UNIVERSITÀ
DEGLI STUDI
FIRENZE

PhD in Co-tutoring
University of Perpignan via Domitia (France)
University of the Study of Florence (Italy)

PhD in
Information Engineering
curriculum Control, Optimization and Complex Systems
(Department of Information Engineering – DINFO- University of Florence)
Applied Mathematics
(Faculté des Sciences, University of Perpignan)

Recent advances in regional controllability of cellular automata

Academic Discipline (SSD) ING-INF/05

Doctoral Candidate
Dr. Sara DRIDI

Supervisors
Samira El Yacoubi (Univ. Perpignan)

Franco Bagnoli (Univ. Florence)

Coordinator
Prof. *Fabio Schoen*

CYCLE XXXII Years 2016/2019

Recent advances in regional controllability of cellular automata

Mathematics has beauty and romance. It's not a boring place to be, the mathematical world. It's an extraordinary place; it's worth spending time there.

Marcus Du Sautoy.

This thesis is dedicated
to the soul of my mother

Acknowledgments

This thesis is a result of research carried out at the Laboratory IMAGES Espace-Dev of University of Perpignan and the University of Florence, Italy.

First and foremost, I would like to express my deep and special thanks to my supervisor **Prof. Samira El YACOUBI**, as well as my co supervisor **Prof. Franco BAGNOLI** for their patience, motivation and their continuous support during my PhD research studies. I would like to thank them also for their friendship and great sense of humour.

I would like to thank also the members of my thesis committee starting by the referees: I would like to thank **Prof. El Hassan ZERRIK** and **Prof. Andrew ADAMATZKY** not only for the time they devoted for reading my manuscript but for agreeing to evaluate my thesis results and for their insightful comments that improved this manuscript. I'm indebted to the rest of committee thesis **Prof. Laurent LEFEVRE**, **Dr. David DEFOUR**, **Dr. Allyx FONTAINE**, **Dr. Duccio FANELLI** and **Prof. Abdelhaq EL JAI** for their extreme patience, guidance, discussion, ideas, comments and encouragements and review of this document.

A special thanks goes to **Dr. Allyx FONTAINE**. I'm grateful for her collaboration in the work described in chapter 5. I am also very grateful to **Prof. Mircea SOFONEA** as well as to **Dr. Annick TRUFFERT** who accepted to take part in this committee thesis. A special thanks to all the members of the lab IMAGES Espace DEV at the University of Perpignan and the members of the Centre for the Study of Complex Dynamics (CSDC) at the University of Florence, Italy, especially, **Dr Marie EL JAI**, **Dr Mohamed Oussama BEN SALEM** , **Dr. Yves MAURISSEN** and **Dr Giovanna PACCINI** and also all the PhD students of our lab: Théo, Nadia, Jamal, Phuong, Pius.

I would like to thank the University of Florence for the funding during my stay at the Centre for the Study of Complex Dynamics (CSDC).

I gratefully acknowledge the Algerian government for giving me a golden opportunity to prepare my thesis in France, I would like to thank them for the funding received during my thesis. I would like to thank also all my professors of the University of Setif, Algeria, where I did my master. I wish sincerely thank my friends from the University of Setif: Narimane Kessir, Soumia Manaa, Fairouz Zoubai, Nesrine

Semcheddine.

I would certainly be careless to not mention my colleagues Asma Mansouri, Wissam Latreche and Ahlem Benraouda for all the support and the fun that we have had during our stay in Perpignan.

I owe special thanks to my closer friends, Sabrina Bouchama and Soraya Chaghoub. I would like to express my thanks for their unconditional support and love.

This thesis couldn't have been accomplished without the love and support of my beautiful family, especially my sister Ilhem. I would like to say a heartfelt thank you for all your support and encouragement. Also, I would like to thank my brother Hakim, my sister Souheila and my father for their encouragement and for believing in me. Last but not the least, I would like to dedicate my thesis again to the soul of my mother.

Sara Dridi

Résumé

Le sujet abordé dans cette thèse concerne la contrôlabilité d'une classe de systèmes spatio-temporels, entièrement discrets de type automates cellulaires (AC). Le but de cette étude est de mettre en lumière de nouvelles pistes pour prouver la contrôlabilité des systèmes complexes. Plus spécifiquement, cette thèse se focalise sur la contrôlabilité régionale qui consiste à se restreindre à une région du domaine où le système devra atteindre un objectif donné à travers des actions ciblées. Le cas d'AC Booléens a été particulièrement examiné tout au long de cette thèse.
La première partie est consacrée à l'étude du problème de la contrôlabilité régionale des AC déterministes lorsque les actions sont exercées sur la frontière de la région contrôlée. Une première démarche que nous avons utilisée s'appuie sur les chaines de Markov et la contrôlabilité est caractérisée en établissant une matrice similaire à leur matrice de transition en utilisant les définitions d'une chaîne ergodique et régulière. Cette étude a été étendue au cas des AC probabilistes qui sont largement utilisés pour modéliser de nombreux phénomènes réels.
Le même problème a été appréhendé en utilisant des outils de la théorie des graphes. Nous proposons des conditions nécessaires et suffisantes pour la contrôlabilité régionale des AC déterministes en utilisant les notions de circuit hamiltonien et de composante fortement connexe. Le contrôle qui assure la contrôlabilité régionale est défini à travers un algorithme préimages.
La deuxième partie est dédiée au problème de la contrôlabilité régionale frontière des AC Booléens qui consiste à agir sur la frontière du domaine pour atteindre un objectif sur une région cible. Nous considérons d'abord des AC linéaires pour lesquels nous donnons un résultat de caractérisation grâce à la condition de Kalman. Nous proposons un algorithme pour déterminer le contrôle qui permet de forcer l'apparition d'une configuration désirée dans la région d'étude. Le cas des AC non linéaires a été également considéré en utilisant un algorithme de recherche des préimages.

Mots clés : Contrôlabilité régionale, Automates Cellulaires Booléens, Chaînes de Markov, Théorie des graphes, Préimages.

