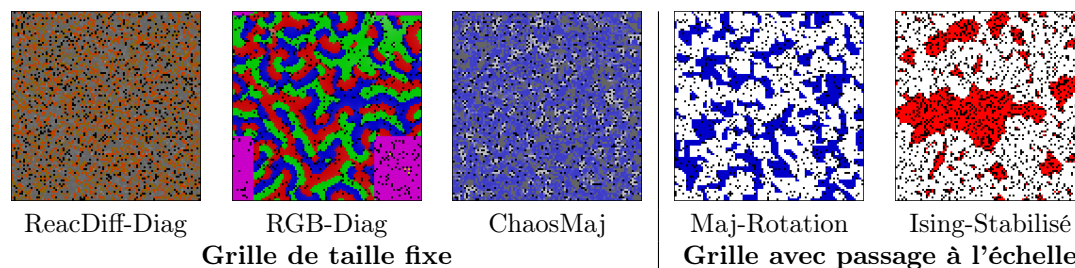


Fig. 1. Représentation des grilles de cellules des automates cellulaires pour différents modèles proposés. Cellules défaillantes en noir, autres états en couleurs.

Solutions avec propagation d'alerte

La figure 1 ci-dessus montre une représentation de configurations de trois des modèles étudiés dans le cas des grilles de taille fixe. La méthode de diagnostic est identique pour les trois premiers (à gauche de la figure) : on utilise un automate cellulaire spécifique mesurant l'activité. Une cellule restée stable pendant trop longtemps (un nombre fixé d'itérations) passe dans un état d'alerte se propageant alors sur toute la grille. Ainsi, l'idée générale est d'utiliser une règle de transition provoquant une activité perpétuelle en l'absence de défaillance, mais laissant progressivement apparaître des zones fixes (provoquant donc une alerte), lorsque trop de cellules défaillantes apparaissent.

Les trois modèles testés fonctionnent bien avec cette méthode : *Greenberg-Hastings* (modèle de réaction diffusion appelé *ReacDiff-Diag* ci-dessous, voir partie 4.2), le modèle *RGB-a-2* (proposé par mes soins) ainsi qu'un modèle mêlant la règle de majorité avec une règle chaotique de manière probabiliste. Ces différentes méthodes sont comparées dans la partie 10.3, notamment sur la façon dont nous pouvons paramétrer le seuil critique de cellules défaillantes à détecter ou leur résistance à l'asynchronisme.

Le modèle *RGB-a-2* est le meilleur candidat trouvé dans le cas d'une grille de taille fixe (ou variant faiblement), celui-ci étant robuste à l'asynchronisme et aux topologies irrégulières, pouvant ainsi être facilement utilisé dans des cas concrets.

Problème du passage à l'échelle

Les modèles sur grille de taille fixe permettent un diagnostic décentralisé efficace, mais sont sensibles aux changements de taille. Trouver des modèles fonctionnant théoriquement sur une grille infinie nous assure ainsi d'avoir un résultat ne dépendant plus de la taille de la grille, et pouvant donc fonctionner sur des réseaux dont la taille évolue significativement, ce qui peut arriver dans des cas concrets.

Le cas du passage à l'échelle se présente comme étant plus abstrait mais aussi plus difficile à aborder. Nous ne pouvons notamment plus utiliser de système de propagation d'états à travers la grille. À l'inverse, pour certains problèmes tels que la classification de la densité (voir partie 4.3), des solutions existent dans le cas de grilles infinies, mais ne fonctionnent plus dans le cas de grilles toriques ou finies (c'est notamment le cas de la règle de majorité sur un voisinage de *Toom*). Une première solution consiste en un automate aboutissant à une configuration où toutes les cellules sont en alerte ou normales en fonction du taux de cellules défaillantes sur la grille, inspiré de la classification de la densité (voir partie B.2), mais cette méthode s'avère difficilement paramétrable, et inadaptée dans le cas de l'apparition progressive des cellules défaillantes. Nous allons donc chercher d'autres types de règles dans lesquelles nous pourrions inclure un paramètre pour choisir le taux de cellules défaillantes à détecter.

Nous nous intéressons alors à des règles de transition locale totalisantes : prenant en entrée le nombre de cellules voisines dans chacun des états, et donnant en sortie un nouvel état (normal ou alerte). Ces règles donneront un état normal en l'absence de cellules voisines en alerte ou défaillantes, et en alerte si aucune cellule voisine n'est dans l'état normal. Sinon, la règle donnera un état d'alerte avec une certaine probabilité, calculée à partir du nombre de cellules voisines dans chacun des états. Enfin, ces règles doivent produire un changement brutal de comportement au niveau du seuil critique de cellules défaillantes à détecter.

Une transition de phase (voir partie 4.4), notion empruntée à la physique statistique, correspond à un changement brutal de comportement d'un système à partir d'une valeur critique d'un paramètre. Par exemple, l'eau connaît deux transitions de phase à pression ambiante, changeant brutalement d'état à 0 et 100°C. Pour l'eau comme pour notre diagnostic, nous souhaitons un changement de comportement global pour un seuil critique donné (dans notre cas le taux de cellules défaillantes), pour un ensemble d'éléments en interaction locale.

Une transition de phase correspondant à un changement brutal de comportement à partir d'un certain seuil critique, nous pouvons raisonnablement nous attendre à ce que seules les règles non-linéaires puissent produire une transition de phase. Expérimentalement, seules les règles testées contenant au moins un carré donnent des résultats probants, mais difficilement paramétrables. Une autre règle, inspirée du modèle d'Ising de physique statistique, présentée dans la partie 9, a l'avantage d'avoir un paramètre permettant de définir le seuil critique de cellules défaillantes à détecter.

Évaluation des modèles proposés

De nombreuses pistes sont explorées dans ce rapport. Pour les évaluer, nous pouvons les comparer sur de nombreux critères, tels que la robustesse à l'asynchronisme, aux changements de topologie et aux changements d'échelle, et la possibilité de choisir le taux de cellules défaillantes à détecter. La partie 10.3 présente un tableau comparatif des modèles proposés.

Mesures, calculs et expérimentations

Pour réaliser les différentes expérimentations et mesures présentées dans ce rapport, les différents modèles étudiés ont été implémentés sur Fiatlux⁸ (un logiciel de simulation d'automates cellulaires). Les mesures et expérimentations ont ensuite été programmées et exécutées sur des grilles finies et toriques. Dans le cas du passage à l'échelle, il n'est bien sûr pas possible de simuler une grille de taille infinie. Nous utiliserons également la théorie du champ moyen, empruntée à la physique, consistant dans notre cas à supposer que les états des cellules d'une configuration à un instant donné sont répartis aléatoirement pour approximer l'évolution de l'activité ou des densités des différents états.

Applications concrètes des méthodes proposées

Le diagnostic décentralisé à partir d'automates cellulaires simples peut trouver des applications dans différents domaines comme celui des réseaux de capteurs sans fils. Les différentes approches développées dans ce rapport peuvent répondre à différents types d'applications concrètes en fonction des contraintes spécifiques du problème, telles que l'asynchronisme, des changements plus ou moins importants de

8. <http://fiatlux.loria.fr>

topologie ou de taille du réseau. Selon le modèle employé, différentes méthodes peuvent être utilisées pour obtenir le diagnostic (voir section 10).

1 Contexte

1.1 Systèmes complexes

Les systèmes complexes se caractérisent par un comportement global souvent difficile à prévoir mathématiquement, ce qui nous amène à recourir à l'observation pour avoir une intuition de leur comportement, et essayer une étude analytique. Les propriétés sur les cas généraux sont en effet indécidables. Définir précisément ces derniers s'avère difficile du fait de la diversité des systèmes pouvant être regroupés sous cette appellation, allant d'interactions entre molécules aux dynamiques sociales d'un groupe d'individus. De manière générale, un grand nombre de comportements caractéristiques sont présents dans les systèmes complexes : une robustesse globale importante malgré une éventuelle fragilité des composants du système, une auto-organisation des composants et l'émergence de comportements cohérents ou l'apparition de motifs. La compréhension et la modélisation de tels systèmes n'est pas aisée, et constitue un domaine d'étude important [HSP04].

On trouve de très nombreux exemples de systèmes complexes dans la nature : les fourmilières bâties par les colonies de fourmis pourtant individuellement inconscientes de la structure créée par ces dernières en est un exemple bien connu, de même que les termitières, dont la structure permet une ventilation passive et le maintien de la température et l'hygrométrie adaptées à leurs hôtes ou encore le cerveau ou les réseaux de gènes.

Aujourd'hui, il existe également de nombreux systèmes complexes artificiels, tels que l'Internet, la bourse ou les réseaux sociaux. Ainsi, mieux comprendre ces systèmes, et notamment comment prédire leur évolution, par exemple par la simulation, est un enjeu majeur regroupant de nombreux domaines tels que la biologie, les sciences sociales et l'informatique.

1.2 Automates cellulaires

Dans les années 1960, John von Neumann se lança dans un projet complexe : construire une machine capable de se reproduire elle-même. Avec les conseils du mathématicien Stanislaw Ulam, il chercha à construire un modèle le plus simple possible d'une telle machine, et c'est ainsi que le premier automate cellulaire naquit. Un automate cellulaire est une grille de cellules pouvant chacune adopter un nombre fini d'états et évoluant au cours du temps, à partir de règles locales prenant en considération l'état des cellules voisines. Ce premier automate fut le *copieur et constructeur universel*, proposé par von Neumann dans les années 1960, qui parvint à construire un motif auto-répliquant sur une grille de cellules à deux dimensions [JvN66].

Par la suite, les automates cellulaires ont été source d'intérêt en mathématique et en informatique théorique. Le *jeu de la vie* proposé par John H. Conway en 1970, automate cellulaire Turing-Complet⁹ a été source de fascination bien au delà du cadre scientifique, et nombreux sont ceux qui se sont pris au jeu de créer diverses structures composées de *puffeurs*¹⁰, *vaisseaux* et *canons*¹¹ pour observer le comportement et l'évolution souvent surprenante des motifs réalisés [Bel15].

Dès 1969, l'ingénieur allemand Konrad Zuse émit l'idée que le monde pourrait être le résultat d'un gigantesque automate cellulaire [Zus69], ce qui impliquerait que les lois physiques soient discrètes. Le jeu de la vie étant Turing-Complet (comme d'autres automates cellulaires, tels que la règle 110 sur un automate unidimensionnel [Coo04]), certains envisagent qu'en simulant le jeu de la vie sur une grille initialisée aléatoirement de taille suffisamment grande, une forme de vie consciente pourrait un jour s'y développer, voire que notre univers ne soit en fait que le produit d'un gigantesque automate cellulaire [OT99].

Les automates cellulaires sont source d'intérêt dans de nombreux domaines, et ils ont notamment permis de simuler et comprendre divers phénomènes physiques et biologiques. Nous pouvons par exemple citer la pigmentation de certains coquillages qui possèdent des motifs complexes que l'on retrouve dans certains automates cellulaires unidimensionnels (par exemple le *Conus textile* avec la règle 30, voir figure 2 ci-contre), les feux de forêt, ou encore le trafic routier.



Fig. 2. Évolution de la règle 30, et un *Conus textile* présentant des motifs similaires.

9. Système formel doté d'une puissance de calcul équivalente à celle des machines de Turing.

10. De l'anglais « puffer » (générateur de fumée) structure mobile laissant une traînée de débris sur son passage.

11. Objets du jeu de la vie dotés de comportements particuliers (https://fr.wikipedia.org/wiki/Jeu_de_la_vie).

Nous qualifierons d'automate cellulaire les systèmes composés d'une grille de cellules répondant aux critères suivants :

1. Chaque cellule possède un état choisi parmi un ensemble fini d'états.
2. À chaque itération, une cellule met à jour son état en fonction de son état courant et de celui de son voisinage (un ensemble fini de cellules défini par la topologie de la grille), à l'aide d'une règle de transition.
3. Toutes les cellules appliquent la même règle de transition à chaque itération.

Nous définirons un formalisme pour appréhender ces automates dans la partie 2.1.

1.3 Comportement émergent

Dans différents domaines tels que les systèmes multi-agents ou les automates cellulaires, la question de l'émergence des comportements complexes observés à l'échelle macroscopique, par rapport à la simplicité des règles appliquées aux agents ou cellules est centrale. En effet, si de nombreuses expériences simples permettent de mettre en évidence ces comportements émergents, les prédire avant de les simuler nous est encore souvent impossible, ce qui a d'ailleurs fortement contribué à la popularité du *jeu de la vie* de J. Conway [Ada10]. Ainsi, tenter de lier les règles de transition locales aux comportements qu'elles permettent d'obtenir est un axe de recherche important.

La classification des comportements observés est également un problème complexe ; Wolfram a proposé une classification des automates cellulaires en fonction de leur comportement en quatre classes [Wol85] : ceux aboutissant à un état fixe (classe I), un état périodique (classe II), chaotique (classe III) ou complexe (classe IV). Cependant, la diversité des comportements observés rend difficile la création de classes idéales pour les regrouper, et la classification des automates dans les différentes classes est parfois difficile, mais différentes approches ont été proposées [Gut90, dSBdSO⁺16].

Nous allons ici chercher à produire un comportement émergent précis (voir section 2.2) à partir de règles locales simples.

1.4 Applications des automates cellulaires

Dans son essai *La science entre technique et culture* [LL13], Jean-Marc Lévy-Leblond s'interroge sur la place des sciences dans notre société. S'il insiste sur la difficulté de définir ce qu'est la science, l'auteur oppose cette dernière à la technique, notamment parce que les applications concrètes sont restées anecdotiques jusqu'à très récemment dans l'histoire de l'humanité. La question des applications concrètes des automates cellulaires, étudiés en informatique théorique et en mathématiques, se présente alors comme plus difficile qu'il n'y paraît.

Une première réponse simple à cette question apparaît évidente lorsque nous considérons les automates cellulaires comme une branche contrainte des systèmes multi-agents [CF08, SFS10]. Comme ces derniers, les automates cellulaires permettent de simuler un grand nombre de phénomènes complexes, notamment observables dans la nature tels que les feux de forêt ou la génération de motifs présents dans la nature, comme sur les *Conus textile* cités plus tôt ou encore sur les brosses tigrées [VNP⁺08].

Au delà des simulations, d'autres applications concrètes ont été trouvées à certains automates cellulaires, telles que la génération de nombres aléatoires ; un automate cellulaire est d'ailleurs utilisé à cette fin dans le logiciel Mathematica [Wol86]. La question du diagnostic décentralisé à l'aide d'automates cellulaires simples traitée ici fait également partie de ces applications.

L'attention portée aux automates cellulaires s'explique cependant certainement pour une autre raison que leur capacité à simuler des systèmes complexes ou leur aspect combinatoire. Ils n'ont d'ailleurs été inventés pour aucune de ces deux applications, et leur origine nous rappelle un autre intérêt mettant en valeur les caractéristiques les plus fascinantes de ces systèmes. Lors de la création du tout premier automate cellulaire, von Neumann avait en tête un projet plus qu'ambitieux. En effet, après avoir donné sa propre définition du vivant, se voulant la plus générale possible, excluant notamment la notion de matière, il a cherché à produire un automate capable des caractéristiques de sa définition du vivant, dont l'auto-reproduction [VNB96]. Le constructeur universel peut alors être considéré comme une expérience de pensée, cherchant à répondre à la question du vitalisme¹² dans le cadre strict de sa propre définition du vivant (l'expérience n'ayant d'ailleurs pu être implémentée que 50 ans plus tard [Pes95]).

12. Pensée selon laquelle les êtres vivants sont mû par une force vitale immatérielle.

2 Définition du cadre du sujet

2.1 Automate cellulaire isotrope

De très nombreux types d'automates cellulaires peuvent être envisagés. Nous allons ici nous restreindre à un type d'automate particulier : les automates cellulaires isotropes.

Automate cellulaire isotrope

Un automate cellulaire isotrope est un automate cellulaire dont la règle de transition locale prend en compte l'état des cellules voisines indépendamment de leur position. La fonction de transition locale prend ainsi en entrée le nombre de cellules voisines se trouvant dans chacun des états de l'espace d'états.

Un automate cellulaire est un cas particulier de système dynamique discret, composé d'une grille régulière de cellules pouvant chacune adopter un certain nombre d'états, et d'une fonction de transition ϕ faisant évoluer l'état des cellules de la grille à chaque itération. Nous nous restreignons ici au cas des grilles bidimensionnelles.

Grille de cellules \mathcal{L}

Nous pourrions définir cette grille de différentes manières :

- $\mathcal{L} = \mathbb{Z}^2$ pour une grille régulière infinie de cellules.
- $\mathcal{L} = \{1, \dots, n\} \times \{1, \dots, m\}$ pour une grille de taille (n, m) (avec $n, m \in \mathbb{N}^*$).
- $\mathcal{L} = \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ pour une grille torique de taille (n, m) .

Dans les expérimentations présentées ici, nous utiliserons une grille torique pour éviter les problèmes de bords et ainsi conserver un voisinage régulier.

Espace des états Q

Ensemble non vide des états qu'une cellule peut prendre à un instant donné. Nous parlerons d'espace d'états binaire lorsque $Q = \{0, 1\}$.

Espace des configurations \mathcal{C}

L'ensemble des configurations que l'automate peut adopter est représenté par l'espace des configurations $\mathcal{C} = Q^{\mathcal{L}}$.

Notation : nous noterons $\mathcal{P}(E)$ l'ensemble des parties finies d'un ensemble quelconque E .

Voisinage d'une cellule $V(c)$

Le voisinage d'une cellule, $V : \mathcal{L} \rightarrow \mathcal{P}(G)$ correspond aux cellules dont l'état peut être pris en compte par la règle de transition locale de l'automate cellulaire. Dans le cas de grilles de cellules régulières, nous utiliserons généralement l'un des voisinages suivants :

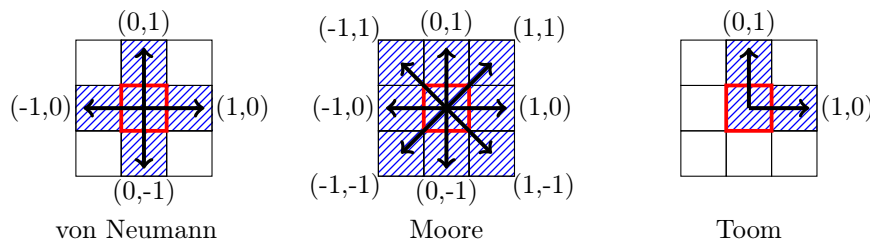


Fig. 3. Voisinages usuels (cellule considérée entourée en rouge, voisines hachurées en bleu).

Si nous définissons chacun des voisinages par un ensemble de vecteurs, nous pouvons déterminer le voisinage d'une cellule donnée de la façon suivante :

Von Neumann : $V_n = \{(-1, 0), (0, -1), (1, 0), (0, 1)\}$.

Moore : $V_m = \{(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, -1), (1, 0), (1, 1)\}$.

Toom : $V_t = \{(0, 1), (1, 0)\}$.

$V(c) = \{c + j; j \in V_x\}$ avec $V_x \in \{V_m, V_n, V_t\}$.

Règle de transition locale f

Chaque cellule de l'automate met à jour son état avec la règle de transition locale $f : Q \times \mathbb{N}^Q \rightarrow Q$, prenant en entrée l'état courant de la cellule considérée ainsi que le nombre de cellules voisines se trouvant dans chacun des états de Q .

Les règles utilisant ces paramètres sont dites *totalisantes-externes* (*outer-totalistic*). De façon à pouvoir facilement appliquer la règle de transition locale sur une cellule donnée, nous utiliserons également la fonction qui compte le nombre d'état du voisinage :

$$\begin{aligned} \eta : \mathcal{C} \times \mathcal{L} &\longrightarrow \mathbb{N}^Q \\ (x, c) &\longmapsto (|\{c' \in V(c); x_{c'} = q\}|)_{q \in Q}. \end{aligned}$$

Fonction de transition globale ϕ

La fonction de transition globale $\phi : \mathcal{C} \rightarrow \mathcal{C}$, associe une configuration de \mathcal{C} à chaque configuration, après application de la règle de transition locale à toutes les cellules. Elle est définie par :

$$\begin{aligned} \phi : \mathcal{C} &\longrightarrow \mathcal{C} \\ x &\longmapsto y \text{ tel que } \forall c \in \mathcal{L}, y_c = f(x_c, \eta(x, c)). \end{aligned}$$

Nous noterons x_0 la configuration initiale (correspondant au temps zéro), puis pour $t \geq 0$, $x_{t+1} = \phi(x_t)$. Pour simplifier les notations, nous utiliserons les définitions suivantes :

Définition 1 — Densité d'un état q sur une configuration x de \mathcal{C}

Pour une grille de taille finie, nous noterons $|x|_q$ le nombre de cellules de la configuration x se trouvant dans l'état q .

$$|x|_q = |\{i, x_i = q\}| \text{ avec } x \in \mathcal{C}, q \in Q.$$

et $d_q(x)$ le ratio de cellules de la configuration x se trouvant dans l'état q .

$$d_q(x) = \frac{|x|_q}{|x|} \text{ avec } x \in \mathcal{C}, q \in Q.$$

Dans le cas où $\mathcal{L} = \mathbb{Z}^2$, la densité d'un état q peut être définie par

$$d_q(x) = \lim_{\substack{n \rightarrow \infty \\ m \rightarrow \infty}} \sum_{i=-n}^n \sum_{j=-m}^m \frac{\sigma(q, x_{ij})}{(2n+1)(2m+1)} \text{ avec } \sigma(q, q') = \begin{cases} 1 & \text{si } q' = q, \\ 0 & \text{sinon.} \end{cases}$$

Notons que pour une grille de taille infinie, la densité d'une configuration est définie si et seulement si la limite existe.

Définition 2 — Cellule défaillante

Une cellule défaillante est une cellule se trouvant dans un état particulier D de Q ayant les propriétés suivantes :

1. L'application de la règle de transition locale f de l'automate sur une cellule se trouvant dans l'état D donnera ce même état D quel que soit son voisinage.
2. L'application de la règle de transition locale f sur une cellule dans un état $q \neq D$ donnera un état $q' \neq D$.

Dit autrement, une cellule défaillante (resp. non-défaillante) restera défaillante (resp. non défaillante) après application de la règle de transition locale.

Nous allons redéfinir plus précisément le cadre et les limites de la problématique étudiée. Pour ce faire, nous commencerons par définir un exemple simple qui nous aidera à expliciter les comportements que nous souhaitons obtenir.

Considérons un automate cellulaire sur une grille torique de taille 10×10 . Supposons maintenant qu'à chaque instant, une cellule normale ait une certaine probabilité de devenir défaillante. Nous pourrions alors vouloir diagnostiquer la présence de plus de 10% de cellules défaillantes de la grille, de manière décentralisée.

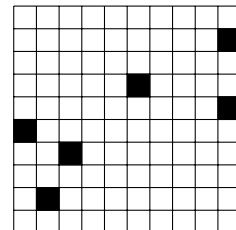


Fig. 4. Représentation de la grille de l'automate

2.2 Diagnostic décentralisé

Nous allons essentiellement nous concentrer sur le cas du diagnostic d'un certain pourcentage de cellules défaillantes sur la grille, comme présenté dans la section précédente. Cependant, nous pouvons considérer différents cas à diagnostiquer :

- Un **seuil critique de cellules défaillantes** dans la grille. Par exemple, détecter la défaillance de plus de 10% des cellules.
- Un **changement de topologie** de la grille de l'automate. Par exemple, détecter la disparition de plus de 10% des liens de voisinage entre les cellules de la grille, de la même façon que pour les cellules défaillantes.
- Un **asynchronisme** dans l'application de règle de transition locale. Ce cas particulier peut être plus simple que les deux précédents, dans la mesure où de nombreux automates cellulaires adoptent un comportement très différent lorsque l'on introduit un asynchronisme dans la règle de transition locale. Plusieurs cas d'asynchronisme peuvent être considérés [Bou13] :
 1. **α -synchronisme** : pour chaque itération de la fonction de transition globale, chaque cellule est mise à jour avec une probabilité α , et laissée inchangée avec une probabilité $(1 - \alpha)$.
 2. **β -synchronisme** : toutes les cellules sont mises à jour à chaque pas de temps, mais la transmission du nouvel état aux cellules voisines est effectuée avec une probabilité β , et l'état observable demeure inchangé avec une probabilité $(1 - \beta)$.
 3. **γ -synchronisme** : toutes les cellules sont mises à jour à chaque pas de temps, mais le nouvel état est transmis à chacune des cellules voisines avec une probabilité γ . Il s'agit d'un cas proche du β -synchronisme où, cette fois, chaque voisine est considérée indépendamment dans sa probabilité d'accéder au nouvel état de la cellule.

Le diagnostic devant être décentralisé, l'information doit être accessible depuis l'ensemble de la grille de cellules, ce qui inclut deux problématiques distinctes :

1. Diffuser l'information à l'ensemble des cellules non défaillantes de la grille.
2. Obtenir un consensus des cellules sur le diagnostic.

2.3 Dynamique des défaillances

Quel que soit le type de défaillance considéré, nous pouvons envisager deux cas simples pour les introduire sur la grille :

- Les défaillances sont présentes dès l'initialisation de l'automate et restent fixées ensuite. Ce cas simple permet de vérifier une méthode de diagnostic et mesurer facilement le temps nécessaire au diagnostic (nombre d'itérations avant détection).
- Apparition / disparition aléatoire de défaillances sur la grille au fil des itérations, la règle de diagnostic devant alors produire un état d'alerte ou non au fil du temps en fonction de l'évolution de l'état de la grille.

2.4 Signal de diagnostic

Maintenant que nous savons ce que nous pouvons chercher à diagnostiquer, il nous faut également définir comment signaler la présence du phénomène à diagnostiquer. Là encore, nous pouvons considérer plusieurs solutions :

- Diffusion d'un état spécifique d'alerte A sur toutes les cellules non défaillantes de la grille.
- Changement brutal et notable de comportement général de l'automate (voir section 4.4).