Abstract

The issue addressed in this thesis concerns the controllability of a class of discrete spatio-temporal systems named cellular automata (CA). The purpose of this study is to highlight new ways to prove the controllability of complex systems. More specifically, this thesis focuses on regional controllability which consists in restricting the study to a subregion of the domain where the system will have to achieve a given objective through targeted actions. The case of Boolean CA have been particularly examined throughout this thesis. The first part is devoted to the study of the problem of the regional controllability of deterministic CAs when the actions are exerted on the boundaries of the controlled region. A first approach that we used relies on Markov chains and controllability is characterized by establishing a matrix similar to their transition matrix using the definitions of a regular and ergodic chain. This study has been extended to the case of probabilistic CAs that are widely used to model many real phenomena.
The same problem has been apprehended using tools of graph theory. We propose necessary and sufficient conditions for the regional controllability of deterministic CAs using the notions of Hamiltonian circuit and strongly connected component. The control that ensures regional controllability is defined through a preimage algorithm.
The second part is devoted to the problem of the boundary regional controllability of Boolean CAs, which consists of acting on the boundary of the domain in order to reach a desired goal in a target region. We first consider linear CAs for which we give a characterization result using the Kalman condition. We propose an algorithm to determine the control that allows to force the appearance of a desired configuration in the study area. The case of nonlinear CAs was also considered using a preimage search algorithm.

Keywords: Regional controllability, Boolean Cellular Automata, Markov chains, Graph Theory, Preimages.

Riassunto

L'argomento trattato in questa tesi riguarda la controllabilità di una classe di sistemi spaziali-temporali, completamente discreti detti automi cellulari (AC). L'obiettivo di questo studio è quello di studiare nuove strade per dimostrare la controllabilità di sistemi complessi. Più specificamente, questa tesi si concentra sulla controllabilità regionale, che consiste nel determinare una regione del dominio in cui il sistema dovrà raggiungere un determinato obiettivo attraverso azioni mirate. In tutta questa tesi è stato esaminato il caso di AC booleani. La prima parte è dedicata allo studio del problema della controllabilità regionale degli AC deterministici quando si eseguono azioni al bordo della regione controllata. Un primo approccio che abbiamo usato si basa sulle catene di Markov e la controllabilità è caratterizzata dalla creazione di una matrice simile alla matrice di transizione di sistemi stocastici, utilizzando le definizioni di una catena ergodica e regolare. Questo studio è stato esteso al caso di AC probabilistici che sono ampiamente utilizzati per modellare molti fenomeni reali. Lo stesso problema è stato affrontato utilizzando strumenti di teoria dei grafi. Proponiamo le condizioni necessarie e sufficienti per la controllabilità regionale degli AC deterministici utilizzando i concetti di circuito hamiltoniano e delle componenti fortemente connesse. Il controllo che garantisce la controllabilità regionale viene definito tramite un algoritmo di ricerca delle pre-immagini.
La seconda parte è dedicata al problema della controllabilità regionale degli AC booleani, che consiste nell'agire al bordo del dominio per raggiungere un determinato obiettivo nella regione di destinazione. In primo luogo, consideriamo gli AC lineari per i quali diamo un risultato di caratterizzazione grazie alla condizione di Kalman. Proponiamo un algoritmo per determinare il controllo che può forzare l'apparizione di una configurazione desiderata nella regione di studio. Anche il caso di AC non lineari è stato considerato utilizzando un algoritmo di ricerca delle pre-immagini.

Parole Chiave: Controllabilità, Automi cellulari, Catene Markov, Teoria dei grafi, Preimmagini

Contents

List of Figures

NOTATIONS

Symbol	**Description**
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of real positive numbers
$\mathbb{Z}$	Set of integers
$L^2([0,T],\mathcal{U})$	Space of L^2-integrable functions from $[0,T]$ to $\mathcal{U}$
$dom(A)$	Domain of the operator A
$Im(H)$	Image of H
$Ker(H)$	Kernel of H
A^*	Adjoint operator of A
$Z^{'}$	Dual of the space Z
$rank(\mathcal{C})$	Rank of the matrix $\mathcal{C}$
p_ω	Restriction to the region ω
ω	Region (sub domain) of Ω
Γ	Boundary of Ω
$\mathcal{L}(Z,\mathbb{R}^{n_c})$	Space of linear maps from Z to $\mathbb{R}^{n_c}$.
$\mathcal{L}(Z)$	$\mathcal{L}(Z,Z)$
$L^2(\Omega)$	Space of functions square integrable on Ω
S	Set of states

Symbol	**Description**
$\mathcal{N}$	The neighbourhood
$\chi_{\mathcal{L}^p}$	Characteristic function of $\mathcal{L}^p$
$p_{i \rightsquigarrow j}$	The probability to reach the state j from the state i
DPS	Distributed Parameter Systems
PDE	Partial Differential Equation
CA	Cellular Automata
A, B, C	Dynamics, control and observation operators
Z	State space (Hilbert space)
U	Control space (Hilbert space)
Y	Observation (output) space (Hilbert space)
$(\phi_A(t))_{t \geq 0}$	Semi-group generated by A, also denoted $(\phi(t))$
AR	Set of arcs
V	Set of vertices

CHAPTER 1

EXTENDED ABSTRACT IN FRENCH LANGUAGE

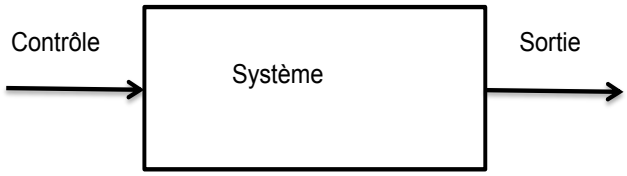
Résumé étendu en langue française

Titre de la thèse :

Nouvelles avancées en contrôlabilité régionale des automates cellulaires

Introduction

La théorie du contrôle est un domaine situé au carrefour de l'automatique, des mathématiques, de l'informatique et de l'ingénierie. Elle consiste à étudier le comportement des systèmes dynamiques sur lesquels des actions sont exercées de manière à les amener vers des objectifs ciblés. Un système contrôlé peut être représenté par le schéma suivant:



Les systèmes considérés peuvent être linéaires ou non linéaires, déterministes ou stochastiques, continus ou discrets et les approches utilisées pour les problèmes de contrôle doivent s'adapter à ces différents types de systèmes et en fonction des applications. Il y a deux représentations possibles des systèmes contrôlés; une dite externe basée sur la notion de fonction de transfert et pour laquelle le système est de type boîte noire. Et une représentation dite interne car basée sur une équation d'état qui implique des variables internes au système. On se base habituellement dans ce cas, sur l'utilisation de modèles mathématiques du système contrôlé, afin de déterminer les actions/entrées à effectuer pour arriver aux mesures/sorties souhaitées. Ces systèmes peuvent être décrits par des équations différentielles ordinaires (EDO), des équations aux dérivées partielles (EDP) ou des équations aux différences. L'évolution du système à partir d'un instant t_0 donné dépend de son état initial et des sollicitations extérieures (commandes, perturbations).