2.5 Robustesse

Un automate peut être plus ou moins robuste à différents types de perturbations, et les disparités selon les règles appliquées sont importantes. Nous pourrions évaluer la robustesse des automates réalisés sur les types de perturbations suivants :

- Altération de l'état des cellules.
- Changement de topologie de la grille en cours d'exécution (notamment suppression de lien de voisinage entre deux cellules)

- Application asynchrone de la règle de transition (par exemple, introduire une probabilité pour une cellule de ne pas appliquer la règle de transition locale lors d’une itération). Les réactions des différents automates à l’introduction d’asynchronismes sont très variables selon les règles de transition, nombre d’entre elles peuvent ainsi produire des comportements très différents lorsqu’elles sont appliquées de manière asynchrone [Fat04].

Notons que le système doit être, de façon évidente, robuste à la situation qu’il doit détecter. Cependant, une méthode robuste aux asynchronismes de la règle de transition serait souhaitable dans la mesure où le synchronisme dans des architectures distribuées pose de nombreuses difficultés [AIR88], et celle-ci sera donc évaluée pour les différentes propositions.

2.6 Formalisation du problème

Nous allons maintenant formaliser le problème pour définir strictement le comportement que nous souhaitons obtenir. Soit une configuration x munie d’une densité de cellules défaillantes initiale $d = d_D(x_0)$. Nous voulons diagnostiquer un seuil critique de cellules défaillantes ρ , et ainsi produire une configuration en alerte ou non en fonction de d , soit :

$$\begin{array}{ll} \text{si } d_D(x_0) < \rho & \text{alors } \lim_{t \rightarrow \infty} d_A(x_t) < (1 - d_D(x_0)) \times \epsilon \\ \text{sinon si } d_D(x_0) \geq \rho & \text{alors } \lim_{t \rightarrow \infty} d_A(x_t) + d_D(x_0) = 1. \end{array}$$

Ces conditions peuvent sembler latitudinaires, notamment parce qu’elles permettent une densité de cellules en état d’alerte $d_A(x_t)$ importante (seulement inférieure à $(1 - d_D(x_0)) \times \epsilon$) même en dessous du seuil ρ . Cependant, les cellules défaillantes empêchant toute transition d’information, des cellules ou groupes de cellules peuvent se retrouver isolés du reste de la grille, empêchant ainsi un consensus de l’ensemble de la grille. Nous ne pourrions donc pas espérer trouver de solution permettant une absence totale de cellules en état d’alerte en dessous du seuil critique, et au mieux borner cette densité.

Durée du diagnostic t_{min}

Pour évaluer une méthode de diagnostic, nous pouvons considérer sa vitesse, c’est-à-dire le nombre d’itérations qu’il faut pour atteindre l’état d’alerte si ce dernier doit se déclencher. Plus formellement, nous appellerons t_{min} le nombre d’itérations (s’il existe et est fini) tel que :

$$t_{min} = \min \{t > 0; d_A(x_t) + d_D(x_0) = 1 \text{ si } d_D(x_0) \geq \rho\}$$

Cas de l’apparition progressive des défaillances

Nous allons maintenant supposer que les défaillances surviennent progressivement. Plus précisément, nous considérons qu’une cellule non défaillante possède une probabilité p_f de devenir défaillante lors d’une itération de la fonction de transition globale, et une cellule défaillante le reste indéfiniment. Soit t_1 le nombre d’itérations nécessaires pour que le seuil de cellules défaillantes ρ soit atteint, t_2 le nombre d’itérations nécessaires pour que la configuration soit en alerte, et d le temps de réaction du diagnostic. Plus formellement :

$$\begin{cases} t_1 = \min\{t > 0; d_D(x_t) \geq \rho\}, \\ t_2 = \min\{t > 0; d_A(x_t) = 1 - d_D(x_t)\}, \\ d = t_2 - t_1. \end{cases}$$

3 État de l’art et applications concrètes

Les automates cellulaires peuvent être utilisés dans des domaines variés. S’ils ont d’abord servi de support pour des expériences abstraites, on les retrouve aujourd’hui utilisés comme modèle de calcul massivement parallèle, outils de simulation de phénomènes physiques et biologiques, diagnostic de panne pour circuits intégrés où encore dans la reconnaissance de schémas et d’images [Das12]. Nous nous intéressons ici au cas particulier du diagnostic décentralisé à l’aide d’automates cellulaires simples. Ce type de système peut alors plus particulièrement concerner des systèmes tels que des circuits intégrés à très grande échelle¹³ (VLSI) ou les réseaux de capteurs sans fil.

13. https://fr.wikipedia.org/wiki/Intégration_à_très_grande_échelle

3.1 Les réseaux de capteurs sans fils

Les réseaux de capteurs sans fils consistent en un ensemble de capteurs (correspondants chacun à un ou plusieurs capteurs associés à une batterie, une unité de calcul et un système de communication sans fil) répartis dans un espace à surveiller (pouvant aller du corps d'un être vivant à des terrains de plusieurs hectares). Leur utilisation est particulièrement indiquée lorsque la zone à surveiller est grande, difficile d'accès, ou dans un environnement hostile pour l'homme (températures extrêmes, champ de bataille, ...). De manière générale, les capteurs seront disséminés aléatoirement sur la zone à surveiller, ce qui implique que la topologie du réseau sera irrégulière. Ces derniers vont alors communiquer de proche en proche pour pouvoir signaler les éventuels changements perçus après un consensus des capteurs du réseau [CKM00]. Cette façon de procéder implique plusieurs contraintes notables :

- **Une topologie irrégulière** du réseau (les capteurs étant disséminés aléatoirement). De plus, les capteurs fonctionnent généralement à l'aide d'une batterie, et ne pourront pas être remplacés. Au fil du temps, des capteurs vont donc épuiser leur batterie et s'éteindre définitivement. Les capteurs peuvent également être abîmés ou détruits pour différentes raisons (conditions météorologiques, ennemis, ...). Lorsque c'est nécessaire, d'autres capteurs seront ajoutés au réseau (à nouveau, généralement disséminés de manière aléatoire) pour remplacer ceux qui se sont éteints, mais il s'agira souvent de la seule opération de maintenance possible (des capteurs placés dans une zone ennemie ou trop dangereuse pour l'homme ne pourront ainsi jamais être réparés ou vérifiés par ceux qui les ont posés).
- **Des capacités fortement limitées** en terme de calcul, de temps et de communication et de mémoire. En effet, l'énergie étant précieuse (puisque les batteries ne seront jamais remplacées de manière générale), les calculs, l'envoi de messages aux autres capteurs et les intervalles de communication doivent être gérés parcimonieusement.
- **Une absence de contrôle centralisé** nécessitant alors une auto-organisation du réseau tout en conservant une gestion de ce dernier à distance.
- **Des enjeux de sécurité complexe**, notamment dans les utilisations militaires en milieu hostile. Les capteurs pouvant être accessibles physiquement uniquement par les ennemis dans certains cas, les attaques possibles sont nombreuses : modification d'un capteur, introduction de répliques modifiées de capteurs (pouvant par exemple altérer les messages transmis, analyser le trafic, diffuser de fausses informations, ...) [WLSC07, Ish96].

3.2 Diagnostic à l'aide d'automates cellulaires

Des méthodes de diagnostic à partir d'automates cellulaires ont également été proposées, notamment dans le but de localiser des cellules défectueuses [MBDS11]. Ce type de diagnostic à partir d'automate cellulaire a également été étudié sur les circuits intégrés à très grande échelle (VLSI), dans l'optique de détecter et localiser une éventuelle défaillance matérielle sur un circuit imprimé très dense [SGC05, SGMC01].

Nous allons maintenant voir plusieurs pistes pouvant aider à la conception d'un système robuste de diagnostic décentralisé.

4 Pistes méthodologiques

Différentes approches peuvent être envisagées en vue de développer un système de diagnostic décentralisé. De plus, un grand nombre d'études et expérimentations ont été menées sur les automates cellulaires, et il est alors intéressant de se pencher sur des problèmes proches qui ont déjà été étudiés en profondeur pour développer des propositions d'automates pouvant répondre au problème de diagnostic décentralisé.

4.1 Diffusion d'une estimation de l'état de la grille

La diffusion d'une estimation de l'état de la grille à l'ensemble des cellules, par exemple encodée ou représentée par l'état de celles-ci, permet d'obtenir une indication de l'état global à partir de n'importe quelle cellule de la grille. Cette approche est utilisée dans la partie A.1 en annexe.

4.2 Modèle de Greenberg-Hastings

Le modèle de réaction-diffusion proposé par Greenberg et Hastings permet de modéliser simplement la propagation de vagues excitatrices à travers une grille de cellule [JMG78]. Cet automate cellulaire pourrait nous aider à réaliser un diagnostic décentralisé grâce à plusieurs propriétés :

- Les vagues excitatrices se propagent sur la grille dans toutes les directions, ce qui permet une diffusion de l'information.
- Les vagues sont déformées par diverses perturbations de l'automate (asynchronisme, cellule défaillante, lien de voisinage rompu), et s'annulent lorsque deux vagues opposées se rencontrent. Ainsi, celles-ci peuvent permettre une détection aisée de l'apparition d'un problème sur la grille de cellules.

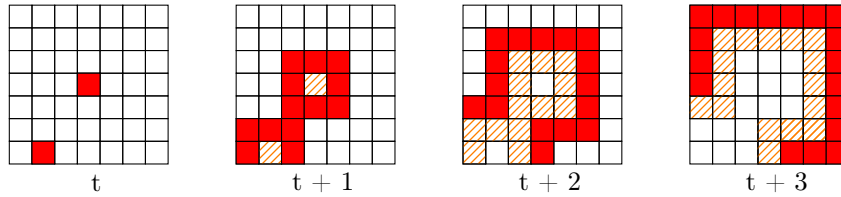


Fig. 5. Propagation d'une vague excitatrice (en rouge) suivie d'un état réfractaire (hachures orange) avec un voisinage de Moore.

Des versions de ce modèle sont utilisées dans la partie 5 et le modèle de vagues cumulatives dans l'annexe A.2

4.3 Problème de la classification de la densité

Le problème de classification de la densité consiste à décider, sur une grille de cellules binaires ($Q = \{0, 1\}$), l'état dont la densité sur la grille est la plus élevée, de manière décentralisée. Concrètement, il s'agit de trouver une règle locale f pour laquelle une configuration \mathcal{C} évoluera vers une configuration où toutes les cellules à l'état 1 (respectivement à l'état 0) si la densité de 1 de \mathcal{C} est supérieure à 0,5 (respectivement inférieure) [Mar13].

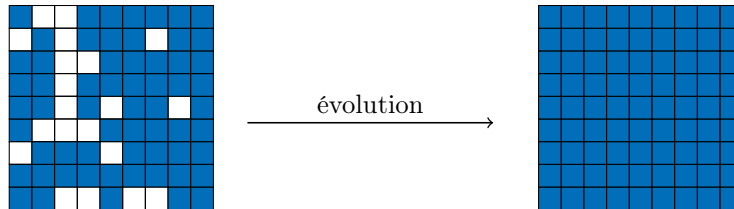


Fig. 6. Une configuration avec une majorité de cellules bleues (respectivement blanches) évoluera en une configuration toute bleue (resp. toute blanche).

Nous pouvons aisément faire un parallèle avec le problème de diagnostic décentralisé d'un certain pourcentage de cellules défaillantes sur la grille. En effet, nous pourrions reformuler le problème ainsi : considérons un automate possédant un état d'alerte particulier q_a . Nous souhaitons que cet état soit adopté par toutes les cellules non défaillantes si la densité de cellules défaillantes de la grille est supérieure à un seuil ρ après un nombre fini d'itérations de la fonction de transition globale. Formellement, pour une configuration C_1 finie possédant une densité de cellules défaillantes $d_{q_a}(C_1)$, on a :

$$\exists k_0 \in \mathbb{N}, \forall k > k_0 \text{ tel que } \phi^k(C_1) = C_2$$

$$\text{avec } \begin{cases} d_D(C_2) + d_A(C_2) = 1 \text{ si } d_D(C_1) > \rho \\ d_A(C_2) = 0 \text{ sinon} \end{cases}$$

Le problème de la classification de la densité ne peut être décidé par un automate cellulaire déterministe en dimension deux sur un espace d'états binaire [FMT16, Fat13], mais une combinaison de la règle de trafic et de la règle de majorité sur un automate sur un automate non déterministe permet de résoudre le problème avec une précision arbitrairement grande [BsFMM13].

4.4 Transition de phase

La détection d'un seuil critique peut s'effectuer par l'observation d'une transition de phase dans le comportement de l'automate. Ce changement de comportement peut ensuite être observé localement sur une cellule selon différents critères, comme les variations d'états de la cellule [CC07, Bou13].

Définition 3 — Transition de phase

La notion de *transition de phase*, empruntée à la physique, correspond au changement d'état d'un système à la suite du changement d'un ou plusieurs paramètres. Les phases correspondent à des régions identifiables de l'espace des *paramètres de contrôle* dans lesquelles l'état du système, caractérisé par des *paramètres d'ordre*, reste le même qualitativement et varie de manière continue. Les *transitions de phase* surviennent lorsqu'un paramètre de contrôle franchit un seuil critique particulier, provoquant un changement brutal et conséquent de comportement.

Dans l'optique d'obtenir un diagnostic décentralisé d'un seuil critique de cellules défaillantes (voir section 2.2), nous pouvons envisager de trouver des règles menant à une transition de phase pour laquelle :

- Les *paramètres de contrôle* correspondent à la densité de cellules défaillantes de la grille.
- Le *seuil critique* de la transition de phase correspond au seuil critique de cellules défaillantes que l'on cherche à détecter.

5 Transition de phase du modèle Greenberg-Hastings

L'introduction d'asynchronisme, de liens ou cellules défaillantes sur la grille produit des changements de comportements notables sur le modèle de réaction-diffusion de Greenberg-Hastings. Notamment, une version stochastique de ce modèle avec une probabilité d'excitation p_r (probabilité qu'une cellule neutre devienne excitée lorsqu'elle est entourée d'au moins une cellule excitée), présente une transition de phase appartenant à la classe d'universalité de la percolation dirigée [BF11]. La transition de phase observée peut ainsi être une piste de diagnostic de défaillance.

Nous avons alors les règles suivantes :

1. Une cellule dans l'état normal peut passer à l'état « excitée » avec une probabilité p_r si au moins une de ses voisines est excitée (t_1).
2. Une cellule excitée passe à l'état réfractaire après une itération (t_2).
3. Une cellule réfractaire redevient normale après une itération (t_3).

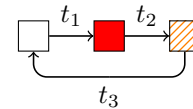


Fig. 7. Transitions possibles

5.1 Évolution de la densité

Les graphiques ci-dessous (figures 8 et 9) représentent les résultats d'expériences réalisées avec FiatLux sur une grille de taille 512×512 , sur 500 itérations à partir d'une configuration aléatoire.

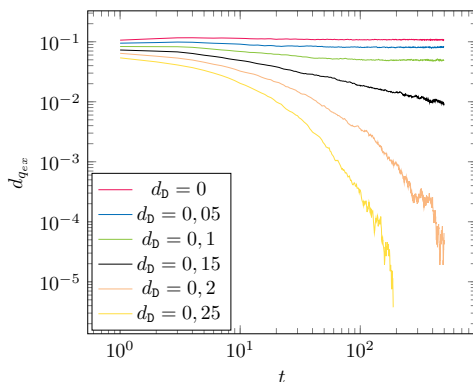


Fig. 8. Évolution de la densité de cellules en état d'excitation pour $p_r = 0,5$

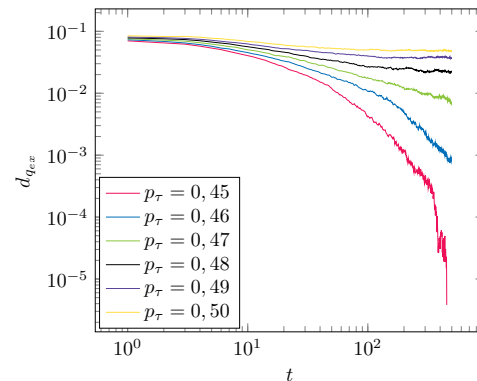


Fig. 9. Évolution de la densité sur une grille avec 10% de cellules défaillantes pour différentes probabilités d'excitation.

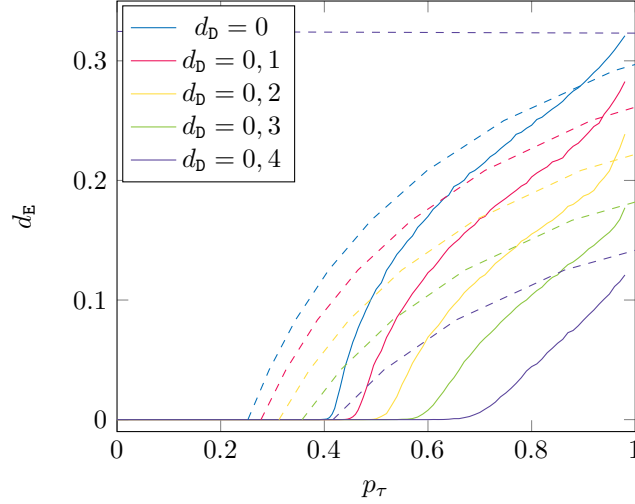


Fig. 10. Évolution de la densité de cellules excitées en fonction de la probabilité d’excitation p_τ pour différentes densités de cellules défailtantes. Les résultats théoriques sous hypothèse du champ moyen sont en pointillés.

Nous allons alors essayer de déterminer pour différents taux de synchronisme (nous considérerons ici un α -synchronisme, comme défini dans la section 2.2) et différentes densités de cellules défailtantes (pour l’instant, ces taux sont fixés lors du tirage des configurations initiales) la probabilité d’excitation p_τ nécessaire pour que la quantité de cellules excitées soit nulle en au plus 1000 itérations. La figure 46 en annexe présente les résultats obtenus.

5.2 Étude analytique de l’évolution des densités

De manière plus analytique, nous allons reprendre l’hypothèse du champ moyen et supposer une indépendance de la distribution des densités des états après application de la fonction de transition.

Soient N_t , E_t , R_t et D_t les densités de cellules respectivement dans les états normal, excité, réfractaire et défailtant au temps t . Nous noterons α la probabilité qu’une cellule soit mise à jour lors d’une itération de la fonction de transition globale.

Sous hypothèse du champ moyen, nous pouvons alors définir D_{t+1} , N_{t+1} , E_{t+1} et R_{t+1} ainsi :

$$\begin{cases} D_{t+1} = D_t \\ N_{t+1} = \alpha[R_t + N_t \times (1 - E_t)^4 + N_t \times (1 - (1 - E_t)^4)] + (1 - \alpha)N_t \\ E_{t+1} = \alpha[N_t \times (1 - (1 - E_t)^4) \times p_\tau] + (1 - \alpha)E_t \\ R_{t+1} = \alpha E_t + (1 - \alpha)R_t \\ D_{t+1} + N_{t+1} + E_{t+1} + R_{t+1} = 1 \end{cases}$$

Notons N , E , R et D les limites de ces densités lorsque t tends vers $+\infty$ (en supposant que ces limites existent). Nous pouvons alors simplifier les équations :

$$\begin{cases} N = \alpha[E + N \times (1 - E)^4 + N \times (1 - (1 - E)^4)] + (1 - \alpha)N \\ E = \alpha[N \times (1 - (1 - E)^4) \times p_\tau] + (1 - \alpha)E \\ R = E \\ D + N + 2E = 1 \end{cases}$$

$$\begin{cases} \frac{1}{p_\tau} = (1 - D - 2E) \times (4 - 6E + 4E^2 - E^3) \\ R = E \\ D + N + 2E = 1 \end{cases}$$

Pour une densité D donnée, la deuxième équation permet alors de lier la probabilité d’excitation p_τ avec la densité de cellules excitées. On peut alors calculer, pour les valeurs de $\frac{1}{p_\tau}$ supérieure à 1, la probabilité p_τ à donner au système pour obtenir, théoriquement, la densité E de cellules excitées sur la grille (courbes en pointillé sur la figure 10).

Notons que nous avons ici maintenu l'hypothèse du champ moyen lorsque t tend vers $+\infty$, considérant ainsi les cellules comme réparties aléatoirement, ce qui n'est plus vrai après itération de la règle de transition, ce qui peut justifier les différences entre les résultats théoriques et expérimentaux de la figure 10, mais capture tout de même le comportement observé expérimentalement.

Le modèle de Greenberg-Hastings produit une transition de phase pouvant servir au diagnostic d'un nombre donné de cellules défaillantes, où le nombre de cellules excitées devient nul au delà d'un certain taux de cellules défaillantes. Ce taux peut être ajusté avec la probabilité d'excitation p_τ , en tenant compte des éventuels asynchronismes. Cependant, une fois la densité de cellules excitées devenue nulle, elle ne peut plus remonter, provoquant un état absorbant (où toutes les cellules non défaillantes sont dans l'état normal).

Pour diagnostiquer un taux de défaillance simplement, nous pourrions nous intéresser à l'activité du système de façon à détecter l'absence de changement d'état d'une cellule après un certain nombre d'itérations.

Définition 4 — Activité

Nous noterons $a(x)$ l'activité d'une configuration $x \in \mathcal{C}$, le rapport du nombre de cellules qui vont changer d'état lors de la prochaine transition, sur le nombre total de cellules de la grille. Ainsi, une activité de zéro correspond à un point fixe lors de l'application de la fonction de transition globale.

5.3 Étude de l'activité

De façon à pouvoir considérer l'activité comme critère de diagnostic, nous allons étudier comment celle-ci est affectée par le taux de cellules défaillantes sur la grille.

5.3.1 Étude expérimentale

La figure 43 (en annexe) montre l'évolution de l'activité des cellules non-défaillantes en fonction de la densité de cellules défaillantes sur la grille. L'annexe A.4 présente également une étude analytique de l'évolution de l'activité.

5.4 Diagnostic à partir de l'activité

5.4.1 Modification du modèle

Pour réaliser un diagnostic à partir de l'activité, nous allons utiliser un second espace d'état Q' qui servira de compteur d'inactivité, et nous assignerons à chaque cellule de l'automate un état de Q ainsi qu'un état de Q' . Ainsi, l'état dans q' augmentera à chaque itération si l'état dans q demeure inchangé, et repassera à 0 (premier état de q') à chaque changement.

La fonction de transition prendra alors en compte les deux états courants de la cellule ainsi que l'état de ses voisines dans Q . Le second espace d'état Q' sert à compter le nombre d'itérations pendant lesquelles la cellule n'a pas changé d'état de Q , on aura $Q' = \{0, 1, \dots, n\}$, et on notera q'^+ l'état suivant de q' dans Q' , par exemple $1^+ = 2$).

Nous allons également ajouter un état d'alerte A dans Q , déclenché si le dernier état de Q' est atteint, c'est à dire si une cellule n'a pas changé d'état de Q pendant n itérations ou plus.

La nouvelle fonction de transition est définie de la manière suivante :

$$f_2 : Q \times Q' \times \mathbb{N}^Q \longrightarrow Q \times Q'$$

$$(q, q', n_N, n_E, n_R) \longmapsto \begin{cases} (q_a, n) & \text{si } q = f(q, n_N, n_E, n_R) \text{ et } q' = n \\ (f(q, n_N, n_E, n_R), q'^+) & \text{si } q = f(q, n_N, n_E, n_R) \\ (f(q, n_N, n_E, n_R), 0) & \text{sinon} \end{cases}$$

5.4.2 Propagation d'une alerte

De façon à obtenir un consensus des cellules sur l'état de la grille, nous allons ajouter une nouvelle règle de propagation de l'alerte : toute cellule non défaillante qui a au moins une voisine en état d'alerte passe en état d'alerte.

Pour évaluer le diagnostic, nous allons, pour une probabilité d'excitation p_τ fixée, tester différentes configurations aléatoires en faisant varier le taux de cellules défaillantes, et compter, pour chaque taux, le nombre de configurations où l'état d'alerte s'est propagé. La figure 12 montre les résultats obtenus pour différentes probabilités d'excitation p_τ et différents seuils de déclenchement $|Q'|$ avec l'état A (courbes bleue, rouge et verte).

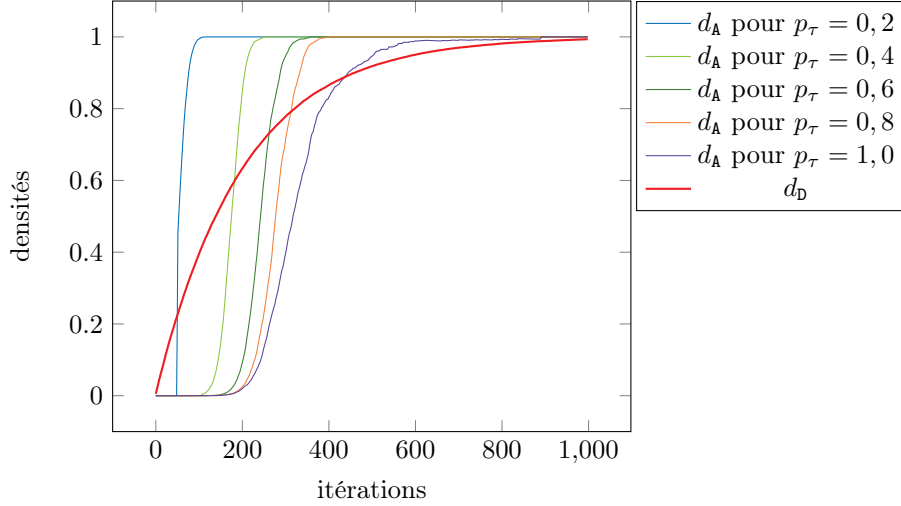


Fig. 11. Évolution de la densité de cellules en alerte pour différentes probabilités d’excitation p_τ , mesurée sur une grille de 200×200 cellules.

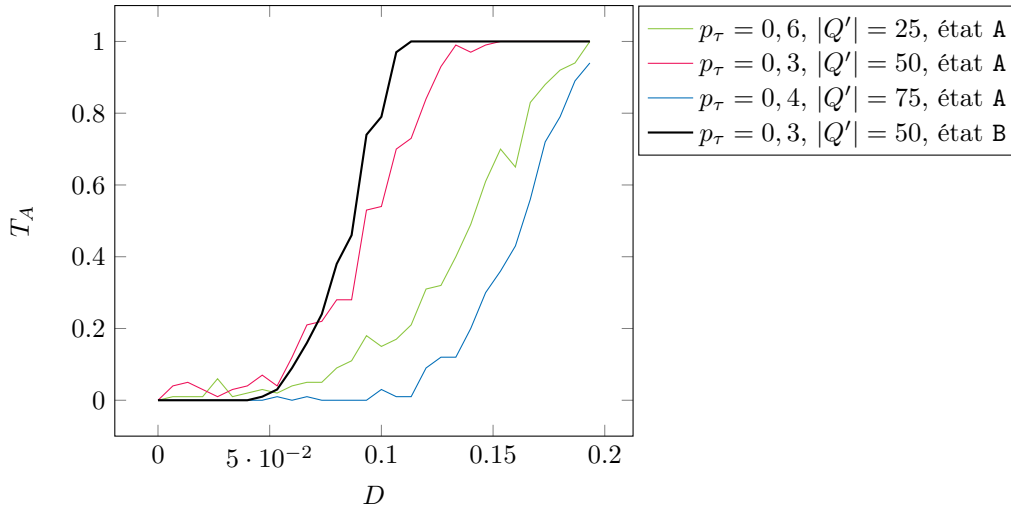


Fig. 12. Taux de configurations en alerte (présence de l’état A ou B) sur 100 tests après 500 itérations sur des grilles initialisées aléatoirement (200×200 cellules) pour différentes densités de cellules défaillantes.

5.4.3 Deux états d’alerte

On constate que de petites zones peuvent facilement rester sans activité pendant un grand nombre d’itérations même avec un taux de cellules défaillantes faible. Ceci peut être considéré comme des faux-positifs. Pour y remédier, nous pouvons par exemple utiliser deux états d’alerte (A et B) fonctionnant de la manière suivante :

- Une cellule restant stable pendant n itérations passe en état d’alerte A.
- Une cellule en état d’alerte A et ayant au moins 4 voisins dans l’état A passe à l’état d’alerte B. Sinon, elle peut redevenir excitée suivant la même règle qu’une cellule normale.
- Une cellule ayant au moins une voisine dans l’état B passe à l’état B (propagation de l’alerte, maintenant appliquée à l’état B) plutôt que l’état A.

Pour tester les différentes méthodes de diagnostic, nous considérerons qu’une configuration est en alerte si plus de 10% des cellules non-défaillantes sont en état d’alerte (puisque l’état d’alerte se propagera au reste des cellules dans les itérations suivantes), soit si $\frac{|c|_A}{1-|c|_D} > 0,1$. Nous noterons T_A le taux de configurations en alerte pour 100 configurations testées après 500 itérations (sauf indication contraire).

On constate une amélioration du diagnostic avec l’ajout du deuxième état d’alerte : les configurations avec moins de 5% de cellules défaillantes ne sont plus en état d’alerte, celles avec plus de 0,11% de cellules défaillantes sont toutes en alerte (courbe noire sur la figure 12).

De façon à pouvoir détecter un pourcentage donné de cellules défaillantes sur la grille, nous allons essayer de déterminer les valeurs à affecter à p_τ et $|Q'|$ en fonction du taux de défaillance à détecter. Une valeur de $|Q'|$ trop faible donne difficilement de bons résultats, et nous aimerions également garder une valeur pas trop élevée pour éviter de ralentir trop fortement le diagnostic, nous allons alors garder $|Q'| = 50$.

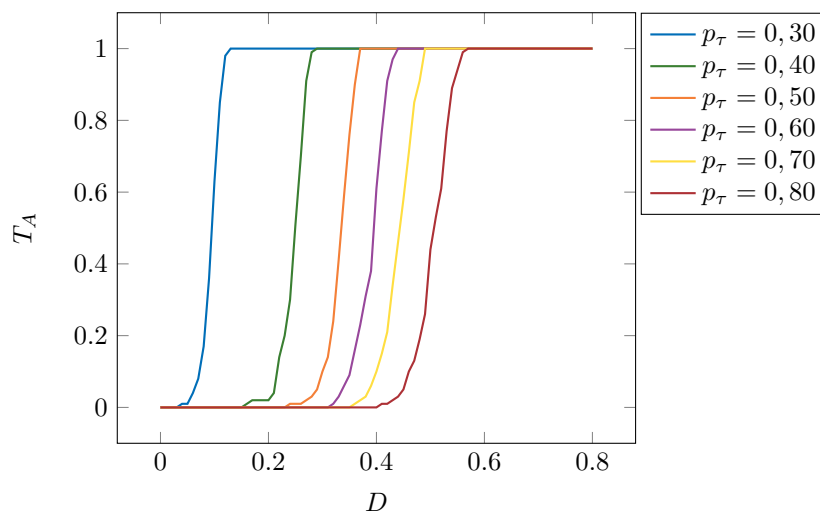


Fig. 13. Taux de configurations en alerte en fonction de la densité de cellules défaillantes pour 100 configurations tirées aléatoirement. Pour effectuer ces tests, la densité de cellules défaillantes à été augmentée de 3,33% toutes les 200 itérations, et le nombre de configurations en alerte a été compté avant chaque augmentation de la densité de cellules défaillantes.

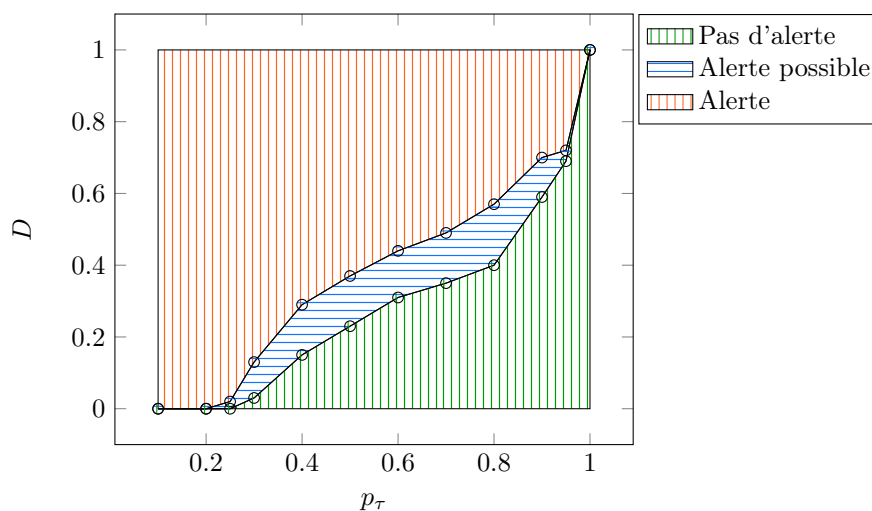


Fig. 14. Déclenchement d'alerte en fonction de la probabilité d'excitation p_τ et de la densité de cellules défaillantes D . La zone verte correspond aux valeurs de D et p_τ pour lesquelles aucune configuration n'a provoqué d'alerte sur les 100 tests effectués, la zone rouge correspond aux valeurs qui ont déclenché un état d'alerte sur toutes les configurations, et la zone bleue correspond aux valeurs pour lesquelles une partie des configurations s'est retrouvée en état d'alerte.

La figure 13 montre les densités de cellules défaillantes à partir desquelles l'alerte à été déclenchée, sur 100 tests effectués pour chaque probabilité d'excitation p_τ .

La figure 14 illustre comment choisir une valeur de p_τ pour diagnostiquer une densité de cellules défaillantes D donnée. Nous avons ainsi une première méthode de diagnostic répondant à la problématique que nous nous sommes posés, nous allons alors nous intéresser à la robustesse de cette méthode.

5.5 Robustesse du diagnostic

Influence de la taille de la grille

La figure 44 (en annexe) montre le taux de configurations en alerte après 200 itérations pour $p_\tau = 0, 3$, sur des grilles de différentes tailles. On constate que la taille de la grille affecte légèrement le diagnostic ; ce n'est pas étonnant du fait que des structures provoquant une zone inactive sur la grille (menant ensuite à un état d'alerte) ont une probabilité d'apparaître plus importante si la grille est plus grande. Les différences engendrées restent cependant faibles.

α -synchronisme

Nous allons maintenant tester la robustesse de la méthode de détection en effectuant l'évaluation du diagnostic pour différents taux d' α -synchronisme, pour p_τ et $|Q'|$ fixés.

La figure 45 (en annexe) montre le taux de configurations en alerte en fonction de la densité de cellules défaillantes pour différents α -synchronismes. On constate qu'en réduisant la probabilité de mise à jour α l'alerte est déclenchée plus tôt (pour une densité de cellules défaillantes plus faible). Cela implique donc qu'il faudra modifier la valeur de la probabilité d'excitation p_τ pour une densité de cellules défaillantes à diagnostiquer D donnée de façon à prendre en compte le taux d' α -synchronisme.

Nous avons une première réponse à la problématique étudiée, mais nous souhaitons trouver des méthodes plus robustes aux asynchronismes. Nous allons maintenant explorer d'autres règles en vue de trouver des automates plus résistants aux asynchronismes que le modèle de *Greenberg-Hastings*.

6 Automates RGB

Les automates cycliques ont déjà été fortement étudiés dans divers travaux [Fis90, FGG91], et correspondent à des automates pour lesquels la règle locale impose de changer selon un ordre circulaire prédéterminé. Les automates présentés dans cette section sont légèrement différents de ces derniers : le premier permet des transitions dans deux sens, et tous prennent plusieurs paramètres qui limitent plus fortement les changements d'état en fonction du voisinage. Nous nous intéressons ici à des automates à trois états : $Q = \{R, G, B\}$ sur un voisinage de Moore et paramétré par deux entiers a et b . Nous allons explorer différentes valeurs de ces paramètres.

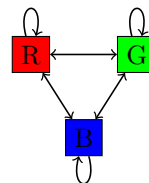


Fig. 15. Transitions possibles de l'automate RGB-ab.

6.1 Automate RGB-ab

Chaque cellule compte les quantités de voisins des deux couleurs qui ne sont pas celle de son état courant. On nommera q^+ l'état à sa gauche et q^- l'état à sa droite sur le graphe ci-contre (figure 15), et n_{q^+} et n_{q^-} les quantités de cellules voisines dans ces états respectifs (ainsi que n_b la quantité de cellules voisines défaillantes).

Chaque règle est paramétrée par deux valeurs a et b ($a, b \in \{0, 1, \dots, 7\}$) et fonctionne de la manière suivante :

$$f : \quad Q \times \mathbb{N}^Q \longrightarrow Q$$

$$(q; n_R, n_G, n_B) \longmapsto \begin{cases} q^+ & \text{si } n_{q^+} > a \\ q^- & \text{si } n_{q^-} > b \\ q & \text{sinon} \end{cases}$$

Nous avons donc 64 valeurs possibles pour le couple de paramètres a et b .

Exploration des règles

L'espace des paramètres possibles étant restreint (64 couples (a, b) possibles), nous pouvons aisément tester tous les paramètres possibles (voir figure 16).

Plusieurs règles ont des comportements surprenants, dont notamment :

- **Règle RGB-30** : cette règle produit un résultat surprenant : elle forme un labyrinthe de couleurs entre-mêlées avec des vagues se propageant sur celui-ci indéfiniment.

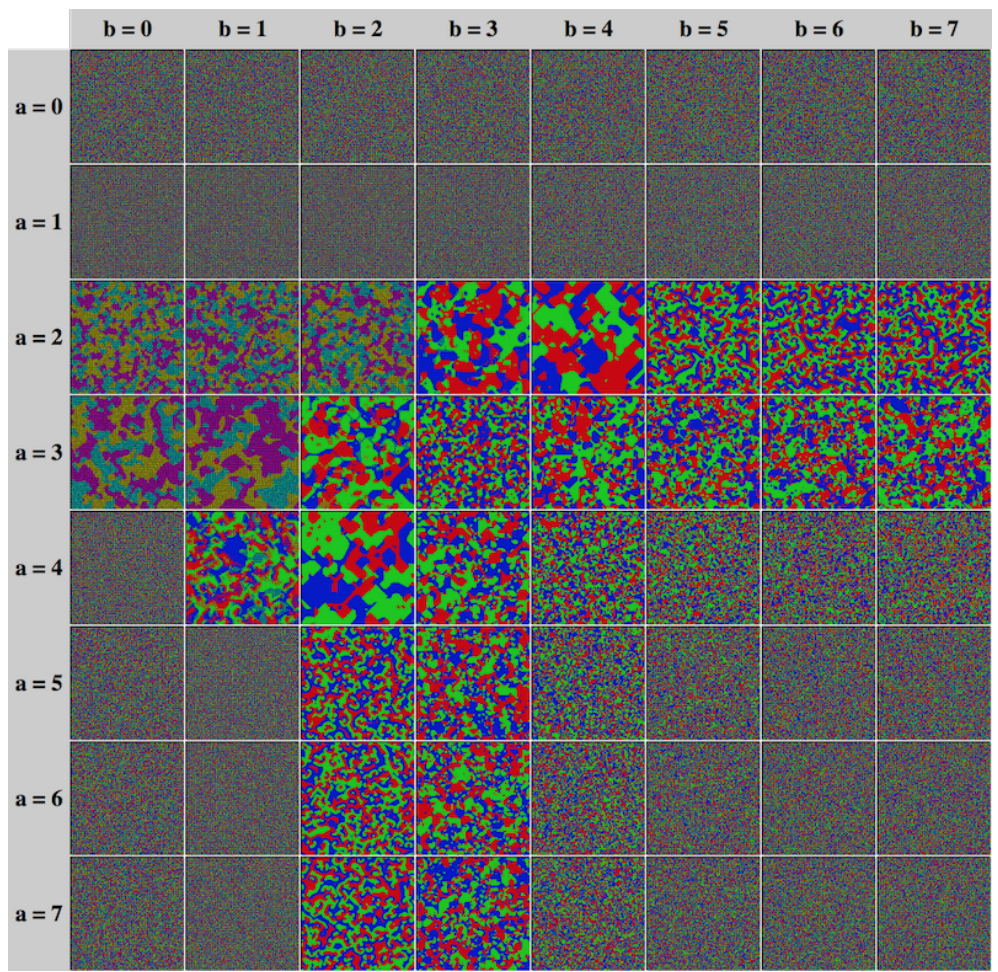


Fig. 16. Toutes les règles possibles sur l'automate RGB-ab (après 50 itérations sur une grille initialisée aléatoirement de 200×200 cellules).

- **Règle RGB-31** : cette règle produit un résultat intéressant : elle se stabilise vers une configuration possédant plusieurs couleurs « entremêlées », mais lorsque l'on introduit une perturbation sur la grille (cellules défaillantes ou asynchronisme), celle-ci va vers un état uniforme, d'une seule couleur.
- **Règle RGB-52** : règle produisant des vagues perpétuelles de trois couleurs, bloquées par un nombre trop important de défaillance sur la grille.

Notons que plusieurs variations de ces règles existent, avec des comportements relativement similaires.

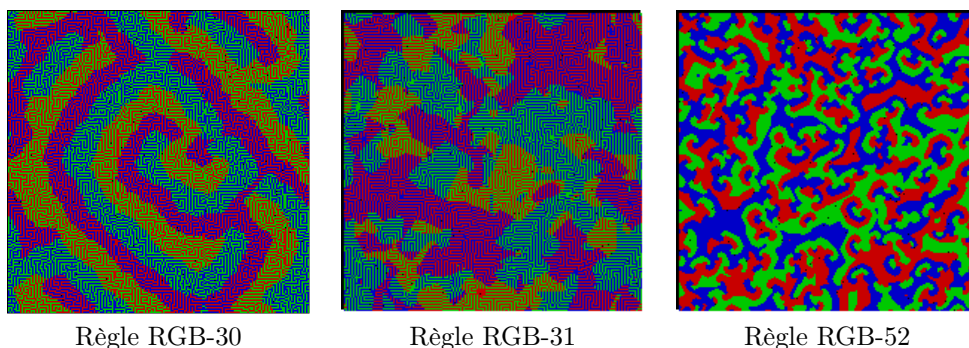


Fig. 17. Configurations après application des règles 3-1 et 5-2 sur des grilles de 200×200 cellules après 50 itérations.

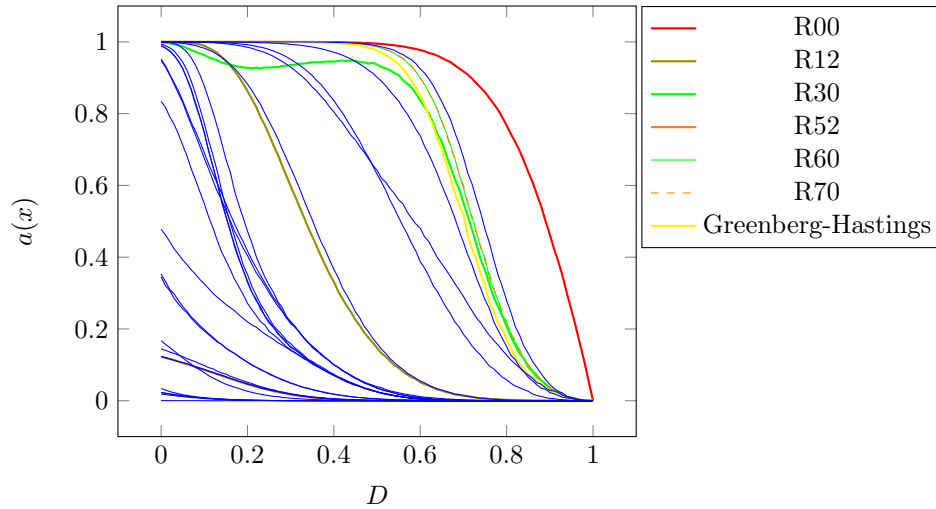


Fig. 18. Évolution de l'activité (cellules non défailtantes uniquement) en fonction de la densité de cellules défailtantes sur les 64 règles (tests effectués sur une moyenne de 15 expériences, après 10 itérations sur une grille de 200×200 cellules, initialisée aléatoirement).

Densités et activités

Sur l'analyse de l'activité en fonction de la densité de cellules défailtantes (figure 18), on constate que les différentes règles provoquent des variations assez différentes de l'activité en fonction de la densité de cellules mortes. Les règles *R60* et *R70* présentent une activité en fonction de la densité de cellules défailtantes très proche de la règle de Greenberg-Hastings.

6.2 Variantes de l'automate

Dans le but de simplifier le modèle RGB-ab présenté dans cette section, nous pouvons considérer deux variantes avec un seul paramètre au lieu de deux, et ne permettant aux couleurs R, G et B de ne « tourner » que dans un seul sens (R vers G, G vers B, B vers R).

Règle RGB-a

$$f : \begin{array}{l} Q \times \mathbb{N}^Q \longrightarrow Q \\ (q; n_R, n_G, n_B) \longmapsto \begin{cases} q^+ \text{ si } n_{q^+} > a \\ q \text{ sinon} \end{cases} \end{array}$$

Règle RGB-ai

$$f : \begin{array}{l} Q \times \mathbb{N}^Q \longrightarrow Q \\ (q; n_R, n_G, n_B) \longmapsto \begin{cases} q^+ \text{ si } n_{q^-} > a \\ q \text{ sinon} \end{cases} \end{array}$$

Exploration des règles

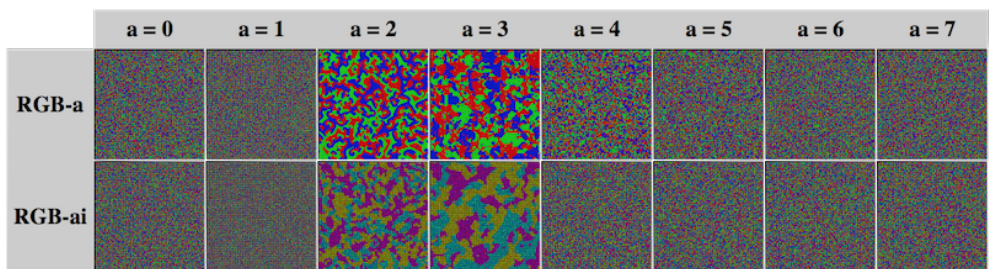


Fig. 19. Toutes les règles possibles sur les automates RGB-a et RGB-ai (après 50 itérations sur une grille initialisée aléatoirement de 200×200 cellules).

Ces deux nouveaux automates peuvent alors être paramétrés avec la valeur $a \in [0, 7]$. Expérimentalement, nous constatons que nous pouvons retrouver des comportements très proches des règles *RGB-31* et *RGB-52* observées précédemment :

- *RGB-a-2* : produit un résultat similaire à la règle *RGB-52*.

- *RGB-ai-2* : produit un résultat proche de la règle *RGB-31* lorsque l'on observe une configuration toutes les 6 itérations.
- *RGB-ai-3* : produit un résultat similaire à la règle *RGB-30* lorsque l'on observe une configuration toutes les 6 itérations.

Ces règles peuvent donc être de bonnes alternatives aux règles *RGB-ab* et ont l'avantage d'être plus simples que ces dernières.

Étude analytique de l'activité

L'activité correspond au nombre de cellules qui vont changer d'état lors de la prochaine itération sur le nombre de cellules au total. Nous pouvons estimer le nombre de cellules ayant eu un changement d'état en multipliant le nombre de cellules par la probabilité qu'une cellule change d'état. Nous avons donc, pour la règle *RGB-a-2* sur une grille de n cellules avec un taux de cellules défaillantes d .

$$P(\text{cellule dans l'état } q_i) = \frac{1-d}{3} = \rho \text{ (avec } q_i \in \{\mathbf{R}, \mathbf{G}, \mathbf{B}\})$$

$$\text{soit } I(c, x) = \begin{cases} 1 & \text{si } F(x)_c \neq x_c \\ 0 & \text{sinon} \end{cases}$$

$$\begin{aligned} a(x) &= \mathbb{E}\left[\sum_{c \in \mathcal{L}} \frac{I(c, x)}{|\mathcal{L}|}\right] \\ &= 1 - P(q_g > 2) \\ &= 1 - [P(q_g = 0) + P(q_g = 1) + P(q_g = 2)] \\ &= 1 - [(1 - \rho)^8 + \binom{8}{1}\rho(1 - \rho)^7 + \binom{8}{2}\rho^2(1 - \rho)^6] \\ a(x) &= 1 - (1 - \frac{1-d}{3})^8 - 8(\frac{1-d}{3})(1 - \frac{1-d}{3})^7 - 28(\frac{1-d}{3})^2(1 - \frac{1-d}{3})^6 \end{aligned}$$

La figure 49 (en annexe) montre l'évolution de la densité de cellules actives après une itération de la règle *RGB-a-2*, mesurée expérimentalement (en rouge) et calculée (en bleu). Cependant, l'hypothèse du champ moyen ne peut fonctionner que lors de l'initialisation, l'organisation nette en vagues produite par la règle *RGB-a-2* change fortement la répartition des couleurs sur la grille.

6.3 Obtenir un diagnostic décentralisé avec la règle *RGB-a-2*

Nous allons maintenant utiliser l'automate cellulaire *RGB-a-2* décrit précédemment dans le but d'obtenir un diagnostic décentralisé d'un certain taux de cellules défaillantes.

Diagnostic à partir de l'activité

Comme proposé dans la section 2.4, nous allons essayer de produire un état d'alerte spécifique qui se propagera à l'ensemble des cellules fonctionnelles (non défaillantes). Pour cela, nous apporterons trois modifications à la règle de transition locale :

1. Une cellule passe en état d'alerte q_a si au moins une de ses voisines est en état d'alerte (*propagation de l'alerte*).
2. Une cellule en état d'alerte reste en état d'alerte (*maintient de l'alerte*). Notons que nous ne considérons pas ici les défaillances, puisqu'une défaillance peut apparaître / disparaître indépendamment des règles de transition de l'automate).
3. Une troisième règle doit permettre à une cellule non défaillante de passer en état d'alerte d'après son observation locale (*déclenchement de l'alerte*).

Dans cet automate, le déclenchement d'alerte se fait à partir d'un critère d'activité comme avec le modèle de Greenberg-Hastings, puisque les sections précédentes ont montré un lien fort entre l'activité des cellules non défaillantes et le taux de cellules défaillantes de la grille. La figure 48 (en annexe) montre le résultat si nous décidons de passer toute cellule qui n'a pas changé d'état pendant plus de 100 itérations dans un état d'alerte. Cependant, une cellule isolée peut facilement rester non active pendant un certain temps avec un faible taux de cellules défaillantes, nous cherchons donc à identifier un petit groupe de cellules restées

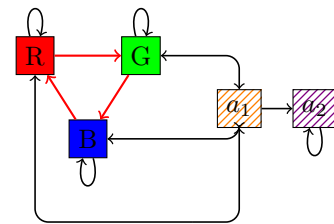


Fig. 20. Transitions entre états

longtemps sans activité (changement d'état) pour déclencher l'alerte. Pour cela, nous allons alors introduire un second état d'alerte, et redéfinir les règles de transition locale.

1. Une cellule colorée (q_R, q_G, q_B) passe dans l'état suivant (sens donné par les flèches rouges épaisses dans la figure 20 ci-contre) si la quantité de cellules voisines dans l'état suivant est supérieure à 2. Sinon, elle ne change pas de couleur.
2. Une cellule colorée passe en état d'alerte q_a si elle n'a pas changé d'état depuis plus de 100 itérations.
3. Une cellule en état d'alerte q_a passe en état d'alerte q_b si elle a au moins 4 voisines en état d'alerte q_a , sinon elle reprend une couleur (état q_R, q_G ou q_B choisi aléatoirement).
4. Une cellule en état d'alerte q_b reste dans ce même état. Une cellule ayant au moins une voisine en état d'alerte q_b passe en état d'alerte q_b (transition non représentée sur la figure 20 par soucis de lisibilité).

6.4 Robustesse du diagnostic

Influence de la taille de la grille

La figure 51 (en annexe) présente les résultats de tests effectués sur des grilles de différentes tailles initialisée aléatoirement avec différents taux de cellules défaillantes.