La théorie du contrôle concerne la possibilité de déterminer un contrôle (entrée du système) pour atteindre un état final désiré. On parle alors de contrôlabilité qui peut

être considérée en temps fini ou infini (asymptotique). Une autre propriété intéressante consiste à étudier la possibilité d'estimer la solution du système dynamique uniquement en fonction des observations (sorties). Il s'agit de l'observabilité qui est une notion duale de la contrôlabilité. Il existe d'autres notions fondamentales constituant l'analyse des systèmes tels que la stabilité, stabilisabilité, détectabilité, robustesse, etc. qui permettent une meilleure compréhension du système et de son fonctionnement.

Nous nous intéresserons à une classe de systèmes dits à paramètres distribués car ils sont décrits par des équations aux dérivées partielles et étudiés en termes d'entrées sorties où les variables d'états, de contrôle (entrée) et d'observation (sortie) dépendent à la fois du temps et de l'espace. La difficulté dans l'analyse des systèmes à paramètres distribués vient, entre autres, du fait que les concepts, assez simples dans le cas des systèmes localisés, peuvent être formulés et étudiés à divers degrés à cause de la variable d'espace qui rend infinie la dimension de l'espace d'état.

La contrôlabilité des systèmes distribués constitue un des plus importants concepts d'analyse à avoir joué un rôle important dans l'histoire de la théorie du contrôle. Le problème de contrôlabilité des systèmes distribués a été largement étudié [39, 76, 77]. Nous nous restreindons dans cette thèse à examiner la contrôlabilité à partir de considérations régionales motivées par de nombreuses applications où il est plus pratique de contrôler le système sur une partie privilégiée de son domaine. Le concept de contrôlabilité régionale a été alors introduit pour les systèmes à paramètres distribués par El jai et ses collaborateurs [102]. Cela rentre dans un grand champs de recherche sur l'analyse régionale qui a fait l'objet de nombreux travaux scientifiques dont on peut citer [40, 43, 99, 103, 104] et toutes les références qu'ils contiennent. La plupart des modèles mathématiques utilisés jusqu'alors pour décrire les systèmes distribués est basée sur des équations aux dérivées partielles. Mais à cause de la complexité croissante des systèmes étudiés, il s'est avéré nécessaire de chercher d'autres approches alternatives. Dans ce contexte, les automates cellulaires commencent à devenir de plus en plus populaires et se présentent comme une bonne alternative aux EDP pour des problèmes d'analyse et de contrôle des

systèmes spatio-temporels avec des entrées/sortes spatialisées. Cette approche innovante en théorie des systèmes a été initiée par El Yacoubi et EL Jai au début des année 2000 et a fait l'objet depuis d'une assez riche activité [45, 47, 49, 51, 52].

Les automates cellulaires (AC) constituent une représentation de systèmes dynamiques discrets. Ils consistent en des réseaux réguliers formés d'unités élémentaires toutes identiques appelées cellules. Chacune possède un état pris dans un ensemble fini et communique avec les cellules voisines afin de construire un nouvel état par application synchrone et uniforme d'une règle de mise à jour. Le comportement général de l'AC est spécifié en fonction d'une règle globale dont les propriétés algébriques et topologiques en déterminent le type.

Au delà de leur définition comme étant un modèle classique de calcul parallèle, les AC ont été largement utilisés pour représenter et simuler divers phénomènes en physique, chimie, écologie ou biologie. Malgré leur simplicité formelle, ils constituent un modèle en soi, capable de produire des comportements d'une grande richesse et qui échappent souvent à certaines prédictions. Ils présentent ainsi un cadre idéal pour étudier les systèmes complexes en lien avec des problématiques transversales.

L'objectif de cette thèse se situe dans le contexte général de la théorie des systèmes à paramètres distribués et se focalise plus particulièrement sur l'étude de la contrôlabilité régionale des AC. S'appuyant sur des travaux antérieurs dont on peut citer [98], qui ont permis de définir un AC comme un système ouvert à son environnement alors qu'il était vu auparavant comme un système autonome, les notions de contrôle, observation, capteur et actionneurs ont été formulées dans le paradigme AC.

En lien avec la théorie du contrôle, l'idée consiste dans ce travail, à explorer différents formalismes et approches dont certains sont propres aux EDP qui permettent de prouver la contrôlabilité régionale des AC.

L'essentiel du travail de thèse est présenté sous forme de 5 chapitres au-delà de l'introduction générale et la conclusion. L'objet de ce premier chapitre étant de présenter une synthèse du travail, rédigée en français, comme exigé par le règlement de l'école doctorale lorsque le manuscript est écrit en une langue étrangère, en

l'occurrence, l'anglais.

Le chapitre 3 introduit les différentes notions abordées dans cette thèse. Il décrit notamment les problèmes de contrôle pour les systèmes à paramètres distribués et présente les réalisations en ce qui concerne la contrôlabilité et l'analyse régionale pour des systèmes décrits par des EDP. Il introduit ensuite les automates cellulaires comme alternative aux EDP pour certains problèmes où ces derniers présentent des limites.

Les résultats importants de la thèse et qui ont été valorisés par des publications, seront présentés dans les chapitres 4, 5, 6 et 7 dont la synthèse est donnée dans la section suivante.