Sur les grilles de taille 500×500 , on constate que presque aucune des configurations testées n'est en alerte pour un taux de cellules défaillantes inférieur à 5%, et que toutes les grilles avec un taux supérieur à 10% sont en alerte après 1000 itérations. La taille de la grille semble également avoir une légère influence : ce résultat n'est pas surprenant, dans la mesure où la probabilité de rencontrer une structure de cellules défaillantes bloquant suffisamment l'activité pour déclencher un état d'alerte augmentera avec la taille de la grille. Ces résultats sont prometteurs dans l'optique d'obtenir un diagnostic décentralisé d'un pourcentage de cellules défaillantes, nous allons alors pouvoir tester sa robustesse, notamment aux asynchronismes et changements de topologie, et essayer de faire varier le seuil critique de cellules défaillantes à détecter, notamment avec la règle de passage du premier au second état d'alerte.

α -synchronisme

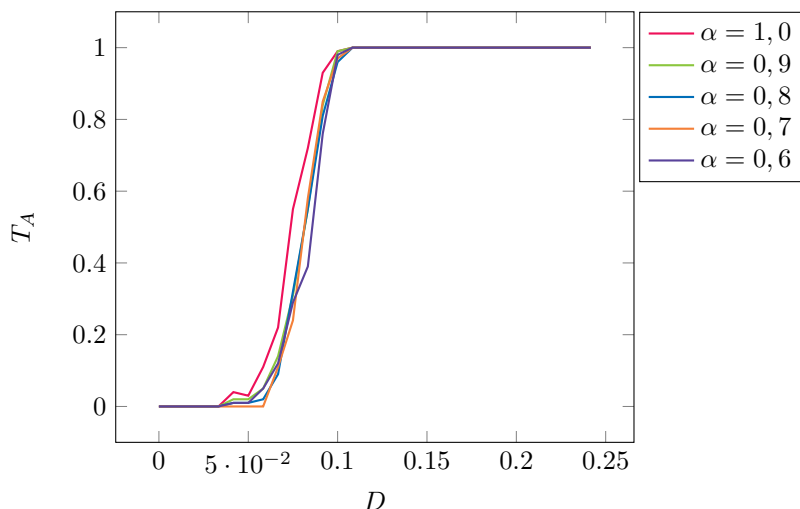


Fig. 21. Taux de configurations en alerte en fonction du taux de cellules défaillantes D pour différents α -synchronismes (tests effectués sur des grilles de 100×100 cellules après 500 itérations).

La figure 21 montre le taux de configurations en alerte pour différents α -synchronismes. Le modèle *RGB-a-2* se montre, contrairement au modèle de *Greenberg-Hastings*, très robuste à l' α -synchronisme pour ce système de diagnostic décentralisé.

Paramétrage du diagnostic

Le modèle *RGB-a* semble plus robuste à l'asynchronisme que le modèle de *Greenberg-Hastings* étudié dans la section précédente. Cependant, ce dernier n'est pas probabiliste, et nous n'avons pour l'instant

pas de paramètre à partir duquel nous pouvons définir le taux de cellules défaillantes à diagnostiquer, de la même façon que nous le faisons avec la probabilité d'excitation p_τ .

Le diagnostic avec la règle *RGB-a-2* est relativement sensible à la présence de cellule défaillantes (celui-ci déclenchant l'état d'alerte pour 5 à 10% de cellules défaillantes). Idéalement, nous aimerions donc pouvoir, avec un paramètre, éventuellement augmenter le seuil de déclenchement de l'alerte en fonction de la densité de cellules défaillantes sur la grille. Pour cela, nous allons ajouter une probabilité ϵ d'appliquer une règle de transition « moins stricte », permettant d'alors d'augmenter l'activité sur la grille, et donc retarder le déclenchement de l'alerte par rapport à la densité de cellules défaillantes. Soit f la nouvelle règle :

$$f : \quad Q \times \mathbb{N}^Q \longrightarrow Q$$

$$(q, (n_R, n_G, n_B)) \longmapsto \begin{cases} q^+ & \text{si } n_{q^+} > 2, \\ q^+ & \text{si } n_{q^+} > 1 \text{ avec une probabilité } \epsilon, \\ q & \text{sinon.} \end{cases}$$

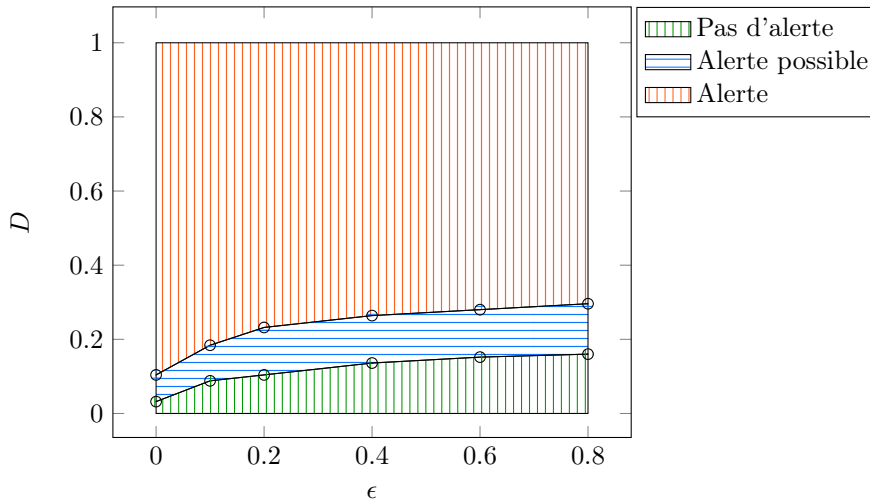


Fig. 22. Déclenchement d'alerte en fonction de la probabilité d'excitation ϵ et de la densité de cellules défaillantes D (mesures effectuées sur 3000 grilles de 100×100 cellules initialisées aléatoirement après 500 itérations).

Cette nouvelle règle ne convient pas pour adapter efficacement le taux de cellules défaillantes à diagnostiquer ; la probabilité ϵ n'affecte que légèrement le seuil de déclenchement de l'état d'alerte.

6.5 Test de diagnostic avec différentes règles RGB-ab

La figure 50 (en annexe) montre le résultat du diagnostic avec différents paramètres du modèle *RGB-ab* (les 64 règles ne sont pas présentes sur ce graphique parce qu'une partie d'entre elles donnent des résultats similaires ou non pertinents, pas exemple alerte systématiquement présente ou absente). On constate que de nombreuses règles donnent des résultats similaires (par exemple *RGB-ab-12*, *RGB-ab-21* et *RGB-ab-52*, proches de *RGB-a-2*), et certaines sont décalées (par exemple *RGB-ab-11*). Enfin, certaines règles donnent un résultat surprenant, comme les règles *RGB-ab-13* et *RGB-ab-51* autour de $D = 0,2$. Ce phénomène semble s'expliquer par la présence, pour certaines règles, de deux phases aux comportements différents, co-existantes sur la grille lorsque le taux de cellules défaillantes avoisine les 20% ; il serait alors intéressant de chercher à comprendre ce qui peut expliquer une telle différence de comportement sur des règles pourtant très proches (par exemple *RGB-a-51* et *RGB-a-52*).

6.6 Observations sur les règles

Vagues RGB-52 et RGB-a-2

Nous observons sur les règles *RGB-52* et *RGB-a-2* des spirales qui se forment sur des configurations tirées aléatoirement et semblent ensuite presque immuables. Ces spirales propagent des vagues de couleurs sur le reste de la grille et absorbent les vagues des autres spirales, sans changer de place, sur un cycle de 24 itérations. Ces spirales permettront d'obtenir, dans de nombreux cas, des cycles de

24 itérations sur les configurations. Trois types de structures sont présents sur les règles *RGB-52* et *RGB-a-2* :

1. *Les spirales* décrites ci-dessus, qui restent fixes.
2. *Les doubles spirales* pouvant se déplacer sur la grille. Lorsqu'elles ont une taille importante, elles peuvent détruire les spirales qu'elles rencontrent.
3. *Les bassins* entre les spirales, « absorbants » les vagues de couleur.

7 Règle semi-chaotique

Les méthodes proposées dans les parties précédentes permettent un diagnostic décentralisé du nombre de cellules défaillantes, mais le diagnostic à l'aide du modèle de *Greenberg-Hastings* est modifié en fonction du taux d' α -synchronisme, et celui avec le modèle *RGB-a-2* ne permet pas de détecter un taux élevé de cellules défaillantes. Nous aimerions alors trouver un modèle à la fois robuste à l' α -synchronisme et capable de détecter un taux de défaillance faible comme élevé. Toujours dans l'idée d'obtenir un diagnostic à partir de l'activité, nous allons explorer d'autres familles de règles fonctionnant de la manière suivante : nous allons coupler une règle chaotique exécutée avec une probabilité $1 - \lambda$, avec une règle « tout zéro » avec une probabilité λ . Dans cette partie, nous considérerons un espace d'états $Q = \{0, 1\}$ et un voisinage de von Neumann. Plus formellement :

$$\begin{aligned} f_1 & : & Q \times \mathbb{N}^Q & \longrightarrow & Q \\ f_2 & : & Q \times \mathbb{N}^Q & \longrightarrow & Q \\ f_\lambda & : & Q \times \mathbb{N}^Q & \longrightarrow & Q \\ & & (q, (n_0, n_1)) & \longmapsto & \begin{cases} f_1(q, (n_0, n_1)) \text{ avec une probabilité } \lambda, \\ f_2(q, (n_0, n_1)) \text{ avec une probabilité } (1 - \lambda). \end{cases} \end{aligned}$$

Nous pouvons considérer différentes fonctions f_1 , mais l'idée ici est d'utiliser une fonction de transition chaotique quelconque. Pour f_2 , nous pouvons utiliser une fonction « tout-zéro » ou une fonction de majorité, par exemple. Dans un premier temps, nous utiliserons les fonctions suivantes :

$$\begin{aligned} f_1 & : & Q \times \mathbb{N}^Q & \longrightarrow & Q \\ & & (q, (n_0, n_1)) & \longmapsto & \begin{cases} 1 \text{ si } q = 1 \text{ et } n_1 \in [a, b] \text{ (règle de survie),} \\ 1 \text{ si } q = 0 \text{ et } n_1 \in [c, d] \text{ (règle de naissance),} \\ 0 \text{ sinon .} \end{cases} \\ f_2 & : & Q \times \mathbb{N}^Q & \longrightarrow & Q \\ & & (q, (n_0, n_1)) & \longmapsto & 0 \end{aligned}$$

Pour comparer plusieurs règles chaotiques, une règle avec l'intervalle de survie $[a, b]$ et l'intervalle de naissance $[c, d]$ sera notée *Rabcd*.

7.1 Règle et voisinage

Diminuer le paramètre λ permet de réduire l'activité en fonction de la densité de cellules défaillantes. Idéalement, nous souhaitons donc avoir une règle ayant une activité non nulle jusqu'à une densité de cellules défaillantes D élevée. Augmenter la taille du voisinage (en passant du voisinage de von Neumann à celui de Moore par exemple) est susceptible de permettre des règles chaotiques avec une activité plus élevée pour une densité de cellules défaillantes D donnée (puisque le nombre de voisins non défaillants est plus élevé en moyenne avec un voisinage plus grand pour une densité de cellules défaillantes fixée).

La figure 23 montre l'évolution de l'activité pour différentes règles avec le voisinage de Moore et de von Neumann pour $\lambda = 0, 7$ (en pointillés) et $\lambda = 0, 9$.

Pour choisir une règle *Rabcd*, en respectant $a < b < |V|, c < d < |V|$, nous avons 225 choix possibles pour le voisinage de von Neumann et 2025 choix possibles pour celui de Moore.

La figure 47 montre l'évolution de toutes les règles parmi les 225 possibles avec le voisinage de von Neumann (79 règles au total), qui ont une activité non nulle pour $D < 1$, et nulle à partir d'un certain seuil D_c . Toutes ces règles peuvent être utilisées pour le diagnostic, mais nous choisirons de préférence une règle pour laquelle le seuil D_c est élevé - celui-ci pouvant ensuite être diminué avec le paramètre λ - et dont l'activité avant le seuil est également élevée.

L'aspect moins lisse des courbes de la figure 47 par rapport à la figure 23 s'explique par la taille des grilles de cellules plus petite (100×100 cellules contre 400×400 cellules pour la figure 23), le nombre

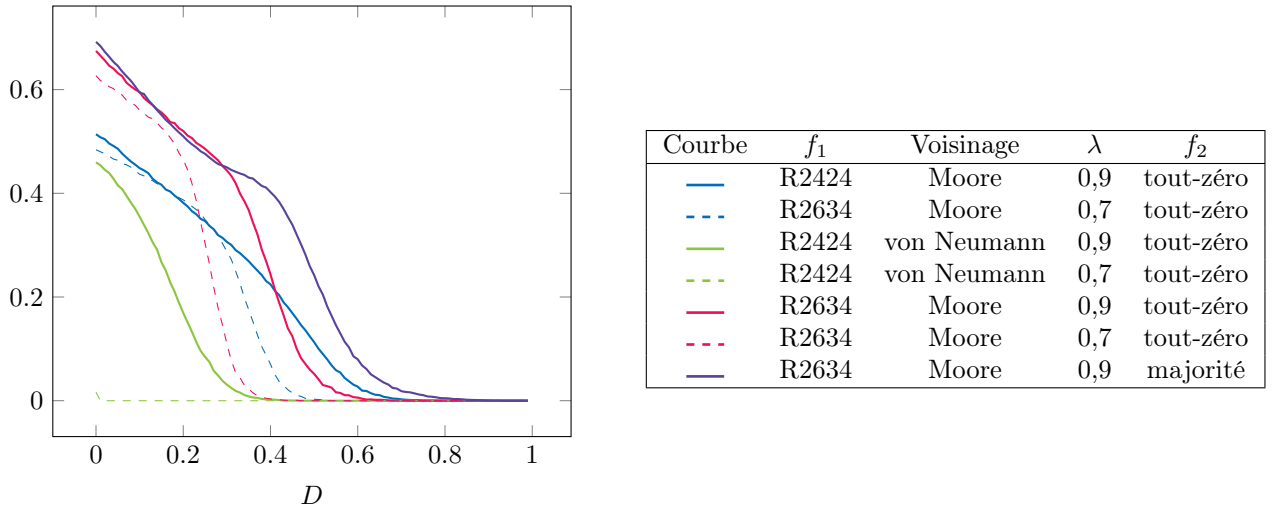


Fig. 23. Évolution de l’activité en fonction du taux de cellules défaillantes (mesures effectuées sur des configurations de 400×400 cellules après 500 itérations).

de règles testées étant bien plus important, des grilles plus petites augmentent fortement la vitesse d’exécution des tests.

Sur la figure 47 (en annexe), on constate un espace de règles répondant de manière assez semblable aux règles *RGB-a* et *RGB-ab* en terme d’activité en fonction de la densité de cellules défaillantes (voir figure 18). Nous pouvons, parmi ces règles, choisir une règle ayant une activité élevée pour D faible et nulle à partir d’un seuil D_c important. Pour autant, il faut que l’activité diminue suffisamment lorsque D augmente pour éviter d’empêcher la propagation des états d’alerte si D_c est trop élevé.

La règle *R0301* (courbe épaisse vert émeraude sur la figure 47) est celle ayant la plus forte activité en fonction de D pour le voisinage de von Neumann. Nous proposons également d’étudier la règle « ou » (*R1818*), donnant toujours 1 s’il y a au moins une cellule voisine vivante, zéro si toutes les voisines sont à zéro (ou défaillantes), qui pourrait présenter l’intérêt de lier le nombre de cellules vivantes de la grille à la densité de cellules défaillantes en plus de l’activité.

7.2 Test du diagnostic

Nous pouvons maintenant mettre en place un diagnostic à l’aide d’un automate cellulaire à deux états (le second mesurant l’activité) fonctionnant sur le même principe que celui utilisé avec le modèle de *Greenberg-Hastings* dans la partie 5 et *RGB-a-2* dans la partie 6. Nous utiliserons différentes règles pour f_2 de façon à observer plus précisément les variations engendrées au niveau du taux de cellules défaillantes détectées. La figure 23 nous montre que l’activité est plus importante sur une partie de la courbe lorsque nous utilisons la règle de majorité par rapport à la règle « tout-zéro ». Ce résultat n’est pas forcément surprenant puisque la règle de majorité donne une réponse variable en fonction du voisinage, nous pouvons penser qu’elle provoque légèrement plus de variations que la règle « tout-zéro », selon la valeur assignée au paramètre λ .

La règle « ou » permet de détecter des taux de cellules défaillantes élevés, mais le diagnostic s’avère moins précis que pour d’autres règles testées (l’écart entre les densités de défaillance D pour lesquelles il n’y a aucune alerte et celles pour lesquelles toutes les configurations sont en alerte est plus important).

8 Comparaison des méthodes basées sur l’activité

Nous avons maintenant trois automates cellulaires permettant d’effectuer le diagnostic d’un taux donné de cellules défaillantes à partir de l’activité des cellules. De façon à pouvoir comparer les trois méthodes proposées, nous allons maintenant les tester sur des cas concrets de façon à évaluer comparativement leurs performances.

8.1 Faible taux de défaillance

Nous allons maintenant comparer les trois méthodes de diagnostic pour détecter un taux de cellules défaillantes de 10% sur une grille de 100×100 cellules. Les trois modèles seront utilisés avec $|Q'| = 50$ et

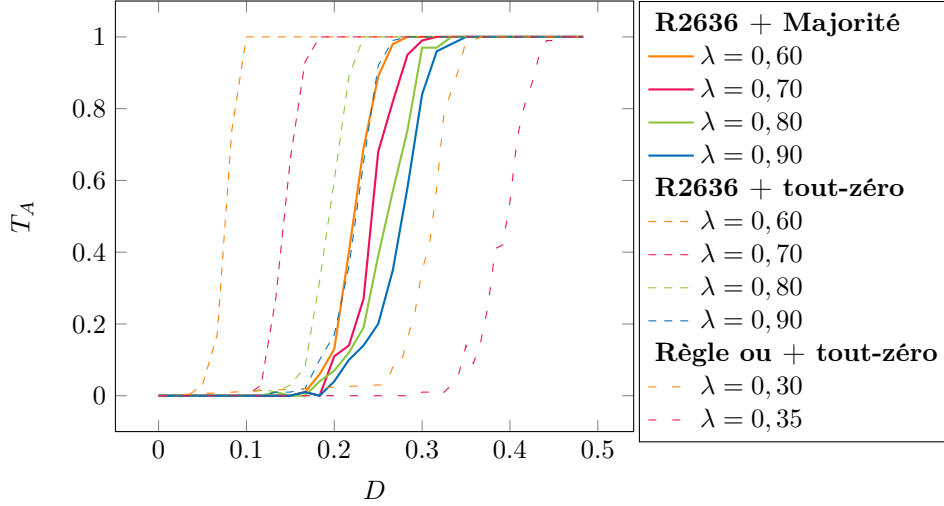


Fig. 24. Évolution du taux de configurations en alerte T_A en fonction du taux de cellules défaillantes D pour différentes valeurs de λ et différentes fonctions f_2 , avec $|Q'| = 50$ sur la règle *R2634* (tests effectués après 500 itérations sur des grilles de 100×100 cellules initialisées aléatoirement, pour 100 configurations par taux de défaillance).

des valeurs (p_τ , λ et ϵ selon le modèle) permettant de diagnostiquer environ 10% de cellules défaillantes.

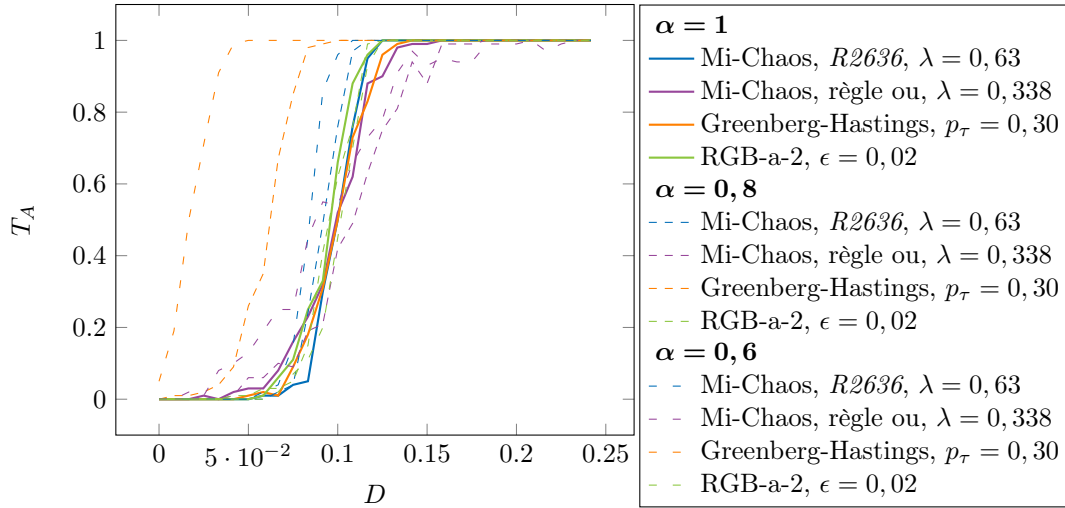


Fig. 25. Évolution du diagnostic pour *Greenberg-Hastings*, *RGB-a-2* et *Mi-Chaos* (tests effectués après 500 itérations sur des grilles de 100×100 cellules initialisées aléatoirement). Les paramètres p_τ , λ et ϵ ont été déterminés avec les figures 14 et 22 et adaptés expérimentalement pour superposer au mieux les trois courbes (pour $\alpha = 1$) et ainsi pouvoir comparer les déclenchements d'alertes de chacun des trois modèles.

La figure 25 nous montre que les trois méthodes de diagnostic ont des résultats fortement similaires pour $\alpha = 1$. Cette expérience n'apporte donc pas d'élément concret permettant de privilégier une méthode particulière par rapport aux autres.

8.2 Faible taux de défaillance avec asynchronisme

Nous allons maintenant reproduire la même expérience pour différents taux de synchronisme. Nous souhaitons idéalement que le taux de cellules défaillantes provoquant le diagnostic reste inchangé pour différents α -synchronismes. Sur la figure 25, on constate que pour $\alpha = 0,8$ et $\alpha = 0,6$, le diagnostic avec le modèle de *Greenberg-Hastings* est fortement affecté, un peu moins pour le modèle *Mi-Chaos*, tandis que la règles RGB se montre très robuste à l' α -synchronisme.

8.3 Taux élevé de cellules défaillantes avec asynchronisme

Nous allons maintenant reproduire l'expérience précédente pour diagnostiquer, cette fois, un taux de 50% de cellules défaillantes. Nous utiliserons *RGB-a-0* pour le modèle *RGB*, ainsi que les autres modèles : *Greenberg-Hastings* et *Mi-Chaos* (règle ou). Comme dans l'expérience précédente, nous adapterons p_τ et λ pour aligner au mieux les taux de cellules défaillantes D déclenchant l'alerte, et nous ferons varier le taux de synchronisme α .

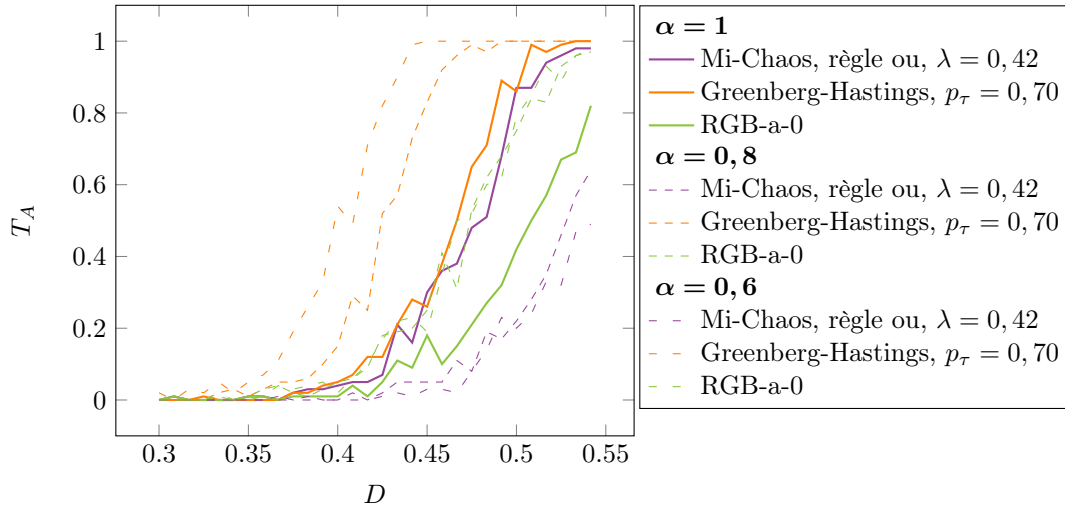


Fig. 26. Évolution du diagnostic pour *Greenberg-Hastings*, *RGB-a-2* et *Mi-Chaos* (tests effectués après 500 itérations sur des grilles de 100×100 cellules initialisées aléatoirement). Les paramètres p_τ , λ et ϵ ont été déterminés avec les figures 14 et 22 et adaptés expérimentalement pour superposer au mieux les trois courbes (pour $\alpha = 1$) et ainsi pouvoir comparer les déclenchements d'alertes de chacun des trois modèles.

On constate que les trois règles se comportent assez différemment face à l' α -synchronisme : la règle *RGB-a-0* se redresse lorsque l'on ajoute de l'asynchronisme, mais le passage de $\alpha = 0,8$ à $\alpha = 0,6$ n'a pas d'effet sur le déclenchement de l'alerte. Le modèle de *Greenberg-Hastings* devient, comme dans l'expérience précédente, plus sensible et déclenche une alerte pour un taux de cellules défaillantes D moins élevé, alors que c'est l'inverse qui se produit pour la règle semi-chaotique « ou ».

8.4 Temps de réaction du diagnostic

Nous avons jusqu'ici mesuré, dans chaque expérience, le nombre de configurations ayant propagé l'état d'alerte après 500 itérations à partir d'une densité de cellules défaillantes D fixée. Nous allons maintenant mesurer le nombre d'itérations nécessaires au déclenchement de l'alerte pour différentes densités de cellules défaillantes données.