Aperçu de la thèse

Chapitre 4 : Approche par Chaînes de Markov pour la Contrôlabilité régionale des Automates Cellulaires

Dans ce chapitre, nous étudions le problème de la contrôlabilité régionale des automates cellulaires deterministes dans les cas unidimensionnels et bidimensionnels et en se focalisant sur les AC dit Booléens caractérisés par un ensemble d'état $S = \{0, 1\}$. Les actions sont supposées s'exercer sur la frontière de la région contrôlée. Cette étude met en lumière une nouvelle approche qui est celle des chaînes de Markov.

Les chaînes de Markov ont été introduites pour la première fois en 1906 par Andrey Markov. Elles constituent des modèles mathématiques utiles qui représentent une classe de processus stochastiques de grand intérêt et qui sont utilisés dans diverses applications. Les chaînes de Markov s'appuyent sur des notions de probabilité pour décrire comment un système passe d'un état à un autre. Un processus de chaîne de Markov satisfait la propriété de Markov qui signifie que le passé et le future sont indépendants lorsque le présent est connu. Dans ce chapitre, nous mettrons l'accent sur la relation entre les chaînes de Markov et les automates cellulaires pour

le problème de contrôlabilité régionale.

Évolution d'un Automate Cellulaire comme limite déterministe d'une chaîne de Markov

Une chaîne de Markov est un processus dynamique décrivant un système dont les états changent au cours du temps. Les changements d'états peuvent être décrits par une matrice de transition.

Définition 1 *Soit M un entier strictement positif. Une matrice P de taille $M \times M$ est appelée matrice de transition lorsque ses éléments $p_{i,j}$ satisfont :*

$$0 \leq p_{i,j} \leq 1 \quad \forall i, j \tag{1.1}$$

$$\sum_{j=1}^{M} p_{i,j} = 1 \quad \forall i \tag{1.2}$$

Définition 2 *La configuration de l'automate cellulaire à l'instant t est définie par l'ensemble $\{s_t(c), c \in \mathcal{L}_c\}$ Elle est exprimée par l'application :*

$$\begin{aligned} s_t : \mathcal{L}_c &\rightarrow S \\ c &\rightarrow s_0(c) \end{aligned}$$

Si s et s' désignent respectivement l'état ou la configuration d'un AC au temps t et $t+1$, l'évolution de l'AC est donnée en fonction de la dynamique globale comme suit :

$$\boldsymbol{s}' = F(\boldsymbol{s})$$

Où F est définie par la fonction:

$$\begin{aligned} F : S^{\mathcal{L}_c} &\rightarrow S^{\mathcal{L}_c} \\ s_t &\rightarrow s_{t+1} \end{aligned}$$

L'évolution de l'AC peut être vue comme un chemin reliant des configurations, c'est à dire qu'on peut introduire une matrice $A_d(\boldsymbol{s}'|\boldsymbol{s})$ de telle sorte que:

$$A_d(\boldsymbol{s}'|\boldsymbol{s}) = p_{\boldsymbol{s},F(\boldsymbol{s})}$$

tels que $p_{\boldsymbol{s},F(\boldsymbol{s})}$ soit la probabilité d'atteindre $F(\boldsymbol{s})$ partant de la configuration s.

Cette matrice peut être vue comme une matrice d'adjacence qui relie des configurations. Elle peut être représentée par un graphe orienté ou comme limite déterministe d'une Chaîne de Markov.
On donne la définition d'un AC linéaire.

Définition 3 *Une dynamique globale F est linéaire si pour chaque paire de configurations $s_1, s_2 \in S^{\mathcal{L}_c}$*

$$\forall \mu \in S, F(\mu s_1 + s_2) = \mu F(s_1) + F(s_2)$$

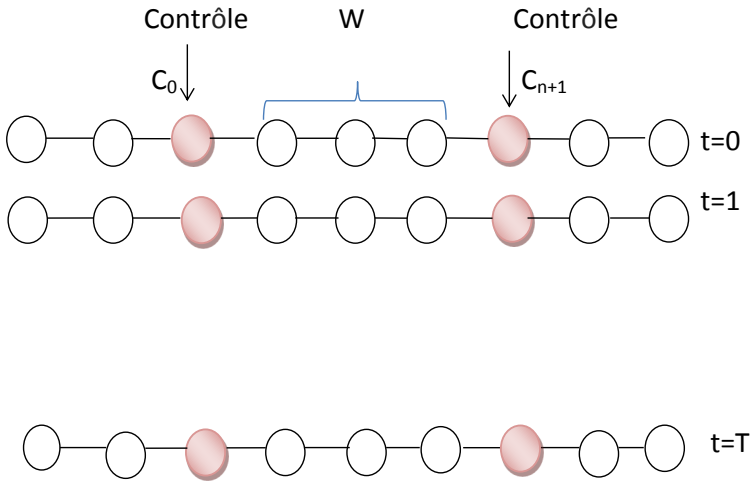
Définition 4 *Si dans un AC, la dépendance du voisinage dépend uniquement de le XOR et le XNOR, alors l'AC est appelée un AC additive. Plus précisément, un AC linéaire n'utilise que des règles XOR.*

Contrôle régional des Automates Cellulaires linéaires unidimensionnelles

Nous abordons un problème particulier de contrôlabilité régionale qui consiste à choisir des actions en agissant sur la frontière de la région cible. L'objectif est de mener le système dynamique à partir d'un état initial à un état désiré choisi dans un temps fini, comme le montre la figure suivante.

Ce problème peut être défini de la façon suivante : considérons un AC unidimensionnel Booléen et une région cible $\omega = \{c_1, \ldots, c_n\}$. On se demande s'il est possible de piloter la configuration initiale $s_0(\omega) = \{s_0(c_1), \ldots, s_0(c_n)\}$ pour l'amener à une configuration désirée donnée au temps T en appliquant des actions sur les frontières $\{c_0, c_{n+1}\}$ de la région ω dans l'intervalle de temps $[0, T-1]$ tels que:

$$s_T(c_i) = s_d(c_i) \quad \forall i = 1, \ldots, n$$



Nous avons utilisé la notion de dérivée Booléenne et ce qui en découle, pour résoudre ce problème.

Définition 5 *Etant donné un AC défini par le quadruplet* $(\mathcal{L}, \mathcal{S}, \mathcal{N}, f)$ *où* f *désigne la fonction de transition locale définie sur* $\mathcal{S}^r$ *à valeurs dans* $\mathcal{S}$ *où* r *représente la taille du voisinage* $\mathcal{N}$*, la dérivée Booléenne partielle de* f *est défini par :*

$$\frac{\partial f_i}{\partial s_j} \overset{not}{=} \frac{\partial f}{\partial s_j} = f(s_i, \ldots, s_j \oplus 1, \ldots, s_{i+r}) \oplus f(s_i, \ldots, s_j, \ldots, s_{i+r})$$

Où $\oplus$ *est l'opération Booléenne XOR et* s_i *désigne l'état de la cellule* i *du réseau unidimentionnel* $\mathcal{L}$*.*

Définition 6 *En termes de dynamique globale de l'AC, la dérivée Booléenne* F' *de* F *est la matrice jacobienne défini par :*

$$J = \frac{\partial f_i}{\partial s_j}|_{i,j=1,\ldots,N}$$

Où N *est la taille de l'AC vue comme le nombre de cellules dans le réseau* $\mathcal{L}$*. L'évolution de l'AC linéaire autonome sera exprimée sous la forme :*

$$s^{t+1} = Js^t,$$

Où s_t est la configuration de l'AC au temps t et s_{t+1} sa configuration au temps $t+1$.

On peut définir alors un automate cellulaire contrôlé par l'équation d'état suivante:

$$s^{t+1} = Js^t \oplus BU^t,$$

où $s_\omega^{t+1} = \{s^{t+1}(c_1), s^{t+1}(c_2), \ldots, s^{t+1}(c_n)\}$, $s_\omega^t = \{s^t(c_1), s^t(c_2), \ldots, s^t(c_n)\}$

B est une matrice de taille $(n \times 2)$ et U^t est le vecteur contrôle de taille (2×1) :

$$B = \begin{pmatrix} \vdots & \vdots \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ \vdots & \vdots \end{pmatrix} \qquad U^t = \begin{pmatrix} u^t(c_0) \\ u^t(c_{n+1}) \end{pmatrix} \tag{1.3}$$

où $u^t(c_0)$ est le contrôle appliqué sur la cellule c_0 et $u^t(c_{n+1})$ est le contrôle appliqué sur la cellule c_{n+1}.

Définition 7 *Un automate cellulaire est dit régionalement contrôlable si pour chaque configuration désirée $s_d \in S^\omega$ il existe un contrôle $U = (U^0, \cdots, U^{T-1})$ où $U^i = (u^i(c_0), u^i(c_{n+1}))$ $\quad i = 0, \ldots, T-1$ tel que*

$$s_T = s_d \quad sur \quad \omega$$

où s_T est la configuration finale au temps T et s_d est la configuration désirée.

Dans cette partie, on définit une matrice similaire à la matrice de transition d'une chaîne de Markov, cette matrice sera construite dans le cas de l'AC, en se basant sur les notions des dérivées Booléennes.

L'évolution de l'AC peut être considérée comme une chaîne de Markov $P_{(s'|s)}$, où $s' = F(s)$ et P est la matrice de transition. Chaque élément de cette matrice est la probabilité de passer d'une configuration s à une autre configuration s'.
Un AC contrôlé peut être représenté par une matrice de transition P où chaque élément $p_{(s,s')}$ représente la probabilité de passer d'une configuration s à une configuration s' en appliquant la paire de contrôles (a, b) sur la frontière de la région cible.

Pour une région ω de taille $|\omega|$, nous avons besoin de définir une matrice carrée $\mathcal{C}$ d'ordre $2^{|\omega|} \times 2^{|\omega|}$, où $2^{|\omega|}$ représente toutes les configurations possibles qui peuvent être représentées dans la région ω.

Exemple

Le plus simple AC Booléen que l'on puisse concevoir c'est l'AC élémentaire (règle de Wolfram) qui consiste en une grille unidimensionnelle de cellules et est caractérisé par un ensemble d'états $\{0, 1\}$ avec un voisinage constitué par la cellule elle même et des deux cellules qui lui sont adjacentes. Comme on a $8 = 2^3$ configurations possibles du voisinage $\{c_{-1}, c_0, c_{+1}\}$ et que chaque configuration peut avoir une image dans $\{0, 1\}$, on aura au total $2^8 = 256$ règles possibles. Ces règles ont été largement étudiées et classifiées par Stephen Wolfram et portent désormais son nom [93]. Chaque règle représentée par un code binaire à 8 digits sera notée par un entier compris entre 0 et 255 qui représentera la conversion décimale du code binaire. Ainsi, la règle 150 décrite ci-dessous s'explique du fait que $150_{10} = 10010110_2$ (sa représentation en binaire).

Motif initial au temps t	111	110	101	100	011	010	001	000
Etat de la cellule centrale à t+1	1	0	0	1	0	1	1	0

Considérons pour la règle 150 de l'AC, une région à contrôlée de taille $|\omega| = 2$. Si la configuration $(00)_2 = 0$ ($(00)_2$ est la représentation binaire de 0), nous avons quatre possibilités d'appliquer le contrôle $(a, b) = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. Nous utilisons la relation $s^{t+1} = Js^t$ pour calculer la prochaine configuration s'. Ainsi on obtient:

$F_{150}(0000) = (00)_2 = 0,$
$F_{150}(0001) = (01)_2 = 2,$
$F_{150}(1000) = (10)_2 = 1,$
$F_{150}(1001) = (11)_2 = 3.$

A partir de la configuration $(00)_2 = 0$ dans la région contrôlée en appliquant le contrôle sur les frontières, on peut atteindre les configurations $0, 1, 2, 3$. On obtient alors $\mathcal{C}_{0,1} = 1, \mathcal{C}_{0,0} = 1, \mathcal{C}_{0,2} = 1, \mathcal{C}_{0,2} = 1, \mathcal{C}_{0,3} = 1$ et on remplit la matrice $\mathcal{C}$ de la même façon pour les trois configurations qui restent $\{(01)_2 = 1, (10)_2 = 2, (11)_2 = 3\}$. La matrice obtenue est :

$$\mathcal{C}_{150} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

La matrice de transition construite $\mathcal{C}$ peut être réduite à une matrice de transition de Markov après normalisation.