Sur la figure 52 (en annexe) on constate que le modèle de *Greenberg-Hastings* nécessite de nombreuses itérations pour que toutes les configurations soient en alerte pour $D = 0,1$ (et certaines ne le sont toujours pas après 1000 itérations). Cependant, si la probabilité d'excitation p_τ est diminuée, ou la densité de cellules défaillantes D augmentée, alors l'alerte se déclenche plus vite et le résultat sera plus proche des résultats obtenus avec les autres modèles. De la même façon, les règles *RGB-a-2* et *Mi-Chaos* peuvent déclencher l'alerte plus lentement que dans cette expérience si les paramètres D et λ ou ϵ sont modifiés.

Il est donc difficile de comparer les différents modèles à partir du nombre d'itérations avant le déclenchement de l'alerte parce que celui-ci dépend fortement de la densité de cellules défaillantes D , de α et du paramètre du modèle (λ , ϵ ou p_τ selon la règle appliquée). Nous devons cependant prendre en compte que lorsque nous sommes proche de la valeur de D provoquant l'alerte, le nombre d'itérations nécessaires au déclenchement de celle-ci sera plus important que pour une valeur de D plus importante.

8.5 Tableau comparatif

Nous pouvons maintenant comparer les trois modèles étudiées sur plusieurs critères, tels que la résistance à l' α -synchronisme et les taux de cellules défaillantes détectables.

Modèle	Greenberg-Hastings	RGB	R2636	Ou
Résistant à l' α -synchronisme	✘ Grand décalage	✓ Oui	✘ Décalage	✘ Moins précis
Fonctionne pour D élevé	✓ Oui	✓ Oui	✘ Non	✓ Oui
Intervalle de D détectable	$[0, 0.7]^1$	$[0, 0.5]$	$[0, 0.3]$	$[0, 0.7]^1$

8.6 Problème de la diffusion d'un état d'alerte

Dans ces trois modèles, le second état d'alerte se propage sur toute la grille dès son apparition. Ce fonctionnement permet un consensus aisé sur l'état d'alerte ou non de la grille, mais pose problème en vue d'obtenir une méthode conforme à la formalisation donnée. En effet, cet état se diffusant sur toute la grille, il suffit qu'il apparaisse à un endroit donné pour déterminer l'évolution de toute la configuration. Ainsi, si un motif de taille finie produit une baisse d'activité suffisamment importante pour déclencher l'alerte, alors la présence de ce motif assure l'évolution de la configuration entière vers l'état d'alerte. De ce fait, la taille de la grille affecte le diagnostic puisque les motifs provoquant l'alerte ont une probabilité d'apparition proportionnelle à la taille de la grille.

9 Diagnostic avec passage à l'échelle

Nous avons vu que la propagation d'un état d'alerte posait problème en tant que moyen d'obtention de consensus sur l'état de la grille. Nous allons maintenant envisager une méthode de diagnostic ne présentant pas ce défaut. Le modèle présenté ici a été envisagé après différents essais sur grille infinie présentés dans l'annexe B.

Idéalement, nous souhaitons obtenir une règle de transition possédant les caractéristiques suivantes :

- Lorsque toutes les cellules voisines sont en alerte ou défaillantes, on passe en état d'alerte.
- Si toutes les cellules voisines sont dans l'état normal, alors on passe dans l'état normal.
- Sinon, on passe en état d'alerte avec une probabilité d'autant plus grande que le nombre de cellules voisines en état d'alerte ou défaillant est grand.

Expérimentalement, nous constatons que les combinaisons linéaires ne permettent pas d'obtenir une transition de phase pour un taux fixé de cellules défaillantes. L'annexe B présente une règle remplissant les conditions citées précédemment. La règle présentée ici est inspirée du modèle d'Ising, et possède un paramètre λ permettant de paramétrer le taux de cellules défaillantes à détecter.

$$f : Q \times \mathbb{N}^Q \longrightarrow Q$$

$$\begin{array}{ll} \text{Si } n_N = 0 & f(q; n_A, n_N, n_D) = A \\ \text{Sinon} & f(q; n_A, n_N, n_D) = \begin{cases} N \text{ avec probabilité } \frac{e^{\lambda \frac{n_N}{S}}}{e^{\lambda \frac{n_N}{S}} + e^{\lambda \frac{n_A}{S}} + e^{\lambda \frac{n_D}{S}}}, \\ A \text{ sinon.} \end{cases} \end{array}$$

avec $S = n_N + n_A + n_D$.

Le modèle d'Ising est un modèle de physique statistique disposant de particularités intéressantes : il présente une transition de phase avec des exposants critiques différents de ceux calculés à l'aide de la théorie du champ moyen [Vel09].

La figure 27 montre l'évolution de la densité relative de l'état d'alerte au cours du temps pour différentes densités de cellules défaillantes initiales. La figure 28 montre le taux de configuration en alerte.

- ✓ **Taille de la grille** : ce modèle est, comme le précédent, robuste au changement d'échelle.
- ✓ **Nombre d'itérations** : le nombre d'itérations nécessaires à l'obtention d'un consensus est proche de celui des modèles précédents (environ 2000 itérations proche du seuil critique).
- ✓ **α -synchronisme** : ce modèle se montre également robuste à l' α -synchronisme mais avec une variation toutefois plus marquée que le modèle précédent.

Les modèles proposés dans les parties précédentes possèdent divers problèmes : les diagnostics basés sur la propagation d'un état d'alerte ne sont pas robustes aux changements d'échelle, le comportement du modèle de *Greenberg-Hastings* est fortement modifié par l'introduction d'asynchronisme, et certaines

1. la borne supérieure est imposée par le passage du premier au second état d'alerte. Pour diagnostiquer un taux de cellules défaillantes $D > 0,7$, il faudrait modifier cette règle. Les modèles comparés ici ne sont donc pas en cause sur cette restriction.

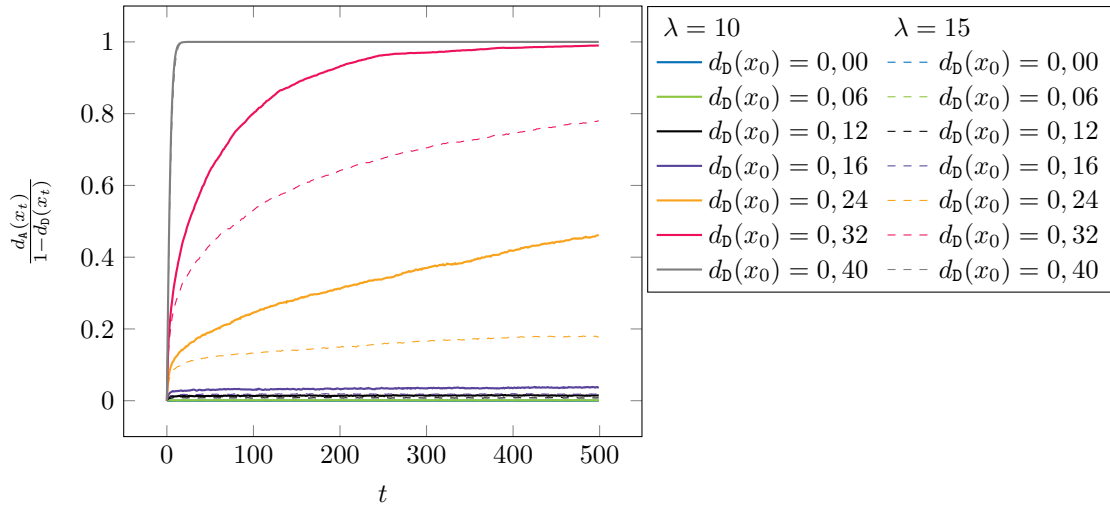


Fig. 27. Évolution de la densité de l'état A au cours du temps pour la règle d'Ising (tests effectués sur des grilles de 200×200 cellules, moyenne mesurée sur 5 tests).

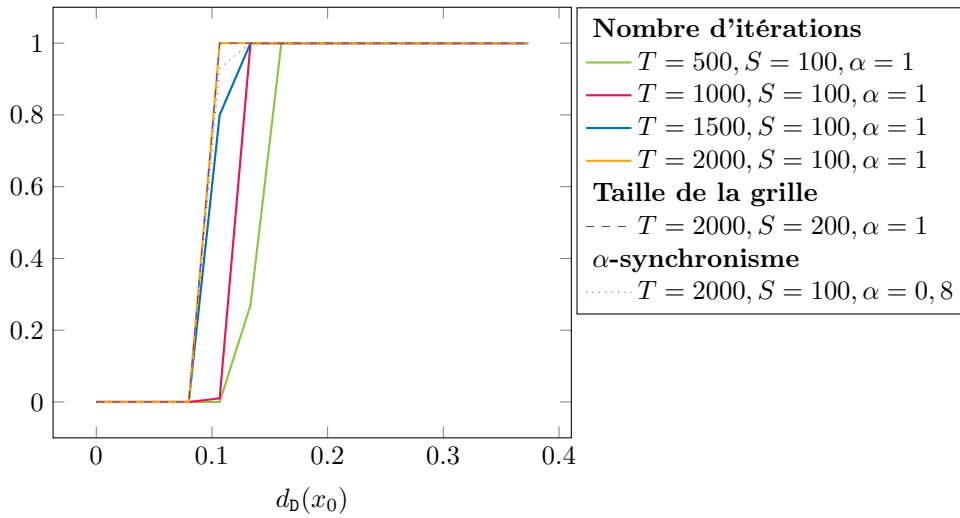


Fig. 28. Taux de configurations en alerte ($d_A(x_t) = 1 - d_D(x_t)$) après T itérations sur des grilles de $S \times S$ cellules pour la règle d'Ising pour $\lambda = 10$.

règles semi-chaotiques ne permettaient pas la détection d'un taux élevé de cellules défaillantes. En revanche, les deux modèles précédents ne présentent pas ces problèmes et sont donc de bons candidats pour la problématique du diagnostic décentralisé d'un certain taux de cellules défaillantes.

Nous allons maintenant faire un bilan sur les différentes méthodes proposées, en vue de comparer les résultats obtenus avec les objectifs fixés initialement, ainsi que réfléchir à la mise en place de ces modèles dans des cas concrets.

Apparition dynamique des défaillances

Si les défaillances apparaissent progressivement à un rythme régulier, et continuent d'apparaître une fois le seuil critique dépassé, nous pouvons nous attendre à ce que le temps de réaction du diagnostic soit d'autant plus court que les défaillances apparaissent rapidement. Pour évaluer ce rapport entre la vitesse d'apparition des défaillances et le temps de réaction du diagnostic, l'expérience présentée figure 30 montre l'évolution de la densité de l'état d'alerte pour différentes évolutions de la densité de cellules défaillantes. La figure 31 donne le temps de réaction du diagnostic en fonction de la vitesse d'apparition des cellules défaillantes.

Ce modèle ouvre des perspectives intéressantes pour un diagnostic décentralisé. Nous n'avons pas eu le temps d'explorer plus amplement son comportement, mais les premiers résultats sont prometteurs concernant notre problématique.

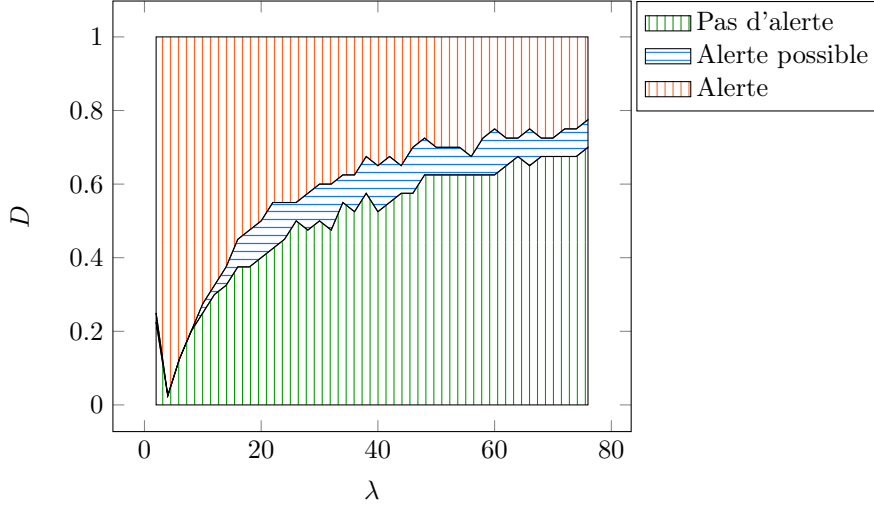


Fig. 29. Déclenchement d’alerte (toutes les cellules non défaillantes en état d’alerte) après 3000 itérations en fonction de λ (mesures effectuées pour 10 tests pour 20 taux de défaillances et 20 valeurs de λ sur des grilles de 100×100 cellules).

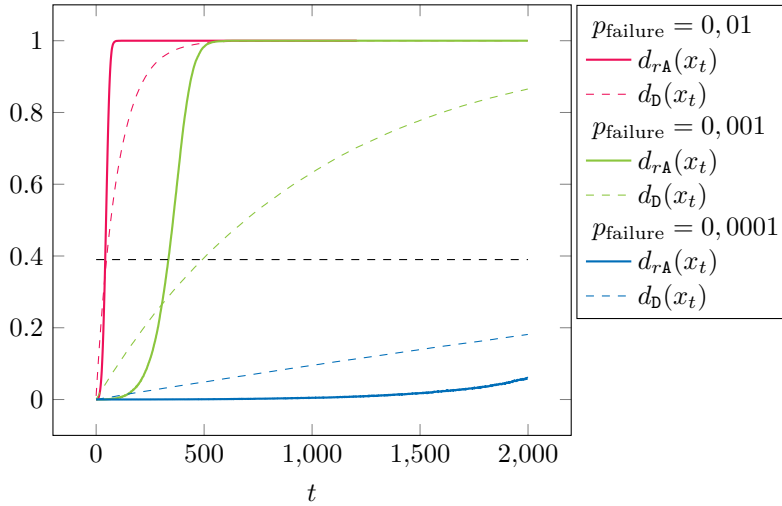


Fig. 30. Évolution des densités relatives de cellules en état d’alerte et des densités de cellules en état défaillant en fonction du nombre d’itérations pour $\lambda = 20$ (tests effectués sur des grilles de 400×400 cellules).

10 Bilan et applications concrètes

Nous allons maintenant voir comment nous pouvons implémenter les méthodes de diagnostic développées dans les sections précédentes pour les utiliser dans des cas concrets, mais aussi comparer les résultats obtenus avec les objectifs fixés lors de la formalisation du problème dans la partie 2.6.

10.1 Diagnostic avec passage à l’échelle

Dans la partie 9, nous avons vu deux modèles pouvant **répondre précisément au problème formalisé dans la partie 2.6**. En effet, pour ces modèles, la densité relative de cellules en état d’alerte (parmi les cellules non défaillantes) est égale à 1 au delà d’un seuil donné de cellules défaillantes ρ , et strictement inférieure sinon. Le modèle d’*Ising-Stabilisé* permet également de paramétrer le seuil ρ avec le paramètre λ , et se montre robuste à la fois aux changements d’échelle et à l’asynchronisme.

Cependant, en plus de répondre au problème donné, nous aimerions également que ce diagnostic soit utilisable concrètement, par exemple sur un réseau de capteur. Pour cela, nous devons encore faire face à deux contraintes supplémentaires :

- Dans le cas d’un réseau de capteur, la topologie du réseau est irrégulière, contrairement aux

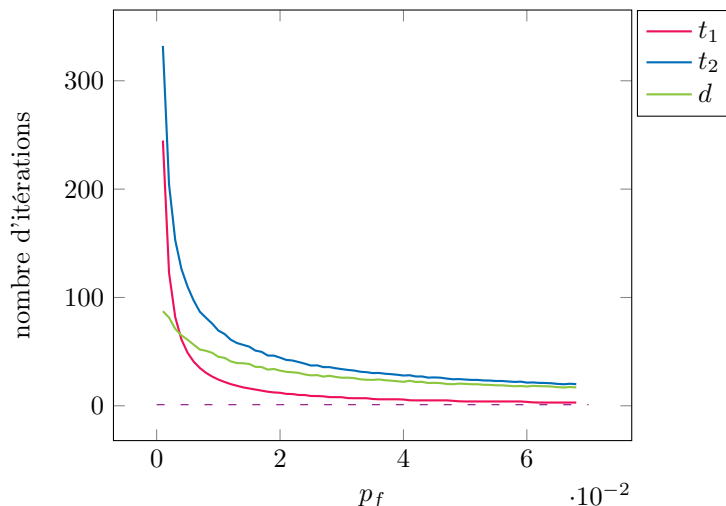


Fig. 31. Durée du diagnostic d en fonction de la probabilité de défaillance p_f . Mesures effectuées avec $\rho = 0,22$ sur des grilles de 200×200 cellules sur une moyenne de 20 expériences pour 100 probabilités de défaillance.

automates cellulaires utilisés jusqu'ici. Nous devons donc nous assurer que le diagnostic reste fonctionnel avec une topologie irrégulière.

- L'état d'alerte se caractérise par l'absence de cellules en état normal sur la grille (toutes les cellules sont alors défaillantes ou en alerte). Cependant, il n'est pas aisé de vérifier cette donnée sans vérifier chacune d'entre elle, nous allons donc chercher à produire une méthode permettant d'obtenir une probabilité que le système soit en alerte en ne regardant qu'un petit nombre de cellules.

Simulation en topologie irrégulière

Pour simuler le modèle sur une topologie irrégulière, nous allons utiliser un voisinage probabiliste spécifique. Nous utiliserons le voisinage suivant : chaque cellule du voisinage de Moore de rayon 2 est sélectionnée avec une probabilité de 0,35 (pour arriver à une moyenne de 9 voisines par cellules comme dans le cas du voisinage de Moore de rayon 1). Ce voisinage permet de créer des variations importantes de topologie, de façon à ne pas demander de modification des paramètres des différents modèles (le nombre de voisins par cellule et les connexion sont ainsi variables sur la grille, mais fixe au cours du temps). Les résultats de l'exécution des différents modèles sur ce voisinage sont résumés dans le tableau ci-dessous.

Obtenir une estimation de l'état de la grille

Les modèles sans propagation d'alerte sont robustes aux changements d'échelle, mais font apparaître l'état d'alerte avant que le seuil critique de cellules défaillantes soit atteint. Déterminer si une configuration donnée est en alerte ou non sans en parcourir toutes les cellules est alors plus difficile que dans le cas de la propagation. En revanche, il est envisageable d'estimer l'état d'une configuration en ne regardant l'état que d'un petit nombre de cellules choisies au hasard. Une configuration en alerte ne possédant aucune cellule dans l'état normal, trouver parmi les cellules choisies une cellule dans l'état normal nous permet donc de savoir que la grille n'est pas en alerte, et donc que le seuil critique de défaillances n'a pas encore été atteint (ou plutôt, n'avait pas été atteint récemment, le diagnostic n'étant pas instantané). Si nous ne trouvons aucune cellule en état normal, la probabilité que la configuration soit en alerte est alors d'autant plus grande que le nombre de cellules choisies aléatoirement est grand.

10.2 Diagnostic à partir de l'activité sur grille finie

Les différentes méthodes de diagnostic à partir de l'activité ne permettent pas de répondre strictement au problème énoncé en partie 2.6 sur une grille de taille infinie à cause de la propagation automatique de l'état d'alerte. En revanche, ces méthodes possèdent un autre avantage non négligeable dans le cas d'une grille finie : ils permettent de s'assurer de l'absence totale de l'état d'alerte sous le seuil critique ρ , contrairement aux méthodes sans propagation qui ne peuvent garantir qu'une densité d'alerte inférieure

à 1 en dessous du seuil. Cette fois, la contrainte supplémentaire à prendre en compte est donc le cas d'une grille à la topologie irrégulière.

10.3 Bilan

	<u>Modèle</u>	<u>Grille infinie</u>	<u>Asynchronisme</u>	<u>Topo. irrégulière</u>	<u>Alerte</u>
Propagation d'alerte	Greenberg	✗ Non	✗ Non		✓ Tout ou rien
	Hastings				
	RGB		✓ Oui		
	MiChaos		✗ Non		
Classification biaisée	Carrés	✓ Oui			✗ Progressif
	Ising				

Selon les cas, deux modèles seront alors préférés aux autres :

- **Le modèle RGB** s'avère robuste à l'asynchronisme, supporte les topologies irrégulières et à l'avantage de n'utiliser l'état d'alerte que si le seuil ρ est atteint. En revanche, il ne pourra pas fonctionner sur des grilles de taille variable ou infinies.
- **Les modèles probabilistes Carrés et la règle d'Ising-Stabilisé** sont les seuls parmi ceux explorés à répondre strictement à la problématique donnée dans la section 2.6. Ils sont robustes au changement d'échelle, à l'asynchronisme et aux topologies irrégulières. En revanche, l'état d'alerte apparaît progressivement en fonction du taux de cellules défaillantes, le diagnostic est donc plus difficile.

11 Évolution de Fiatlux

Ce stage a également été l'occasion de mettre à jour et faire évoluer le logiciel de simulation d'automates cellulaires *FiatLux*¹⁴, développé par Nazim Fatès. Les modifications apportées concernent l'implémentation des nouveaux modèles proposés (tels que *RGB-a*, *RGB-ab*, *Greenberg-Hastings* avec cellules défaillantes, etc.), les expérimentations décrites dans ce rapport, ainsi que l'ajout de modèles personnalisables et une refonte importante de l'interface utilisateur.

11.1 Nouveaux modèles

Modèles proposés

L'ensemble des modèles décrits dans ce rapport ont été implémentés sur *Fiatlux* pour permettre de les simuler facilement, et effectuer différents tests et mesures sur chacun d'eux. La figure ?? ci-contre présente les 13 nouveaux modèles implémentés dans le nouvel onglet « diagnostic » du sélecteur de modèles.

Modèles programmables

Certains modèles de *Fiatlux* peuvent être paramétrés de façon à permettre de simuler un grand nombre de règles aisément, tels que les modèles *Totalistic* ou *OuterTotalistic*, où l'on peut définir pour chaque quantité de cellules vivantes ou mortes parmi les cellules voisines le nouvel état que la règle doit donner. Cependant, ces options restent limitées lorsque l'on souhaite tester des règles plus « exotiques ». Dans l'optique de simplifier la création de nouveaux modèles, nous avons alors choisi de permettre d'éditer le code de certains modèles, et d'écrire de nouveaux modèles en Java, directement dans *Fiatlux*, avec un éditeur de code intégré (figure 32 ci-dessous). Cet éditeur reste très simple, mais offre tout de même une coloration syntaxique et la numérotation des lignes, permettant ainsi de tester des règles rapidement, particulièrement pour la recherche ou en milieu étudiant.

14. <http://fiatlux.loria.fr>

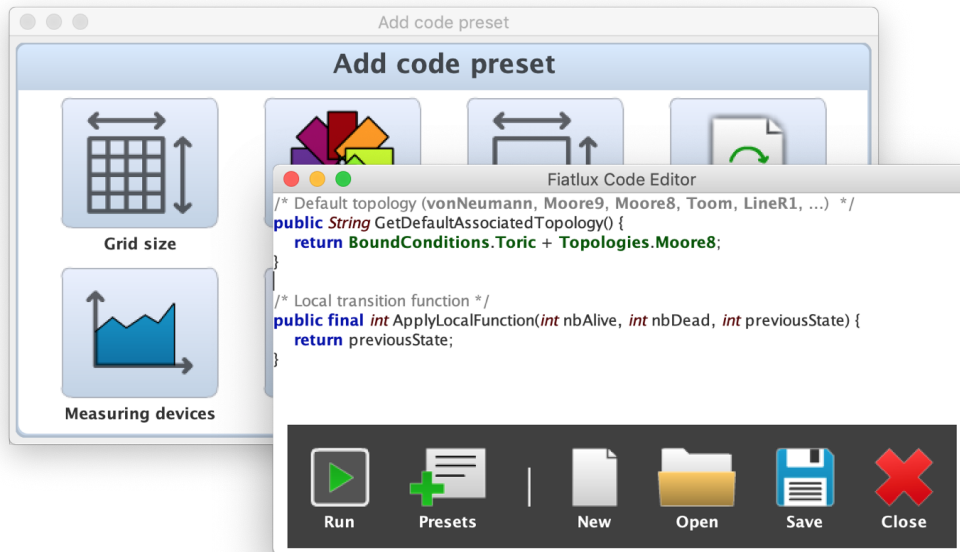
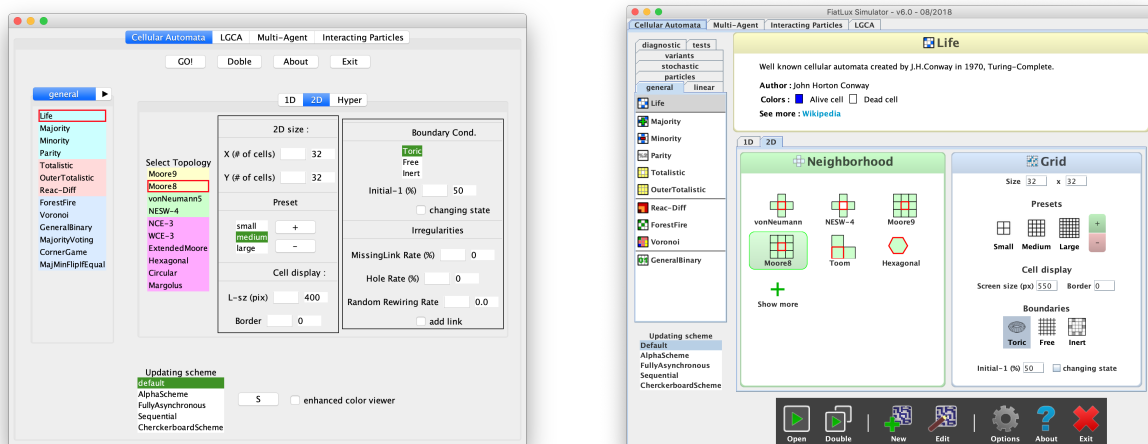


Fig. 32. Éditeur de code de Fiatlux

11.2 Interface utilisateur

L'interface utilisateur de *Fiatlux* se compose de deux fenêtres principales : la première permettant de choisir et configurer un modèle (création de simulation), et la seconde permettant d'afficher et contrôler la simulation lancée. L'interface de création de simulation propose de nombreuses options, et permet de configurer un nombre important d'automates cellulaires en définissant la topologie de la grille, ses dimensions et conditions de bords, le synchronisme de la règle, l'initialisation et éventuellement certains paramètres de la règle de transition. Cependant, la profusion de ces paramètres peuvent rendre l'interface difficile d'accès pour les utilisateurs ne connaissant pas bien le logiciel, et en décourager certains d'explorer les nombreux modèles qui peuvent être simulés avec *Fiatlux*. Pour cette raison, j'ai entrepris une refonte importante de l'interface utilisateur dans l'optique de le rendre le plus intuitif et accessible possible sans atteindre l'efficacité de l'ancienne interface. La figure 33 montre l'évolution de l'interface de création de simulation.