Une configuration j est atteignable à partir d'une configuration i et on note $i \rightsquigarrow j$, s'il existe une probabilité strictement positive; ou en termes de la théorie des graphes, s'il existe un arc entre i et j. Parfois, il est impossible d'atteindre une configuration j en un seul pas, mais en un nombre fini m de pas. Nous utilisons donc la définition d'une chaîne de Markov régulière. On dit qu'une chaîne de Markov est régulière s'il existe une puissance $m > 0$ telle que toutes les composantes de P^m sont strictement positives. D'où le résultat suivant :

Théorème 1 *Un Automate cellulaire linéaire est régionalement contrôlable si il existe une puissance $\mathcal{C}^T$ de $\mathcal{C}$, dont tout les coefficients sont strictement positifs.*

Remarque 1 *On peut obtenir le temps requis T pour atteindre la contrôlabilité régionale des AC linéaires en calculant le plus long chemin entre les sommets.*

Pour vérifier la contrôlabilité en temps minimal, nous pouvons utiliser cette définition.

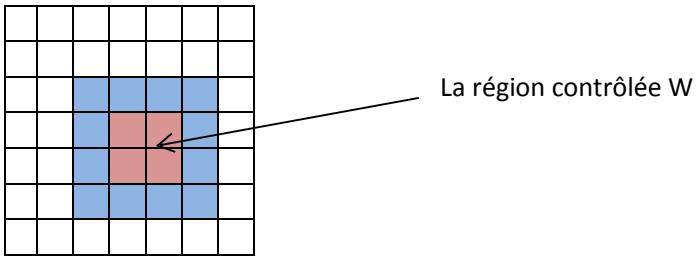
Définition 8 *La chaîne de Markov est appelée chaîne ergodique lorsqu'il est possible d'y aller d'un état à un autre (pas nécessairement en un seul pas).*

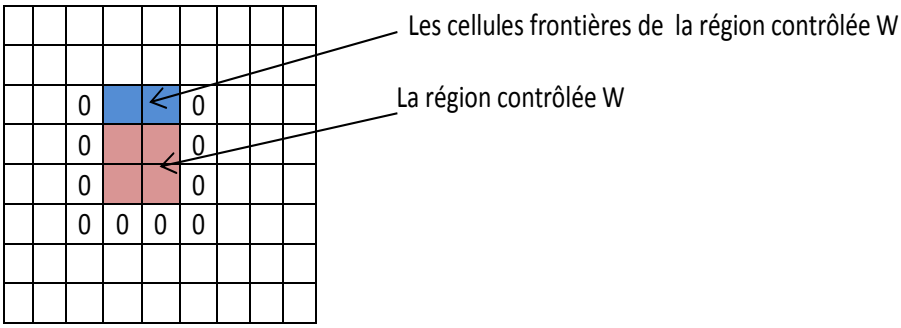
Nous donnons le résultat suivant:

Théorème 2 *Un AC linéaire est régionalement contrôlable s'il est érgodique.*

Cas des Automates Cellulaires bidimensionnels

On considère l'AC bidimensionnel de taille $\mu \times \eta$, L'idée est d'imposer le contrôle sur la frontière de la région contrôlée ω afin de forcer l'apparition d'une configuration donnée dans la région cible ω, voir la figure suivante :





Remarque 2 *Nous pouvons aussi définir des contrôles asymétriques, c'est-à dire que nous gardons une partie des frontières fixes (par exemple à 0) et agissons sur un sous-ensemble de la frontière de la région contrôlée (cellules roses) afin d'obtenir l'état souhaité.*

Lemme 1 *La matrice équivalente de la matrice jacobienne J, c'est une matrice tri-diagonale représentée comme suit :*

$$\mathcal{T}_R = \begin{pmatrix} D & U & ... & ... & ... & 0 & 0 & 0 \\ L & D & U & ... & ... & 0 & 0 & 0 \\ 0 & L & D & U & ... & 0 & 0 & 0 \\ ... & ... & L & D & U & .. & ... & \\ ... & ... & ... & L & D & U & ... & ... \\ ... & ... & ... & ... & L & D & U & ... \\ 0 & 0 & 0 & ... & ... & L & D & U \\ 0 & 0 & 0 & 0 & 0 & 0 & L & D \end{pmatrix}_{\mu\eta\times\mu\eta} \tag{1.4}$$

Où D, L et U sont l'une des matrices suivantes d'ordre $\mu\times\mu$: $[0], [I], [\mathcal{M}_1], [\mathcal{M}_2], [I+\mathcal{M}_1], [I+\mathcal{M}_2], [\mathcal{S}]$ et $[I+\mathcal{S}]$, et $\mathcal{S}$ est la somme de $\mathcal{M}_1$ et $\mathcal{M}_2$, $\mathcal{M}_1$ et $\mathcal{M}_2$ sont définies comme suit :

$$\mathcal{M}_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{1.5}$$

et

$$\mathcal{M}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{1.6}$$

Cette matrice $\mathcal{T}_R$ est de taille $(\mu\eta \times \mu\eta)$. L'état du système s' est obtenu en multipliant $\mathcal{T}_R \times s$, tel que s peut être représentée par une matrice d'ordre de colonne 1D $(\mu\eta \times 1)$ obtenu à partir de la 2D binaire $(\mu \times \eta)$ [49].

l'évolution d'un AC bidimensionnel est donnée par:

$$s^{t+1} = \mathcal{T}_R s^t$$

Le théorème obtenu s'applique également aux automates cellulaires non linéaires et bidmiensionnels linéaires. D'où le résultat :

Théorème 3 *Un Automate Cellulaire non linéaire est régionalement contrôlable s'il existe une puissance $\mathcal{C}^T$ tels que tous ses coefficients sont strictement positifs.*

Définition 9 *Un état $i \in S$ est dit absorbant si $P_{ii} = 1$ et donc nécessairement $P_{ij} = 0$ pour chaque $i \neq j$.*

En utilisant cette définition on a le résultat suivant :

Théorème 4 *Un automate cellulaire est non régionalement contrôlable en exerçant une action sur sa frontière à un temps donné t s'il admet un état absorbant.*

Aussi on a le résultat suivant :

Théorème 5 *Un AC déterministe est non régionalement contrôlable aux frontières, si l'AC admet un état de configuration i où $\mathcal{C}_{ij} = 0 \quad \forall j = 1, \ldots, 2^{|\omega|}$*

Chapitre 5: Approche par la théorie des graphes pour la contrôlabilité régionale des Automates Cellulaires

Ce chapitre est consacré à la contrôlabilité régionale des automates cellulaires unidimensionnels et bidimensionnels en proposant une nouvelle approche basée sur la théorie des graphes. Suite au chapitre précédent, nous prouvons la contrôlabilité régionale à travers quelques notions de la théorie des graphes tels que le circuit hamiltonien et la composante fortement connexe. Nous finissons par proposer un algorithme pour créer les préimages afin de déterminer le contrôle nécessaire pour atteindre une configuration désirée.

L'évolution dans le cas de l'AC déterministe Booléen peut être représentée par un graphe orienté dont les sommets correspondent aux configurations obtenues à

partir de la représentation binaire convertie en décimale. Il existe un arc entre deux sommets v_1 et v_2 si la configuration correspondante à v_2 peut être atteignable à partir de la configuration obtenue à partir de v_1 en appliquant la fonction de transition locale sur la représentation binaire de v_1. L'évolution d'un automate cellulaire contrôlé dans une région ω peut être exprimée par un graphe orienté où les sommets représentent les configurations dans la région contrôlée et ils sont reliés entre eux par un arc s'il y a un contrôle frontière (l, r) telle que $F(l.\lambda(v_1).r) = \lambda(v_2)$ où $\lambda(v_1)$ est la représentation binaire du sommet v_1 et F est la fonction de dynamique globale.

Dans ce chapitre, nous étudions le problème de la contrôlabilité régionale des AC déterministes en explorant une approche originale de la théorie des graphes.

Tout d'abord, nous commençons par rappeller quelques notions de la théorie des graphes qui seront utilisées plus loin dans ce chapitre :

Définition 10 *Un graphe G consiste en une paire $(V(G), AR(G))$ où $V(G)$ est un ensemble fini non vide dont les éléments sont appelés sommets et $AR(G)$ est un ensemble fini dont les éléments sont appelés arêtes.*

Définition 11 *La matrice d'adjacence est une matrice associée à un graphe G dont ses éléments représentent le nombre d'arêtes orientées allant du sommet v_i vers le sommet v_j.*

Définition 12 *Un graphe est dit orienté si ses arêtes sont parcourues dans un seul sens.*

On utilise dans ce chapitre la matrice $\mathcal{C}$ construite dans le chapitre précédent. D'où les résultats suivants :

Condition nécessaire et suffisante : circuit hamiltonien

Un circuit hamiltonien dans un graphe G est défini comme un cycle qui traverse chaque sommet exactement une fois et retourne au sommet de départ. En d'autres termes, le circuit hamiltonien est un chemin qui part d'un sommet original et passe à tous les sommets une seule fois sauf au sommet original. Le circuit Hamiltonien est nommé après l'invention d'un jeu de puzzle en 1857 par le mathématicien William

Rowan Hamilton, qui impliquait la recherche d'un circuit Hamiltonien. La recherche d'un circuit hamiltonien a des applications réelles dans de nombreux domaines.
Dans cette partie, nous prouvons la contrôlabilité régionale pour les AC unidimensionnels et bidimensionnels en utilisant une méthode basée sur l'existence d'un circuit hamiltonien. L'AC est régionalement contrôlable si tout les états sont atteignables dans la région contrôlée c'est à dire qu' à partir de chaque sommet, nous pouvons atteindre un autre sommet en un nombre fini de pas. L'existence d'un circuit hamiltonien assure que tous les sommets qui représentent des configurations, sont visités une seule fois et assure qu'il existe un temps T pour lequel toutes les configurations sont accessibles.

Définition 13 *[85] Un circuit hamiltonien d'un graphe $G = (V, AR)$ est un chemin orienté de G qui inclut chaque sommet exactement une fois.*

Et nous obtenons le résultat suivant :

Théorème 6 *Un automate cellulaire est régionalement contrôlable si et seulement s'il existe un temps t tel que le graphe associé à la matrice de transition $\mathcal{C}^t$ contient un circuit Hamiltonien.*

Comme le problème de la preuve de l'existence d'un circuit hamiltonien dans un graphe est NP-complet, la complexité temporelle peut être exponentielle en fonction du nombre de sommets du graphe de transition. Nous améliorons ce critère dans la partie suivante avec une solution en temps polynomial qui donne une condition nécessaire et suffisante.

Condition nécessaire et suffisante : composante fortement connexe

En théorie des graphes, un graphe orienté est composé de noeuds. Ces noeuds ne sont généralement pas reliés entre eux par des arcs, mais dans certains cas, il existe un chemin qui relie chacun des deux sommets du graphe. Nous introduisons pour cela la notion de composante fortement connexe.

Dans cette section, nous donnons une condition nécessaire et suffisante pour le problème de la contrôlabilité régionale en utilisant la notion de composante fortement connexe.

Définition 14 *[85] Une composante fortement connexe (CFC) d'un graphe orienté G est un sous-ensemble maximal de sommets de $C \subset V$ tel que pour chaque paire de sommets v_1 et v_2, il existe un chemin orienté de v_1 jusqu'à v_2 et de v_2 jusqu'à v_1.*

D'où le résultat suivant :

Théorème 7 *Un AC est régionalement contrôlable pour une règle donnée si le graphe de transition Υ associé à la matrice $\mathcal{C}$ n'a qu'une seule CFC.*

Le critère communément utilisé pour décider si un système est contrôlable ou non est la condition de Kalman. Ce critère ne donne aucune information sur les contrôles utilisés pour obtenir un état désiré donné à partir d'un état initial. L'approche utilisée dans ce chapitre et qui est basée sur la recherche de préimages, permet d'obtenir les contrôles recherchés. Dans cette partie nous donnons un algorithme efficace pour la création des préimages.

Pré-images d'une région contrôlée

Soit $\{s_1^i, s_2^i, \ldots, s_n^i\}$ la configuration au temps i de la zone ciblée par le contrôle et Soient ℓ et r les contrôles sur la frontière de cette zone.