Création de simulation



Ancienne interface

Nouvelle interface

Fig. 33. Évolution de l'interface de sélection de modèle de Fiatlux.

La nouvelle interface contient un certain nombre de changements majeurs, notamment :

- La liste des modèles (à gauche de la fenêtre) propose des icônes pour chaque modèle, un nouveau rangement en catégories, et un affichage indépendant du système d'exploitation.
- Une description du modèle sélectionné est affichée (cadre jaune en haut de la fenêtre), incluant l'auteur, la palette de couleurs (états) et d'éventuels liens vers des pages web sur le modèle concerné.
- Les voisinages sont affichés visuellement et avec des noms plus intuitifs. Les paramètres de la grille tels que les tailles prédéfinies ou les conditions de bords disposent également d'icônes.

Simulateur

L'interface de la fenêtre de simulation a également été totalement revue :

- Les boutons de contrôle principaux de la simulation (initialisation, lecture / pause, ...) sont sous forme de grands icônes.
- L'ensemble des contrôles et outils supplémentaires sont regroupés dans des panneaux dans la partie droite de la fenêtre, devenus rétractables. Cette nouvelle organisation est plus claire (indiquant pour chaque contrôle ce dont il s'agit), et permet de n'afficher que les paramètres dont nous avons besoin.
- Un nouveau mode « réduit » permet de n'afficher que les contrôles principaux et l'écran de la simulation, pour plus facilement présenter des modèles avec un vidéo-projecteur.



Ancienne interface



Nouvelle interface

Fig. 34. Évolution de l'interface du simulateur de Fialux.

Conclusion

Le cas du diagnostic décentralisé est, à l'image d'autres problèmes tels que la classification de la densité, une question vaste pouvant être abordée de différentes manières et avec différentes contraintes. En effet, utiliser une grille de taille finie ou infinie, ou une application de la règle locale de façon synchrone ou non, peut modifier significativement les approches possibles et les résultats que l'on peut espérer obtenir.

Nous avons cherché à trouver un automate *simple* répondant à la problématique du diagnostic décentralisé, ce qui suppose une approche particulière : nous souhaitions trouver un automate produisant un comportement émergent complexe, et ce type de comportement émergent ne peut généralement pas être prédit à l'avance. Nous avons donc dû explorer des espaces de règles et réaliser de nombreuses expériences pour confirmer ou infirmer chacune des intuitions suivies.

Les premières pistes explorées, telles que le modèle à grille d'état bidimensionnelle (voir annexe A.1) ou les vagues cumulatrices (annexe A.2) n'ont pas donné de résultat probants. En effet, ces modèles sont compliqués (impliquent de nombreux états et de nombreuses règles), non robustes aux asynchronismes ou variations de topologies, et ne présentent pas de comportement émergent.

Par la suite, diverses méthodes basées sur l'activité des cellules et la propagation d'un état d'alerte sur la grille ont donné de meilleurs résultats, permettant un diagnostic sur une grille de taille fixe, et tolérants les asynchronismes. Cependant, là encore, le comportement global n'est pas un phénomène émergent, ces modèles ne sont pas robustes au changement d'échelle et différents motifs de cellules défaillantes peuvent fausser le diagnostic.

L'introduction et l'étude de différents modèles, notamment les règles *RGB*, nous a également amené à découvrir de nouveaux comportements et motifs ; les vagues se déplaçant sur le labyrinthe de la règle *RGB-30* n'ont d'ailleurs à notre connaissance jamais été observées ou étudiées sur des automates cellulaires. Cette étude nous a également menés à de nombreuses nouvelles questions et problèmes, tels que la classification de la densité sur une grille présentant des cellules défaillantes ou encore les propriétés des vagues observées sur les modèles *RGB*.

Nous avons enfin cherché une solution plus simple et robuste au changement d'échelle et à l'asynchronisme. L'utilisation de règles de transition stochastiques permet d'introduire du bruit, d'obtenir des comportements moins dépendants de la topologie et du synchronisme, et donc plus robustes. Cependant, il est plus difficile de prévoir le comportement de ce type de règles à l'avance, et nous avons alors réalisé des simulations pour de nombreuses règles possibles pour trouver celles qui permettraient d'obtenir le comportement souhaité. C'est ainsi qu'a été trouvée la règle inspirée du modèle d'*Ising* de physique statistique. L'inspiration du modèle d'*Ising* a permis d'avoir un paramètre de contrôle λ , définissant le seuil critique de cellules défaillantes provoquant la transition de phase, et ainsi de paramétrer le seuil critique du diagnostic aisément.

Si nous avons pu trouver plusieurs automates cellulaires présentant des résultats satisfaisants par rapport à la problématique du diagnostic décentralisé, de très nombreuses questions restent ouvertes. Les automates cellulaires présentent en effet des comportements très variés, souvent imprévisibles, et mieux comprendre le lien entre les interactions locales et le comportement global de ces derniers est un enjeu majeur. Les phénomènes émergents ne sont en effet pas juste des constructions trop complexes pour qu'on en comprenne le fonctionnement, mais dressent les limites de certaines approches scientifiques et philosophiques, telles que le réductionnisme. Ainsi, les étudier et tenter de comprendre les liens entre le comportement global et les interactions locales constitue une voie prometteuse en vue de mieux comprendre les limites de certains raisonnements. Représenter un système par un ensemble d'éléments interagissants entre eux et formant un groupe semble alors insuffisant, à partir du moment où l'on ne parvient plus à expliquer le comportement du tout à partir de l'ensemble de ses parties. L'idée qu'un élément du groupe interagisse avec ce même groupe – dont il fait donc partie – pourrait constituer un des paradoxes qui nous empêchent, encore aujourd'hui, de réellement comprendre la nature des phénomènes émergents et de les prédire.

Bibliographie

- [Ada10] Andrew Adamatzky. *Game of Life Cellular Automata*. Springer Publishing Company, Incorporated, 1st edition, 2010.
- [AIR88] Michel Adam, Philippe Ingels, and Michel Raynal. Algorithmes distribués synchrones et systèmes répartis asynchrones : concepts, mises en oeuvre et expérimentations. Research Report RR-0862, INRIA, 1988.
<https://hal.inria.fr/inria-00075692>.
- [Bel15] Alex Bellos. *Alex et la magie des nombres*. Robert Laffont, 2015.
- [BF11] Hugues Berry and Nazim Fatès. Robustness of the critical behaviour in the stochastic greenberg-hastings cellular automaton model. *Int. Journ. of Unconventional Computing*, 0 :1–21, 2011.
- [Bou13] Olivier Bouré. *Le simple est-il robuste ? une étude de la robustesse des systèmes complexes par les automates cellulaires*. PhD thesis, Université de Lorraine, 2013.
<https://tel.archives-ouvertes.fr/tel-00918545>.
- [BsFMM13] Ana Bušić, Nazim Fatès, Jean Mairesse, and Irène Marcovici. Density classification on infinite lattices and trees. *Electron. J. Probab.*, 18 :no. 51, 22, 2013.
- [CC07] M. Copelli and P. R.A. Campos. Excitable scale free networks. *The European Physical Journal B*, 56(3) :273–278, Apr 2007.
<https://doi.org/10.1140/epjb/e2007-00114-7>.
- [CF08] Vincent Chevrier and Nazim Fatès. Multi-agent Systems as Discrete Dynamical Systems : Influences and Reactions as a Modelling Principle. Research report, INRIA, 2008.
<https://hal.inria.fr/inria-00345954>.
- [CKM00] David W Carman, Peter S Kruus, and Brian J Matt. Constraints and approaches for distributed sensor network security (final). *DARPA Project report, (Cryptographic Technologies Group, Trusted Information System, NAI Labs)*, 1(1), 2000.
- [Coo04] Matthew Cook. Universality in elementary cellular automata. *Complex systems*, 15(1) :1–40, 2004.
- [Das12] Debasis Das. A survey on cellular automata and its applications. In P. Venkata Krishna, M. Rajasekhara Babu, and Ezendu Ariwa, editors, *Global Trends in Computing and Communication Systems*, pages 753–762, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [dSBdSO⁺16] Núbia Rosa da Silva, Jan M. Baetens, Marcos William da Silva Oliveira, Bernard De Baets, and Odemir Martinez Bruno. Classification of cellular automata through texture analysis. *Information Sciences*, 370-371 :33 – 49, 2016.
<https://doi.org/10.1016/j.ins.2016.07.005>.
- [Eva01] Kellie M. Evans. Larger than Life : Digital Creatures in a Family of Two-Dimensional Cellular Automata. In Robert Cori, Jacques Mazoyer, Michel Morvan, and Rémy Mosseri, editors, *Discrete Models : Combinatorics, Computation, and Geometry, DM-CCG 2001*, volume DMTCS Proceedings vol. AA, Discrete Models : Combinatorics, Computation, and Geometry (DM-CCG 2001) of *DMTCS Proceedings*, pages 177–192, Paris, France, 2001. Discrete Mathematics and Theoretical Computer Science.
<https://hal.inria.fr/hal-01182968>.
- [Fat04] Nazim Fatès. *Robustesse de la dynamique des systèmes discrets : le cas de l’asynchronisme dans les automates cellulaires*. PhD thesis, École normale supérieure de Lyon, 2004.
- [Fat13] Nazim Fatès. Stochastic cellular automata solutions to the density classification problem. *Theory of Computing Systems*, 53(2) :223–242, Aug 2013.
<https://doi.org/10.1007/s00224-012-9386-3>.
- [FGG91] Robert Fisch, Janko Gravner, and David Griffeath. *Cyclic Cellular Automata in Two Dimensions*, pages 171–185. Birkhäuser Boston, Boston, MA, 1991.
https://doi.org/10.1007/978-1-4612-0451-0_8.

- [Fis90] Robert Fisch. Cyclic cellular automata and related processes. *Physica D : Nonlinear Phenomena*, 45(1) :19 – 25, 1990.
<http://www.sciencedirect.com/science/article/pii/016727899090170T>.
- [FMT16] Nazim Fatès, Irène Marcovici, and Siamak Taati. Two-dimensional traffic rules and the density classification problem. In Matthew Cook and Turlough Neary, editors, *22th International Workshop on Cellular Automata and Discrete Complex Systems (AUTOMATA)*, volume LNCS-9664 of *Cellular Automata and Discrete Complex Systems*, pages 135–148, Zürich, France, June 2016. Springer.
<https://hal.inria.fr/hal-01290290>.
- [GDW94] S. Grime and H.F. Durrant-Whyte. Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2(5) :849 – 863, 1994.
[https://doi.org/10.1016/0967-0661\(94\)90349-2](https://doi.org/10.1016/0967-0661(94)90349-2).
- [Gut90] Howard A. Gutowitz. A hierarchical classification of cellular automata. *Physica D : Nonlinear Phenomena*, 45(1) :136 – 156, 1990.
[https://doi.org/10.1016/0167-2789\(90\)90179-S](https://doi.org/10.1016/0167-2789(90)90179-S).
- [HSP04] Cindy E. Hmelo-Silver and Merav Green Pfeffer. Comparing expert and novice understanding of a complex system from the perspective of structures, behaviors, and functions. *Cognitive Science*, 28(1) :127 – 138, 2004.
[https://doi.org/10.1016/S0364-0213\(03\)00065-X](https://doi.org/10.1016/S0364-0213(03)00065-X).
- [Ish96] Yoshiteru Ishida. An immune network approach to sensor-based diagnosis by self-organization. *Complex Systems*, 10(1) :73, 1996.
- [JMG78] S.P.Hastings J. M. Greenberg. Spatial patterns for discrete models of diffusion in excitable media. *Society for Industrial and Applied Mathematics*, 34(3), May 1978.
- [JvN66] Arthur W. Burks John von Neumann. Theory of self-reproducing automata. *University of Illinois Press*, 1966.
- [LL13] J.M. Lévy-Leblond. *Le grand écart : la science entre technique et culture*. Modélisation des imaginaires. Innovation et création. Manucius, 2013.
<https://books.google.fr/books?id=tuYmlwEACAAJ>.
- [Mar13] Irène Marcovici. *Automates cellulaires probabilistes et mesures spécifiques sur des espaces symboliques*. Theses, Université Paris-Diderot - Paris VII, November 2013.
<https://tel.archives-ouvertes.fr/tel-00933977>.
- [MBDS11] Ilora Maity, Gunjan Bhattacharya, Sukanta Das, and Biplab Sikdar. A cellular automata based scheme for diagnosis of faulty nodes in wsn. 10 2011.
- [OT99] Tom Ostoma and Mike Trushyk. Cellular automata theory and physics : A new paradigm for the unification of physics. 08 1999.
- [Pes95] Umberto Pesavento. An implementation of von neumann’s self-reproducing machine. *Artificial Life*, 2(4) :337–354, 1995.
- [SFS10] Antoine Spicher, Nazim Fatès, and Olivier Simonin. Translating discrete multi-agents systems into cellular automata : Application to diffusion-limited aggregation. In Joaquim Filipe, Ana Fred, and Bernadette Sharp, editors, *Agents and Artificial Intelligence*, pages 270–282, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- [SGC05] B. K. Sikdar, N. Ganguly, and P. P. Chaudhuri. Fault diagnosis of vlsi circuits with cellular automata based pattern classifier. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(7) :1115–1131, July 2005.
<https://ieeexplore.ieee.org/document/1458937/?part=1>.
- [SGMC01] Biplab K Sikdar, Niloy Ganguly, Purnabha Majumder, and P Pal Chauhuri. Design of multiple attractor gf (2/sup p/) cellular automata for diagnosis of vlsi circuits. In *VLSI Design, 2001. Fourteenth International Conference on*, pages 454–459. IEEE, 2001.

- [SHG09] Nadathur Satish, Mark Harris, and Michael Garland. Designing efficient sorting algorithms for manycore gpus. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–10. IEEE, 2009.
- [Vel09] Yvan Velenik. Le modèle d’Ising. Lecture, February 2009.
- [VNB96] John Von Neumann and Arthur Walter Burks. *Theory of self-reproducing automata*. University of Illinois Press Urbana, 1996.
- [VNP⁺08] Deblauwe Vincent, Barbier Nicolas, Couteron Pierre, Lejeune Olivier, and Bogaert Jan. The global biogeography of semi-arid periodic vegetation patterns. *Global Ecology and Biogeography*, 17(6) :715–723, 2008.
<https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1466-8238.2008.00413.x>.
- [WLSC07] John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary. Wireless sensor network security : A survey. *Security in distributed, grid, mobile, and pervasive computing*, 1 :367, 2007.
- [Wol85] Stephen Wolfram. Twenty problems in the theory of cellular automata. *Physica Scripta*, 1985(T9) :170, 1985.
<http://stacks.iop.org/1402-4896/1985/i=T9/a=029>.
- [Wol86] Stephen Wolfram. Random sequence generation by cellular automata. *Advances in Applied Mathematics*, 7(2) :123 – 169, 1986.
<http://www.sciencedirect.com/science/article/pii/019688588690028X>.
- [Zus69] Konrad Zuse. *Allgemeine Betrachtungen*, pages 48–67. Vieweg+Teubner Verlag, Wiesbaden, 1969.
https://doi.org/10.1007/978-3-663-02723-2_4.

A Annexe : autres modèles et expérimentations

Cette section regroupe des résultats expérimentaux et les différentes pistes explorées mais non retenues en vue d'obtenir un diagnostic décentralisé.

A.1 Automate isotrope à grille d'états bidimensionnelle

Nous nous intéressons ici à un automate cellulaire isotrope sur une grille de cellules régulière. L'objectif de cet automate est de diffuser une information sur l'état de la grille (notamment, son taux de cellules défaillantes) à l'ensemble des cellules.

A.1.1 Espace des états

Une grille d'états à deux dimensions est utilisée pour représenter deux données : la stabilité de l'état de la cellule (sur l'axe des abscisses, ici notée *stabilité*) et la moyenne de cellules en bon état de fonctionnement (sur l'axe des ordonnées, notée *défaillance*).

L'état le plus proche de l'origine correspond à l'état **OK**, d'une cellule en bon état de fonctionnement, et celui le plus éloigné à l'état **KO** : la cellule est morte (pouvant correspondre aussi à l'absence de réponse de cette même cellule à ses voisins immédiats). À l'initialisation, toutes les cellules sont dans l'état **OK** si elles fonctionnent.

Le nombre de lignes et de colonnes de l'espace d'états dépend de deux paramètres s et d , à adapter expérimentalement.

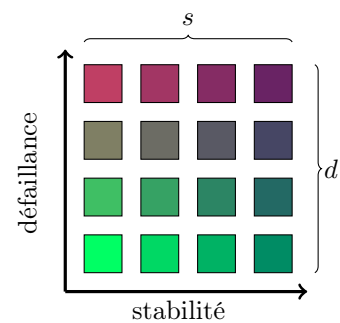


Fig. 35. Espace des états (Q)

Notation : nous noterons q_{ij} l'état à la ligne i et à la colonne j de l'espace d'état, en comptant à partir de 1 en partant de l'origine du repère de la figure 35 ($i \in [1, d]$ et $j \in [1, s]$).

A.1.2 Fonction de transition locale

La fonction de transition locale d'une cellule prend en considération son état courant ainsi que celui de ses voisines immédiates. On calcule séparément la ligne et la colonne du prochain état de la cellule de la manière suivante :

1. **Nouvelle ligne (état de défaillance) :** Calcul de la moyenne des lignes de son état courant et de l'état de ses voisines immédiates.
2. **Nouvelle colonne (état de stabilité) :** Si la nouvelle ligne calculée est identique à celle de l'état courant, déplacement d'une colonne vers la droite (si possible), sinon retour à la toute première colonne.

A.1.3 Diagnostic : estimation du nombre de cellules défaillantes

En sélectionnant aléatoirement une cellule de la grille et en nous intéressant à son état, nous pouvons ainsi obtenir (si celle-ci n'est pas défectueuse) :

- Une estimation du taux de cellules défectueuses de la grille (d'autant plus précise que s est élevé).
- Une indication de la stabilité de son état. Une stabilité élevée indiquant que la cellule n'a pas détecté de changement de moyenne depuis plusieurs itérations (mais ne représente aucunement une garantie que la grille est stable).

Notons que les cellules qui ont une cellule morte parmi leurs voisines garderont une estimation plus élevée que celles qui n'en ont pas.

A.1.4 Évaluation de la méthode de diagnostic

Nous allons définir une méthode d'évaluation de l'estimation du nombre de cellules défaillantes.

- **Donnée considérée :** taux de cellules défaillantes dans la grille.
- **Estimation globale :** rapport exact de cellules défaillantes dans la grille $D = d_{q_{ds}}(c)$.

- **Estimation locale** : probabilité qu'une cellule choisie au hasard donne une *bonne estimation* du taux exact de cellules défaillantes.

Pour calculer cette probabilité, nous devons définir ce qu'est une *bonne estimation*. Pour cela, nous utiliserons deux fonctions f et v , où v retourne un intervalle de \mathbb{R} correspondant à l'estimation du taux de cellules défaillantes et f renvoie 1 si l'estimation globale D est dans l'intervalle de l'état considéré. Nous définissons alors cette probabilité et les fonctions f et v ainsi :

$$\begin{aligned} v : Q &\longrightarrow \mathcal{P}(\mathbb{R}) \\ q_{ij} &\longmapsto \left[\frac{i-1}{d}, \frac{i}{d} \right] \\ & \quad i \in \{1, d\}, j \in \{1, s\} \end{aligned} \quad (1)$$

$$\begin{aligned} f : Q &\longrightarrow \{0, 1\} \\ q_{ij} &\longmapsto \begin{cases} 1 & \text{si } D \in v(q_{ij}) \\ 0 & \text{sinon} \end{cases} \end{aligned} \quad (2)$$

$$P(\text{bonne estimation}) = \frac{\sum_{g \in \mathcal{L}} f(\eta(g))}{|\mathcal{L}|} \quad (3)$$

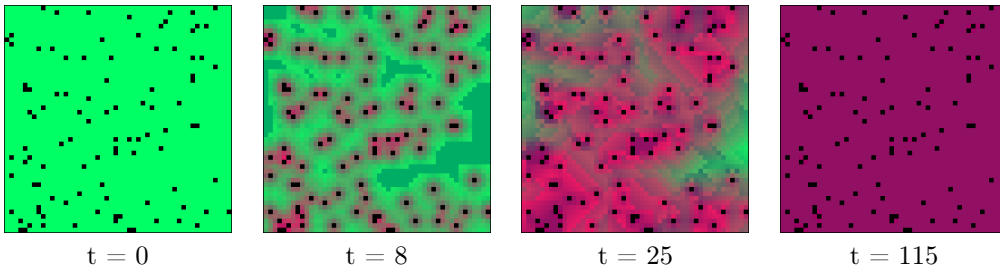


Fig. 36. Évolution d'une configuration de l'automate

A.1.5 Intérêt du modèle

Ce modèle a la particularité d'être isotrope, mais aussi extrêmement robuste aux asynchronismes (une application asynchrone de la règle de transition ralenti la stabilisation de l'état de la grille, mais celle-ci finit par arriver au même état avec ou sans asynchronisme). Cependant, l'estimation donnée par ce modèle n'est pas la bonne, il est donc à adapter en vue d'obtenir le comportement souhaité.

A.2 Réaction diffusion avec vagues cumulatives

Adaptation du modèle de Greenberg-Hastings (voir section 4.2) dans le but d'obtenir des vagues excitatrices autour des cellules défaillantes qui se cumulent pour détecter un nombre donné de cellules défaillantes. L'objectif est de faire en sorte que les cellules défaillantes produisent des vagues excitatrices, qui, lorsqu'elles se rencontrent, provoquent de nouvelles vagues plus intenses, jusqu'à un seuil d'alerte donné.

A.2.1 Espace des états

L'espace d'états est défini par un paramètre n indiquant le nombre d'états d'excitation et le nombre d'états réfractaires, et comporte au total $2n + 3$ états :

- Un état normal par défaut : \square
- L'état défaillant q_d : \boxed{X}
- Un état d'alerte q_a : $\boxed{\text{points}}$
- n état(s) d'excitation q_{a1} à q_{an}
- n état(s) réfractaire(s) q_{r1} à q_{rn}

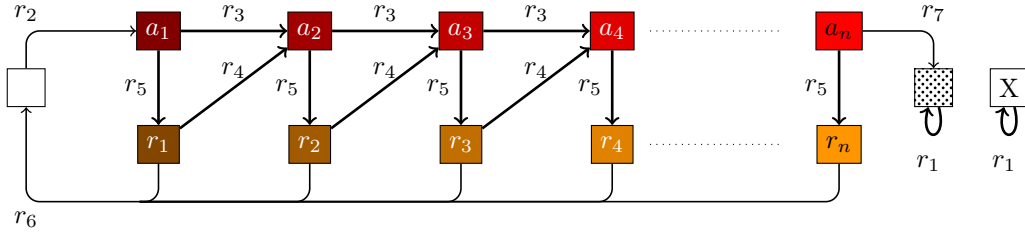


Fig. 37. Transitions entre états (dans un souci de lisibilité, toutes les transitions possibles ne sont pas représentées ici)

A.2.2 Règle de transition

La règle de transition des cellules suit les conditions suivantes :

1. Une cellule défaillante ou en état d'alerte ne plus plus changer d'état (r_1).
2. Une cellule non défaillante ayant au moins une voisine en état d'alerte (q_a) adopte ce même état d'alerte (non représenté sur la figure 37).
3. Une cellule dans l'état normal ayant une cellule défaillante dans son voisinage passe au premier état d'excitation q_{a1} (r_2).
4. Une cellule dans un état d'excitation (q_{a_i}) passe dans l'état d'excitation suivant ($q_{a_{i+1}}$) si elle est entourée d'au moins 4 voisines dans le même état (q_{a_i}), et 2 voisines dans l'état réfractaire correspondant (q_{r_i}) (r_3), sinon, elle passe à l'état réfractaire correspondant q_{r_i} (r_5). Si la cellule doit passer dans l'état d'excitation suivant, mais est déjà dans l'état d'excitation maximal (q_{a_n}), alors elle passe en état d'alerte q_a (r_7).
5. Une cellule réfractaire (q_{r_i}) peut immédiatement passer à l'état d'excitation suivant ($q_{a_{i+1}}$) si elle est entourée d'au moins une voisine dans l'état d'excitation suivant (r_4), sinon, elle redevient normale (r_6).

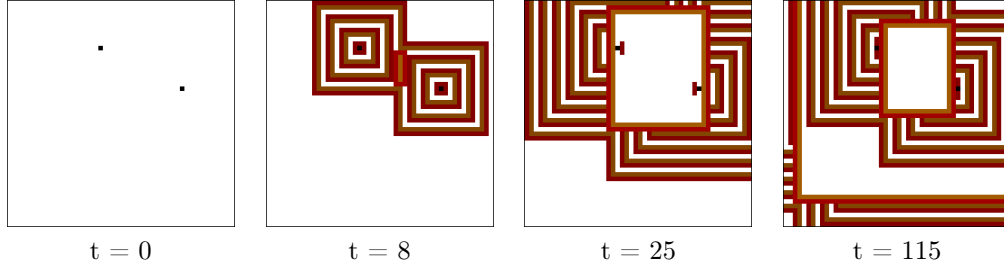


Fig. 38. Évolution d'une configuration de l'automate avec deux cellules défaillantes

A.2.3 Résultats

Test sur 30 configurations de taille 50×50 , détection de 5 cellules défaillantes ou plus ($n = 5$), sur 10000 itérations.

Cellules défaillantes	1	2	3	4	5	6	7	8	9	10
Configurations en alerte	0	0	1	5	14	21	25	26	29	28
Taux d'erreur	0%	0%	3,3%	16,7%	46,7%	30,0%	16,7%	13,3%	3,3%	6,7%

A.2.4 Remarques sur le modèle

Ce modèle permet de détecter un nombre fixe de cellules défaillantes sur la grille, indépendamment de sa taille. Cependant, les vagues excitatrices peuvent facilement se déformer du fait des cellules défaillantes et ainsi provoquer des erreurs de détection (faux positifs et faux négatifs) pour certaines configurations. De plus, ce modèle n'est pas robuste aux asynchronismes ou liens de voisinage défaillants, nous privilégierons donc d'autres pistes.

A.3 Règles « Larger than Life »

Des règles dites « *Larger Than Life* » en référence au Jeu de la vie de Conway ont été étudiées. Celles-ci utilisent un voisinage d'un rayon supérieur, et, comme le jeu de la vie original, une règle de survie ainsi qu'une règle de naissance, dépendantes du nombre de cellules voisines en vie. Kellie M. Evans propose notamment une règle sur un voisinage de Moore de rayon 5 avec les conditions suivantes [Eva01] :

- *Règle de naissance* : une cellule morte devient vivante si elle a entre 34 et 45 voisines en vie.
- *Règle de survie* : une cellule vivante survit si elle a entre 34 et 58 voisines en vie.

Cette règle possède plusieurs points communs avec la règle de Conway : elle permet l'apparition de structures évoluant de manière spécifique, telles que des glisseurs¹⁵, et ces structures émergent de grandes configurations aléatoires avec une densité appropriée (voir exemple ci-contre, figure 39).

Comme pour le jeu de la vie, la plupart des configurations aléatoires vont se stabiliser vers une densité de cellules donnée, mais il existe un intervalle de densités initiales propice à l'émergence de comportements complexes, en dehors duquel on atteindra généralement une configuration stable, ou une configuration dans laquelle toutes les cellules sont mortes. Les variations de cette règle ou du voisinage considéré permettent également d'obtenir d'autres intervalles pour ces comportements.

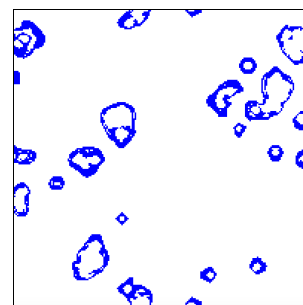


Fig. 39. Evolution d'une configuration aléatoire (densité de 0,5) après 50 itérations

A.3.1 Étude de la densité sur les différents voisinages

Nous allons étudier les évolutions de configurations aléatoires de différentes densité avec les voisinages proposés figure 40.

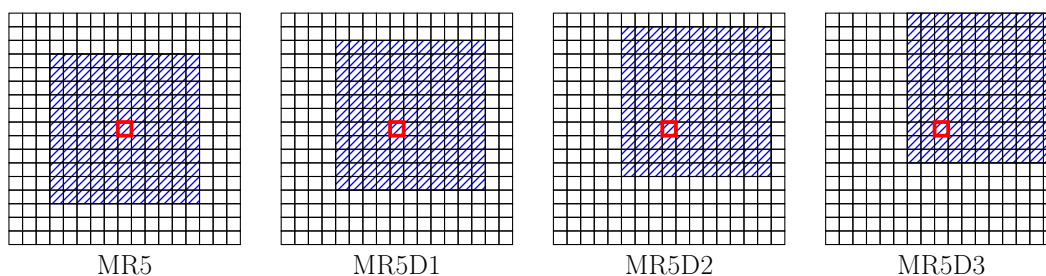


Fig. 40. Voisinages considérés

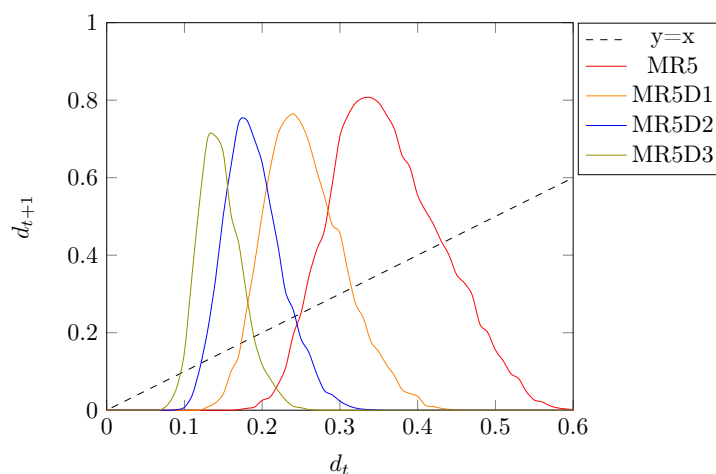


Fig. 41. Évolution de la densité ($d_{t+1} = f(d_t)$) pour différents voisinages

15. [https://fr.wikipedia.org/wiki/Planeur_\(jeu_de_la_vie\)](https://fr.wikipedia.org/wiki/Planeur_(jeu_de_la_vie))

A.3.2 Variations de la règle

Nous allons explorer l'espace des règles fonctionnant de la manière suivante : une cellule en vie reste en vie si le nombre de ses voisins vivants est dans son intervalle de survie (règle de survie) ; une cellule morte redevient vivante si le nombre de ses cellules vivantes est dans son intervalle de naissance. Nous noterons ces règles **B** a - b **S** c - d où $[a, b]$ est l'intervalle de naissance et $[c, d]$ l'intervalle de survie ($a, b, c, d \in \mathbb{N}$).

Pour trouver des règles intéressantes, il nous suffit de tirer au sort 4 nombres positifs (a, b, c, d), inférieurs ou égaux au degré maximal du voisinage (ici 120, pour un voisinage de Moore de rayon 5) et respectant $a \leq b$ et $c \leq d$. Ensuite, nous appliquons les règles trouvées sur des configurations aléatoires, et nous pouvons alors pré-sélectionner celles faisant varier la densité de cellules sur plusieurs itérations, sans aboutir rapidement à une densité de 0 ou de 1.

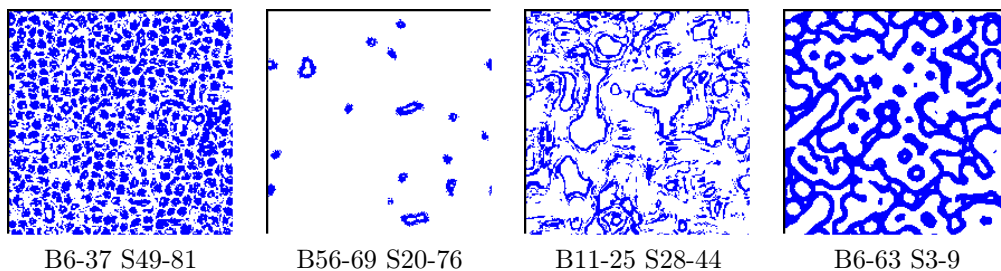


Fig. 42. Évolution de plusieurs règles sur des grilles de 200×200 après 100 itérations.

Voici quelques règles obtenues qui présentent un comportement intéressant :

- **B6-37 S49-81** : règle produisant des « paquets » en mouvement régulier, immobilisés lorsque le nombre de cellules défaillantes devient important.
- **B56-69 S20-76** : effet vagues (fond piscine), mouvement semblant perpétuel.
- **B11-25 S28-44**, **B6-63 S3-9** : totalement chaotique, changement de phase net lorsque des cellules défaillantes apparaissent (cycle clignotant de taille 2).

A.3.3 Conclusion

Ces règles possèdent des propriétés intéressantes, mais sont complexes à calculer du fait du voisinage de grande taille, ce qui peut être trop contraignant dans le cadre de ce sujet. D'autres pistes plus simples, sur un voisinage plus restreint seront donc préférées.

A.4 Expériences sur le modèle de Greenberg-Hastings

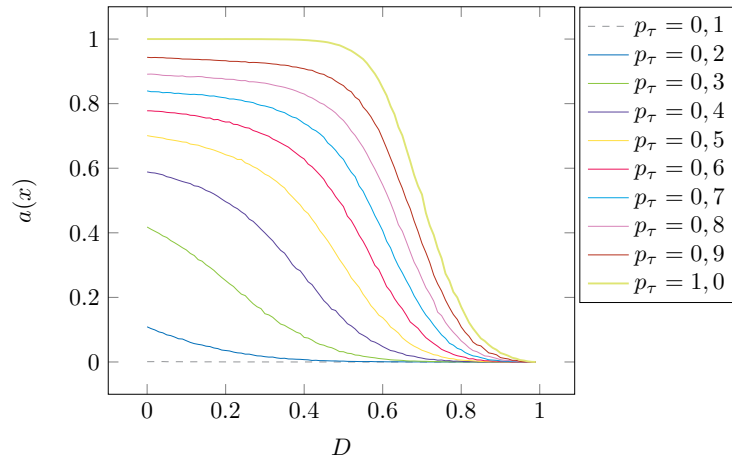


Fig. 43. Évolution de l'activité des cellules non-défaillantes en fonction de la densité de cellules défaillantes (sur une moyenne de 5 expériences, après 10 itérations sur des grilles de 200×200 cellules initialisées aléatoirement).

La figure 44 montre des mesures du nombre de configurations en alerte en fonction de la densité de cellules défaillantes pour différentes tailles de la grille pour observer l'influence de la taille de la grille sur le diagnostic.

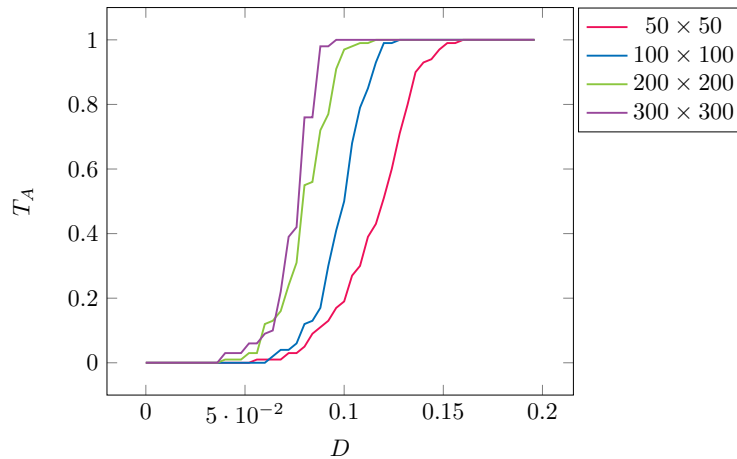


Fig. 44. Influence de la taille de la grille sur le diagnostic

La figure 45 montre le taux de configurations en alerte pour différents α -synchronismes. On constate un décalage régulier pour $\alpha < 1$.

Étude analytique de l'activité

Nous appellerons ici N , E et R les états normal, excité et réfractaire, et n_N , n_E , n_R les quantités de cellules voisines dans chacun de ces états pour une cellule donnée. Nous garderons N , E , R et D pour les densités de cellules dans les différents états, et noterons n le nombre de cellules de la configuration x . Soit $\mu(N, E, R, D)$ une configuration aléatoire où l'état de chaque cellule est tiré indépendamment avec une probabilité N d'être dans l'état N , E dans l'état E , R dans l'état R et D dans l'état D .

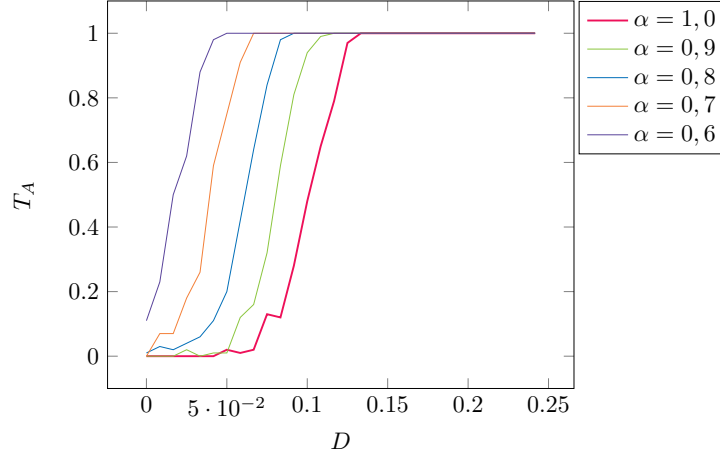


Fig. 45. T_A en fonction de D pour différents α -synchronismes (test effectués sur des grilles de 100×100 cellules après 500 itérations, pour $p_\tau = 0,3$ et $|Q'| = 50$).

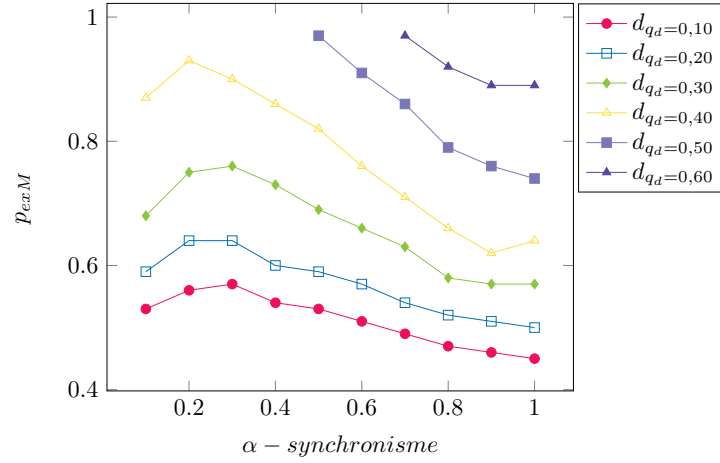


Fig. 46. Probabilité d'excitation maximale produisant $d_{q_{ex}} = 0$ en moins de 1000 itérations, à partir d'une configuration aléatoire de taille 512×512 .

$$\begin{aligned}
 a(x) &= \sum_{c \in \mathcal{L}} \frac{I(c,x)}{|\mathcal{L}|} \\
 \text{avec } I(c,x) &= \begin{cases} 1 & \text{si } F(x)_c \neq x_c \\ 0 & \text{sinon} \end{cases} \\
 a(\mu(N, E, R, D)) &= \mathbb{E} \left[\sum_{c \in \mathcal{L}} \frac{I(c, \mu(N, E, R, D))}{|\mathcal{L}|} \right] \\
 &= P(q = N, n_E > 0) \times p_\tau + P(q = E) + P(q = R) \\
 &= N \times P(n_E > 0) \times p_\tau + E + R \\
 &= N \times (1 - P(n_E = 0)) \times p_\tau + E + R \\
 &= N \times (1 - (N + R + D)^8) \times p_\tau + E + R \\
 &= 1 - D - N \times (1 - E)^8 \times p_\tau \\
 a(\mu(N, E, R, D)) &= 1 - D - (1 - D - 2E) \times (1 - E)^8 \times p_\tau
 \end{aligned}$$

A.5 Règle semi-chaotique

La figure 47 présente l'évolution de l'activité en fonction du taux de cellules défaillantes sur la grilles pour les différentes règles testées.

A.6 Modèle RGB

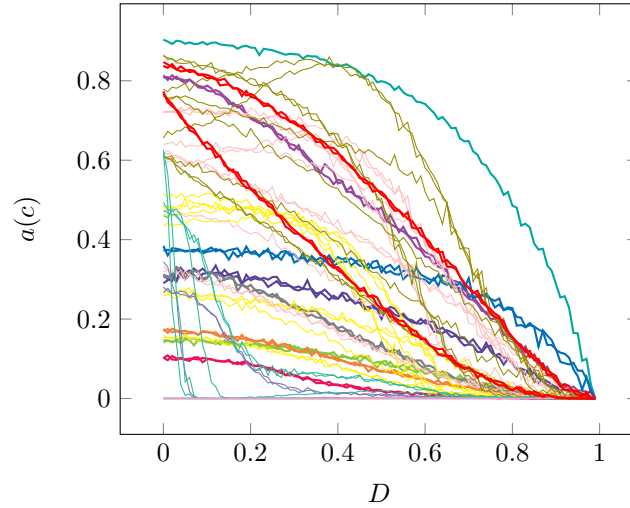


Fig. 47. Évolution de l'activité en fonction du taux de cellules défaillantes pour différentes règles (mesures effectuées sur des configurations de 100×100 cellules après 500 itérations, avec le voisinage de *von Neumann*). Les paramètres des 79 règles représentées ici ne sont pas indiqués puisque nous cherchons uniquement à sélectionner, parmi les règles possibles, une règle ayant une activité importante jusqu'à un taux de défaillance D élevé, mais nulle à partir d'un certain seuil.

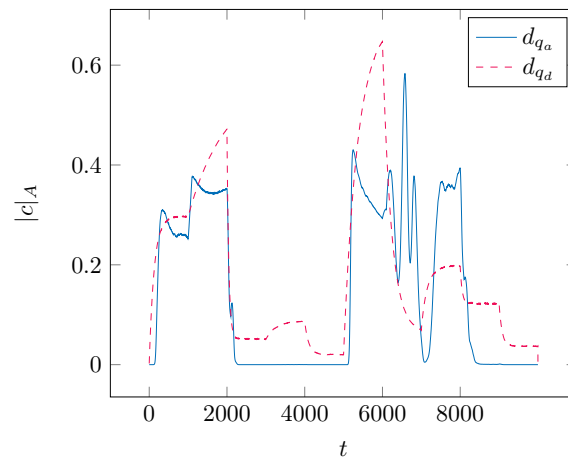


Fig. 48. Évolution de la densité de cellules en état d'alerte (restées inchangées pendant plus de 100 itérations) et de la densité de cellules défaillantes au cours du temps sur une grille de 500×500 cellules.

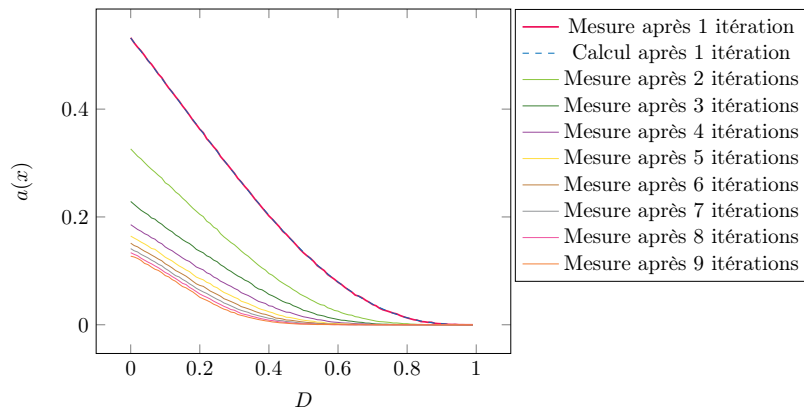


Fig. 49. $a(x) = f(d_{q_{def}})$ pour la règle *RGB-a-2*

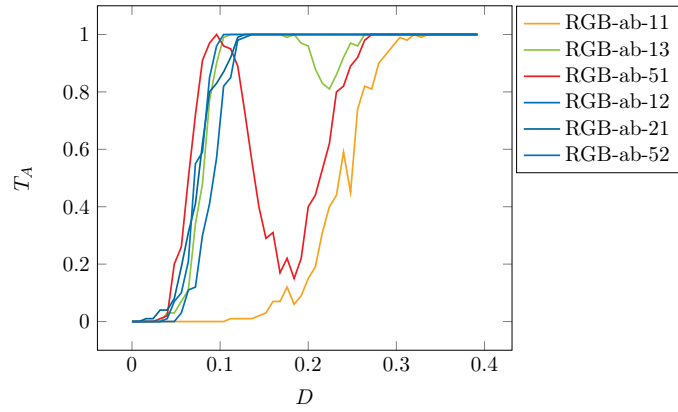


Fig. 50. T_A après 500 itérations sur des grilles de 100×100 cellules initialisées aléatoirement pour différents taux de cellules défailantes (100 configurations testées pour chaque taux de défaillance).

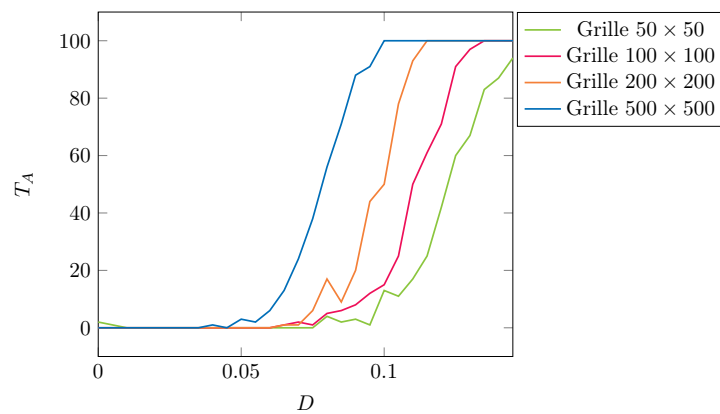


Fig. 51. T_A en fonction de D après 1000 itérations sur des grilles initialisées aléatoirement pour différentes densités de cellules défailantes.

A.7 Comparaison des modèles

Temps de déclenchement du diagnostic

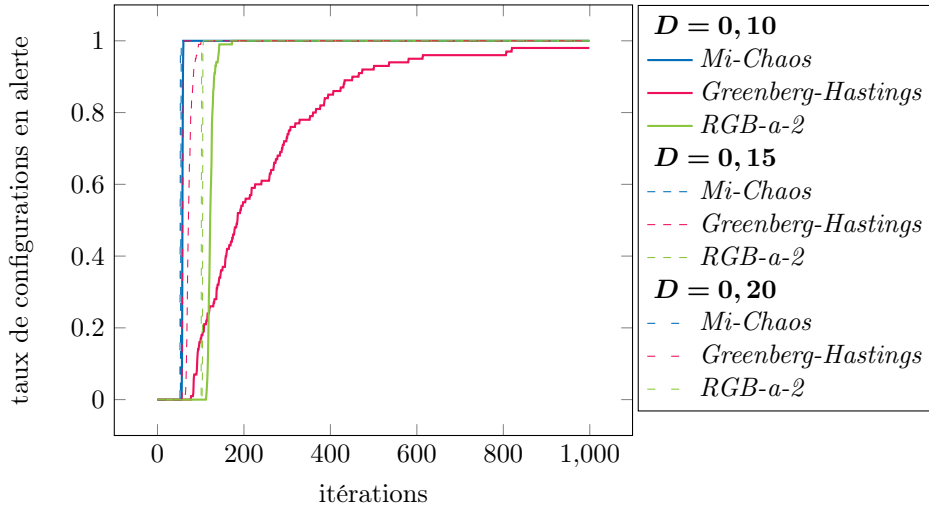


Fig. 52. Nombre d'itérations nécessaires au déclenchement de l'alerte pour différentes densités de cellules défaillantes initiales (tests effectués sur 100 configurations de 200×200 cellules).

B Annexe : autres modèles sur grille infinie

Nous avons vu que la propagation d'un état d'alerte posait problème en tant que moyen d'obtention de consensus sur l'état de la grille. Nous allons maintenant envisager une méthode de diagnostic ne présentant pas ce défaut. Pour cela, nous pouvons par exemple conserver l'utilisation de l'activité et coupler deux automates cellulaires fonctionnant par intermittence de la manière suivante :

- Le premier automate applique une des règles utilisées précédemment (*Greenberg-Hastings*, *RGB* ou *MiChaos*) en utilisant deux états pour mesurer l'activité. Cette fois, une absence prolongée d'activité n'entraîne pas d'état d'alerte.
- Après T_1 itérations, on arrête le premier automate. On constitue alors une nouvelle configuration à partir de celle obtenue avec le premier automate, tel que les cellules les plus actives deviennent des cellules dans un état 1, les autres dans un état 0 (les cellules défaillantes restent défaillantes). Le second automate peut alors réaliser une classification de la densité pour déterminer l'état majoritaire, et ainsi trancher sur l'activité présente sur la grille (supérieure ou inférieure à un seuil donné).
- En fonction du résultat obtenu par le second automate après T_2 itérations, on passe en état d'alerte ou on reprend le fonctionnement du premier automate pendant T_1 itérations.

Cette méthode a l'avantage de ne plus utiliser d'état se propageant sur toute la grille, mais peut poser d'autres problèmes :

1. L'alternance des deux règles nécessite une synchronisation globale des cellules, par exemple grâce à un compteur dans chaque cellule. Cette alternance peut alors être perturbée par d'éventuels asynchronismes, et la méthode de diagnostic peut ainsi être compromise.
2. Les paramètres T_1 et T_2 doivent être suffisamment grands pour permettre d'obtenir une adaptation de l'activité au nombre de cellules défaillantes, mais aussi un consensus de la classification de la densité avec le second automate. Cependant, plus T_1 et T_2 sont grands et plus la réponse du diagnostic sera longue.
3. L'alternance des deux règles nécessite également la présence d'un compteur d'itérations pour chaque cellule, et donc un état supplémentaire, en plus de celui évaluant l'activité et ceux utilisés par les deux règles appliquées, ce qui complexifie le modèle de manière non négligeable.
4. La règle de la majorité avec un voisinage de Toom permet une classification de la densité sur une grille infinie, mais dans le cas d'un espace d'états binaire. Ici, les cellules défaillantes ont la propriété, en plus de ne permettre aucune transmission d'information, de ne pas pouvoir changer d'état, ce qui risque d'affecter la classification de la densité.

B.1 Classification de la densité avec défaillance

La règle de majorité avec le voisinage de *Toom* classe la densité sur une grille de taille infinie. Cependant, nous n'avons pas de garantie sur une grille présentant des cellules défaillantes, celles-ci ne pouvant d'ailleurs pas changer d'état. Nous allons donc nous intéresser plus particulièrement à la règle de majorité dans le cas d'une grille possédant des cellules défaillantes.

Dans un premier temps, nous avons plusieurs possibilités pour adapter cette règle dans le cas de cellules défaillantes. En effet, le voisinage de *Toom* comptant trois cellules, il n'y a, dans le cas d'un espace d'état binaire, pas de cas d'égalité; dans le cas de défaillances, nous avons donc trois états et une égalité possible. Nous avons alors quatre choix en cas d'égalité : favoriser l'une des couleurs (V ou M), ou maintenir l'état inchangé, ou inverser l'état. Dans les quatre cas, il semble que nous ne puissions plus obtenir de classification de la densité lorsque la grille contient des cellules défaillantes.

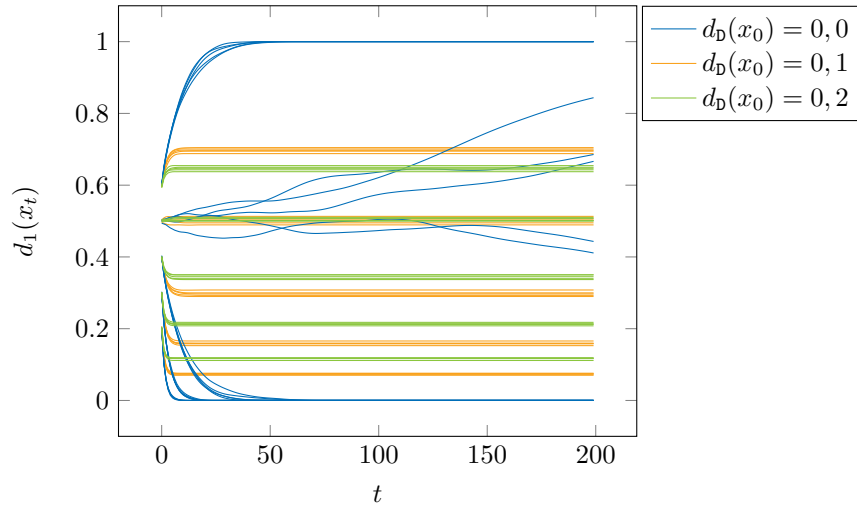


Fig. 53. Évolution de la densité de cellules dans l'état 1 au cours du temps avec différentes densités initiales (0,3 à 0,6), pour différents taux de cellules défaillantes sur la grille. Ces mesures ont été effectuées sur des grilles toriques de taille 200×200 , avec la règle de majorité sur le voisinage de Toom.

La figure 53 montre l'évolution de la densité de cellules vivantes pour différentes densités initiales, et avec différents taux de cellules défaillantes. Ces mesures ont été effectuées sur des grilles toriques. Notons que la règle de majorité avec voisinage de *Toom* ne permet pas de classifier la densité sur une configuration torique dans tous les cas (des lignes verticales, horizontales ou diagonales peuvent empêcher un consensus), la classification parfaite avec cette règle n'est assurée que dans le cas de grilles de taille infinie. Cependant, nous utilisons ici des grilles toriques pour simplifier les expérimentations, la classification pouvant fonctionner sur une grille torique dans la majorité des cas, nous pouvons ainsi comparer les résultats avec et sans présence de cellule défaillantes.

Les résultats présentés ont été effectués avec une règle neutre (laissant la cellule sans changement d'état en cas d'égalité). On constate que la classification fonctionne pour des densités initiales différentes de 0,5 en l'absence de cellules défaillantes (en bleu sur la figure 53). Malheureusement, la classification ne marche plus avec 10 et 20% de cellules défaillantes; la densité évolue « dans la bonne direction », mais le consensus n'est jamais atteint (pas d'état « tout vivant » ou « tout mort »).

B.1.1 Voisinage de Toom sans défaillance

Une autre solution pourrait permettre de se rapprocher de la classification de la densité en évitant les problèmes posés par les cellules défaillantes : conserver le voisinage de Toom, mais « remplacer » les cellules défaillantes du voisinage par d'autres cellules. Concrètement, il s'agit de considérer le voisinage de Toom (hachures bleues sur la figure 54), mais si une ou plusieurs cellules de ce voisinage sont défaillantes, les remplacer par des cellules non-défaillantes du voisinage de Moore (hachurées en rouge sur la figure 54) choisies aléatoirement. Une alternative est de prendre trois cellules non-défaillantes (si possible) au hasard dans le voisinage de Toom.

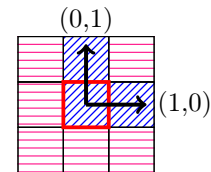


Fig. 54. Toom modifié

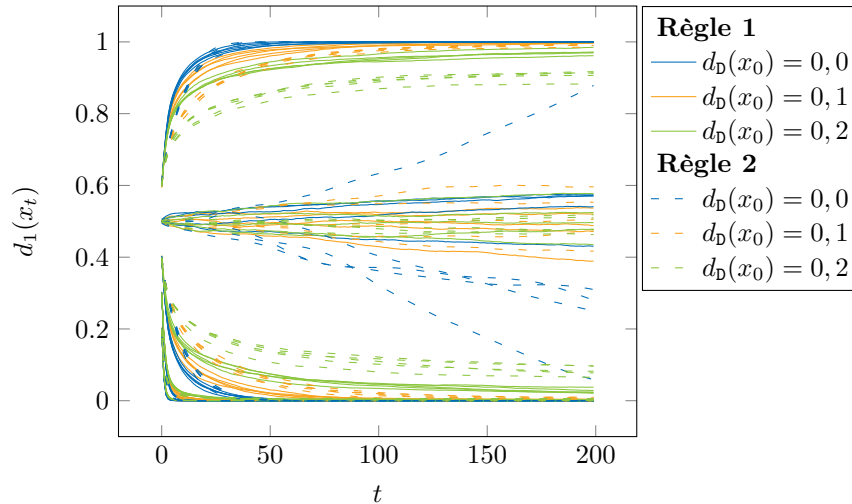


Fig. 55. Évolution de la densité de cellules dans l'état 1 au cours du temps avec différentes densités initiales (0,3 à 0,6), pour différents taux de cellules défaillantes sur la grille, avec le voisinage de Toom complété. Ces mesures ont été effectuées sur des grilles toriques de taille 200×200 . **Règle 1** : voisinage de Toom, remplacé par des cellules du voisinage de Moore choisies au hasard si présence de cellules défaillantes, **règle 2** : choix de trois cellules non-défaillantes au hasard parmi celles du voisinage de Moore.

La figure 55 montre les résultats obtenus avec ces nouveau voisinage. Cette fois, le consensus est bien obtenu pour les trois densités de cellules défaillantes (0, 0, 1 et 0, 2) avec le seconde règle, nous pouvons donc utiliser cette règle pour classifier la densité. La seconde règle montre des résultats faiblement affectés par le taux de cellules défaillantes. Pour $d_1(x_0)$ proche de 0,5, la seconde règle ne parvient pas à obtenir de consensus rapidement, de même que la règle de majorité sur le voisinage de Toom en l'absence de cellules défaillantes. En revanche, la première s'éloigne d'une densité de 0,5 en l'absence de cellules défaillantes (courbes bleues en pointillés), ce qui pourrait être préférable dans certains cas.

B.1.2 Diagnostic à deux états

Une autre variante est également envisageable : utiliser le modèle de Greenberg-Hastings en ne regardant non pas l'activité des cellules, mais la présence de l'état d'excitation, ce qui permet de supprimer la nécessité d'avoir un état évaluant l'activité des cellules. Les fonctionnement est le suivant :

- Lors de l'initialisation, toutes les cellules sont dans l'état normal (**N**).
- Une cellule normale (**N**) ayant une voisine défaillante devient excitée avec une probabilité p_{τ_1} ; les cellules défaillantes « diffusent » alors des signaux d'excitation.
- Les cellules normales ayant des voisins excités ont une probabilité p_{τ_2} de devenir excités à leur tour, et celles excités passent à l'état réfractaire (**R**) en une itération, comme dans le modèle probabiliste de *Greenberg-Hastings*.
- Après T_1 itérations, comme dans la méthode proposée plus haut, on passe à une seconde règle produisant une classification de la densité, les cellules excités (et éventuellement réfractaires) deviennent vivants et les autres morts. Nous souhaitons alors qu'un nombre important de cellules défaillantes produise un nombre également important de cellules excités, provoquant un état « tout vivant » lorsque le seuil de cellules défaillantes ρ est atteint, et « tout mort » dans le cas contraire.

B.1.3 Problème de l'asynchronisme

En cas d'asynchronisme, les compteur d'itérations (second état des cellules) pourront avoir des valeurs différentes selon les cellules, et ainsi le passage de l'application de la première règle (réaction diffusion) à la seconde (classification de la densité) pourra se faire à différents instants selon les cellules, laissant pendant quelques itérations les deux règles s'appliquer en même temps (et donc avec des cellules dans des états qui ne sont pas attendus par les règles appliquées). Sur une période (c'est à dire $T_1 + T_2$ itérations), un éventuel asynchronisme pourra provoquer un décalage léger de quelques itérations pouvant se résorber rapidement, mais celui-ci risque ensuite d'augmenter à chaque période, jusqu'à produire un mélange des

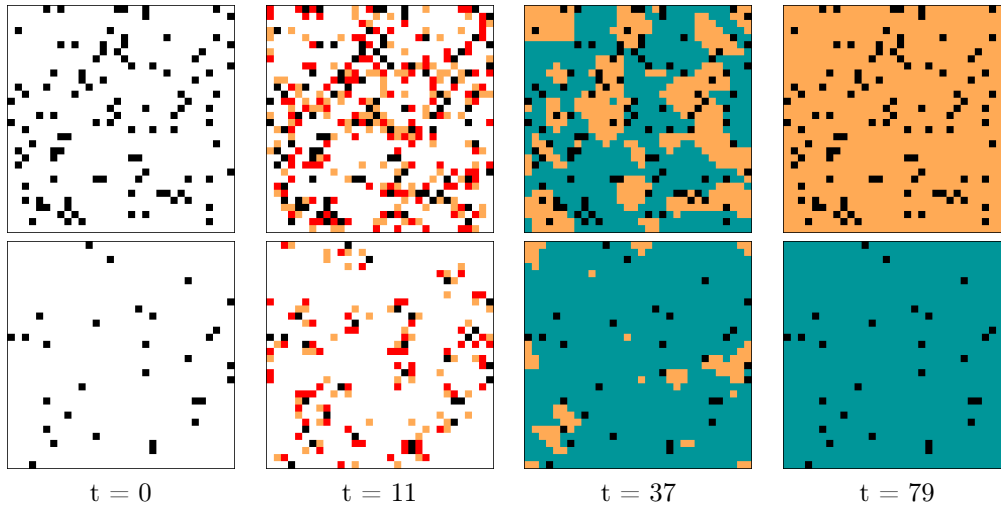


Fig. 56. Évolution d’une configuration de 32×32 cellules avec une densité de cellules défaillantes de 10% (en haut) et 3% (en bas), $T_1 = 30, T_2 = 100, p_{\tau_1} = 0, 3, p_{\tau_2} = 0, 1$. Lors de l’initialisation (première colonne), toutes les cellules non défaillantes sont dans l’état normal (N). Pendant les 30 premières itérations (T_1), la règle de *Greenberg-Hastings* est appliquée, puis pendant les 100 suivantes la règle de classification de la densité. Ici, la configuration atteint un état tout vivant (état V en orange ici), correspondant à une alerte en haut, et tout mort (état M, en turquoise ici) en bas. Ce résultat dépend de D, p_{τ_1} et p_{τ_2} .

deux règles sur toute la grille, ne permettant plus aucun diagnostic. L’idéal serait donc de trouver un moyen de « re-synchroniser » tous les compteurs à intervalle régulier. Remarquons notamment qu’à la fin d’une période, toutes les cellules non-défaillantes repassent actuellement à l’état normal quelque soit leur état précédent, il n’y a donc aucune transmission d’information lors de cette conversion. Nous pourrions par exemple ajouter une étape à ce moment dans l’objectif de synchroniser tous les compteurs de la grille en quelques itérations. Nous allons pour le moment nous contenter de prendre, parmi l’ensemble du voisinage, la valeur la plus fréquemment trouvée. Notons que dans tous les cas, le modèle de *Greenberg-Hastings* n’étant pas robuste à l’asynchronisme, le seuil détecté sera modifié en fonction de α .

B.1.4 Test du diagnostic

Nous allons maintenant pouvoir mesurer l’efficacité du diagnostic pour différents taux de cellules défaillantes comme nous l’avons fait avec les modèles *RGB*, *Greenberg-Hastings* et *MiChaos*. Pour ce modèle, nous allons à nouveau tirer 100 configurations aléatoires pour différents taux de cellules défaillantes, et cette fois attendre $T_1 + T_2$ itérations pour observer les densités des états V et M et ainsi obtenir le diagnostic obtenu (ou non, si pas de consensus).

Nous allons maintenant pouvoir évaluer cette nouvelle méthode de diagnostic sans propagation.

- ✓ **Influence de la taille de la grille** : c’est sur ce critère que les trois modèles basés sur l’activité posaient problème, notamment parce que si un motif finis de cellules défaillantes provoquait l’alerte, alors plus la taille de la grille augmente et plus la probabilité de déclenchement d’une alerte augmente. On constate sur la figure 57 qu’un changement de la taille de la grille ne provoque plus de fort décalage du seuil de défaillance détecté (voir figure 51 pour comparer avec le modèle *RGB* par exemple).
- ✗ **Résistance à l’ α -synchronisme** : ce modèle étant basé sur celui de *Greenberg-Hastings*, l’asynchronisme déclenche également ici un décalage du seuil détecté. Ce décalage n’est pas problématique si l’ α -synchronisme est constant et α proche de un. En revanche, sur la figure 57, on constate que le seuil détecté est décalé, mais le consensus également plus lent (il faudra probablement augmenter T_1 et T_2 pour contrer ce phénomène).
- ✓ **Vitesse du diagnostic** : nous devons, pour ce modèle, définir T_1 et T_2 en fonction de la taille de la grille et des paramètres du modèle. Après cela, nous avons un diagnostic toutes les $T_1 + T_2$ itérations, ce qui constitue une différence non négligeable avec les autres modèles étudiés. Ce point ne posera pas de problème si l’on peut avoir des itérations rapides et synchrones (ou presque), mais constituera un défaut non négligeable dans le cas contraire.

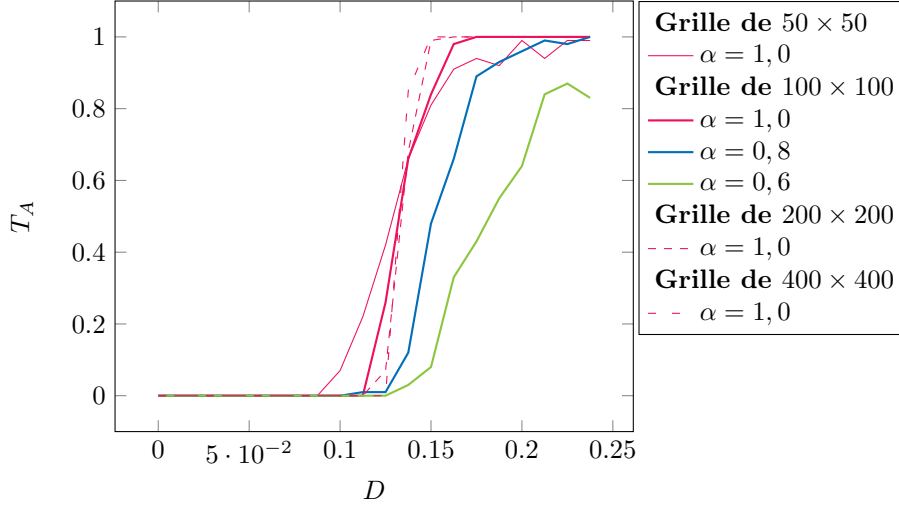


Fig. 57. Taux de configurations en alerte (pour 100 configurations initialisées aléatoirement avec différents taux de cellules défaillantes, avec $p_{\tau_1} = 0,5$, $p_{\tau_2} = 0,2$).

B.2 Classification de la densité biaisée

Le modèle proposé dans les parties précédentes repose sur l’alternance de deux comportements distincts, ce qui pose plusieurs problèmes : la nécessité de synchroniser les cellules à l’aide d’un compteur d’itérations, le diagnostic n’est disponible qu’à certaines itérations (toutes les $T_1 + T_2$ itérations), et la résistance à l’asynchronisme est limitée. Nous allons maintenant essayer de trouver une solution plus simple, en classifiant cette fois la densité de façon « biaisée » par rapport à la présence de cellules défaillantes. Dit autrement, nous souhaiterions qu’une grille initialisée avec certaines densités (par exemple, ou une cellule non défaillante ait une probabilité p_A d’être dans un état d’alerte A, et $1 - p_A$ dans un état normal N), atteigne l’état « tout-alerte » ou « tout-normal » selon le taux de cellules défaillantes de la grille (supérieur ou inférieur à un seuil critique ρ).

B.2.1 Règle de transition locale

Nous allons conserver l’idée générale d’un choix aléatoire en fonction du voisinage de la cellule. Pour simplifier le modèle, nous allons définir une probabilité qu’une cellule passe à l’état N en fonction des quantités de cellules voisines dans chacun des états.

$$f_k : \quad Q \times \mathbb{N}^Q \longrightarrow Q$$

$$(q; n_A, n_N, n_D) \longmapsto \begin{cases} N \text{ avec une probabilité } p_N = \max\left(\frac{n_N - kn_D}{n_A + n_N + n_D}, 0\right), \\ A \text{ avec une probabilité } p_A = (1 - p_N) \frac{n_A + kn_D}{n_A + n_N + n_D}, \\ q \text{ sinon.} \end{cases}$$

B.2.2 Test de la classification biaisée

Sur la figure 58, on constate que la règle donnée permet d’atteindre l’état « tout-normal » pour $d_D(x_0) = 0$, et « tout-alerte » pour $d_D(x_0) \geq 0,3$. Ce résultat est encourageant mais pas optimal : il n’y a pas de consensus absolu pour toutes les valeurs de $]0, 0,3[$. Le paramètre k permet d’adapter la densité de 1 en fonction du taux de défaillance.

B.2.3 Résistance à l’asynchronisme

Sur la figure 59, on constate que l’ α -synchronisme a une influence négligeable sur l’évolution des densités. Cette règle est donc particulièrement robuste à l’asynchronisme pour les critères qui nous intéressent ici.

B.2.4 Règle des carrés

Nous allons maintenant chercher à simplifier la règle de transition locale de façon à pouvoir étudier analytiquement l’évolution des densités de cellules vivantes et mortes en fonction de la densité initiale

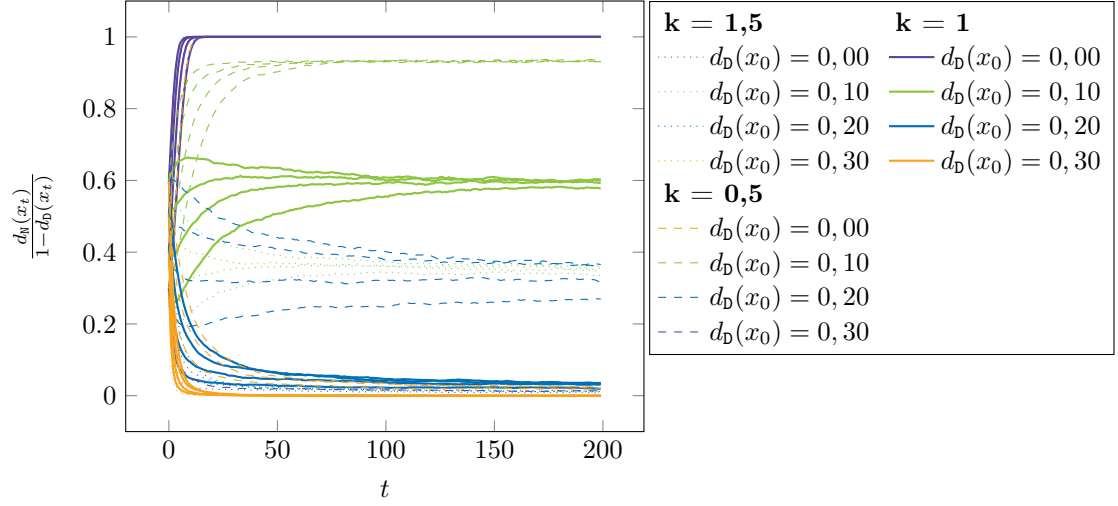


Fig. 58. Évolution de la densité de cellules dans l'état 1 au cours du temps avec différentes densités initiales (0,3 à 0,6), pour différents taux de cellules défaillantes sur la grille, avec le voisinage de Toom complété. Ces mesures ont été effectuées sur des grilles toriques de taille 200×200 .

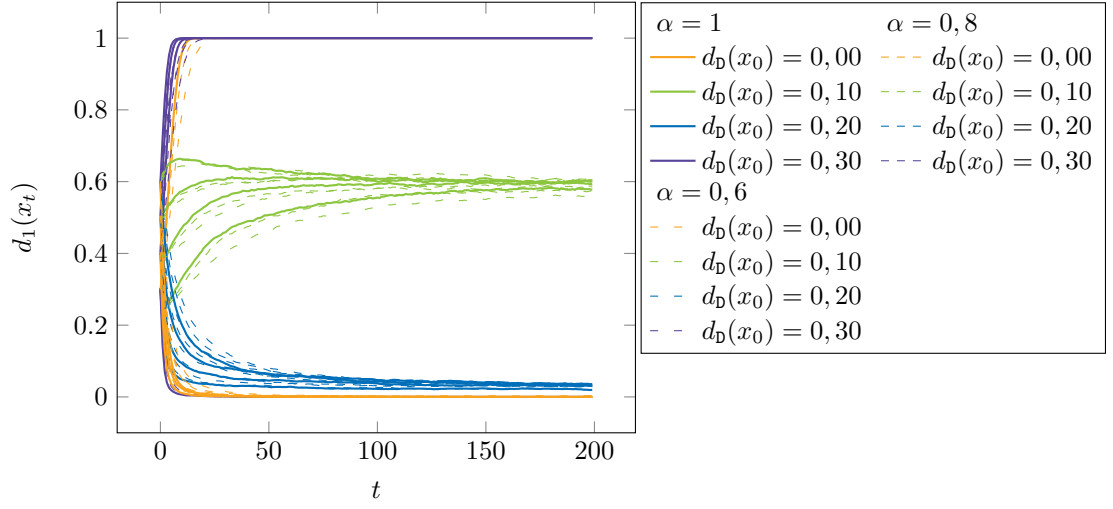


Fig. 59. Évolution de la densité de 1 pour f_1 pour différents taux de synchronisme.

de cellules défaillantes.

$$f_1 : \quad Q \times \mathbb{N}^Q \longrightarrow Q$$

$$(q; n_A, n_N, n_D) \longmapsto \begin{cases} \mathbb{N} & \text{avec probabilité } \frac{n_N^2}{n_A^2 + n_N^2 + n_D^2}, \\ \mathbb{A} & \text{sinon} \end{cases}$$

Soient N_t , A_t et D les densités de cellules dans l'état normal, l'état d'alerte et défaillantes au temps t (D n'est pas indicée puisque la densité de cellules défaillantes est considérée comme constante ici). Sous hypothèse du champ moyen, on a :

$$\begin{cases} N_{t+1} = \frac{N_t^2}{N_t^2 + A_t^2 + D^2} (1 - D) \\ A_{t+1} = \frac{A_t^2 + D^2}{N_t^2 + A_t^2 + D^2} (1 - D) \end{cases}$$

A l'équilibre (lorsque $N_{t+1} = N_t$, $A_{t+1} = A_t$), nous noterons ces densités N , A et D , nous avons alors :

$$2N^2 - 3(1 - D)N + 1 - 2D + 2D^2 = 0$$

$$\Delta = b^2 - 4ac = (-3(1 - D))^2 - 8(1 - 2D + 2D^2)$$

Il existe donc une solution si $\Delta \geq 0$, soit si $D < D_c = \frac{1}{(1 + 2\sqrt{2})} \approx 0,29$

Ce seuil ne correspond pas à celui observé expérimentalement, mais nous montre qu'il existe également un seuil critique de cellules défaillantes dans l'hypothèse du champ moyen.

B.2.5 Test du diagnostic

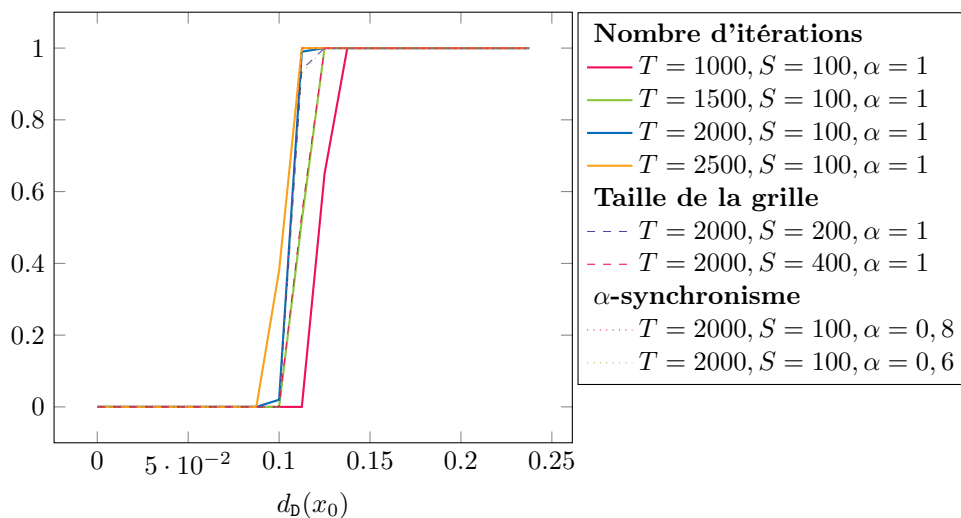


Fig. 60. Taux de configurations en alerte ($d_A(x_t) = 1 - d_D(x_t)$) après T itérations sur des grilles de $S \times S$ cellules pour la règle carré.

- ✓ **Taille de la grille** : pour ce modèle, l'augmentation de la taille de la grille ne modifie pas le diagnostic, il est donc robuste au passage à l'échelle, contrairement aux modèles basés sur la propagation d'un état d'alerte étudiés dans les parties précédentes.
- ✓ **Nombre d'itérations** : les configurations proche du seuil critique peuvent nécessiter un grand nombre d'itérations avant d'atteindre l'état « tout-alerte ». Expérimentalement, le nombre d'itérations avant la mesure n'affecte plus fortement le diagnostic après environ 2000 itérations. Notons que pour un taux de défaillance éloigné du seuil critique, l'évolution sera généralement beaucoup plus rapide.
- ✓ **α -synchronisme** : ce modèle est également très robuste à l'asynchronisme, à l'inverse du modèle de *Greenberg-Hastings* qui produit une transition de phase nette.