L'idée est de trouver un contrôle frontière (de la région contrôlée) donné sous la forme d'une séquence $(\ell^0, r^0), (\ell^1, r^1), \ldots, (\ell^{T-1}, r^{T-1})$ telle l'on obtienne une configuration souhaitée $\{s_1^T, s_2^T, \ldots, s_n^T\}$ au temps T et ceci en partant d'une configuration initiale $\{s_1^0, s_2^0, \ldots, s_n^0\}$ donnée.

Définissons dans ce qui suit quelques notions utiles et présentons la structure de données nécessaire pour résoudre le problème.

Fonction distance

Nous définissons la fonction distance

$$\Delta_i : \text{sommet} \mapsto \text{liste de sommets}$$

qui associe à chaque sommet $v \in \Upsilon$, la liste de sommets à partir desquels v peut être atteignable dans un chemin de longueur i. Υ est le graphe de transition introduit dans le chapitre précédent, assosié à la matrice $\mathcal{C}$. Ainsi $\Delta_i(v)$ donne toutes les configurations initiales à partir de laquelle la configuration désirée v peut être atteignable en i pas en appliquant le contrôle.

Soit T le temps auquel nous voudrons atteindre un état désiré. En représentant Δ comme une application, nous allons successivement construire les fonctions Δ_i en recherchant les prédécesseurs à partir de Δ_{i-1}. On commence par Δ_1 et on continue jusqu'à Δ_T. Si δ est le nombre maximum de prédécesseurs d'un sommet, la complexité par rapport au temps est $O(T \times \delta \times |V|) = O(T \times \delta \times 2^{|\omega|})$.

Chemin pour la contrôlabilité

Nous abordons deux problèmes dans cette section.

Problème 1

Trouvez une configuration (d'état) qui peut être amenée vers une configuration (d'état) souhaitée b_f au bout de k pas de temps et trouver la séquence de contrôles requise.

Pour résoudre ce problème, nous construisons la fonction de distance Δ_i pour $1 \leq i \leq k$ puis nous considérons les l'ancêtres b_i à distance k de b_f (stocké dans $\Delta_k(b_f)$). S'il n'y a pas un ancêtre, cela signifie qu'il n'est pas possible d'atteindre cet état en k pas. Sinon, on peut trouver le chemin de longueur k dont le sommet final est b_f. Pour ce faire, nous choisissons un prédécesseur de b_f, disons b_{k-1}, qui peut être atteignable en $k-1$ pas, *i.e.* un des sommets de la liste $\Delta_{k-1}(b_f)$. Ensuite, nous cherchons le premier parmi les prédécesseurs suivants, disons, b_{k-2}, qui peuvent être atteignables en $k-2$ pas, *i.e.* l'un des sommets dans la liste $\Delta_{k-2}(b_f)$ et ainsi de

suite jusqu'à ce que nous trouvons b_1. Nous obtenons le chemin des configurations $b_1, \ldots, b_{k-1}, b_f$. Il ne reste qu'à trouver le contrôle approprié en appliquant des règles à chaque configuration avec le contrôle sur la frontière $(0,0), (0,1), (1,0), (1,1)$. Au total, une fois la fonction de distance construite, le coût pour obtenir les contrôles est en $O(k)$.

problème 2

Trouvez tous les contrôles nécessaires pour obtenir un état désiré b_f en k pas d'une configuration b_1.

Pour ce problème, au lieu de vérifier si la liste des ancêtres n'est pas vide (et prendre un parmi les sommets), nous devons vérifier si parmi les ancêtres il y a la configuration initiale b_1. Par conséquent, la complexité temporelle est en $O(k \times \delta)$.

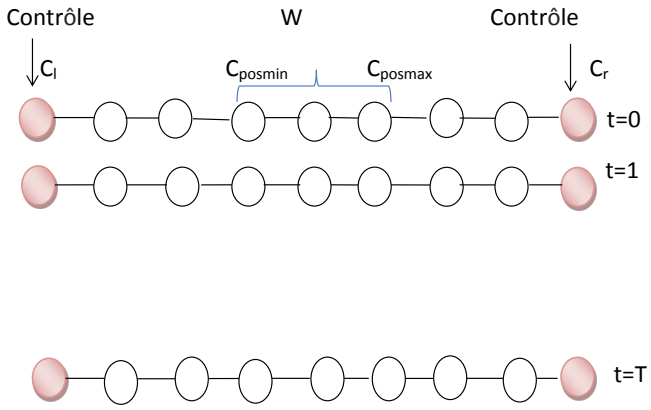
Chapitre 6 : Condition de Kalman et approche algorithmique pour la contrôlabilité régionale des Automates Cellulaires par des actions frontières sur le domaine

R. Kalman a identifié la notion de contrôlabilité comme l'une des propriétés principales pour déterminer le comportement d'un système dynamique. L'étude de la contrôlabilité pour les systèmes linéaires exige l'utilisation de la condition de rang qui est devenue omniprésente dans l'analyse des systèmes linéaires. Nous abordons dans ce chapitre le problème de la contrôlabilité régionale frontière des AC Booléens élémentaires (1D).

Considérons un AC Booléen défini sur un réseau $\mathcal{L}_c$, supposé fini et composé de N cellules intérieures et de 2 cellules sur les frontières. Nous désignons par c_l et c_r respectivement les cellules sur les frontières (gauche et droite).

Soit $\omega = \{c_{posmin}, \ldots, c_{posmax}\}$ un sous-ensemble de l'espace cellulaire unidimensionnel où la configuration souhaitée doit être atteinte. Nous sommes intéressés par trouver les séquences appropriées de contrôles requis sur la frontière de l'AC, $(u_l^0, u_l^1, \ldots, u_l^{T-1})$ et $(u_r^0, u_r^1, \ldots, u_r^{T-1})$, de manière à faire passer le système d'un état initial donné s_0 à une configuration souhaitée s_d sur la sous-région ω à un temps

donné T, tel que : $s_T(c_i) = s_d(c_i) \quad \forall i = posmin, \ldots, posmax$, où $s_T(c_i)$ et $s_d(c_i)$ sont respectivement l'état final et l'état désiré de la cellule c_i au temps T. La configuration souhaitée s_d est supposée être atteignable dans l'évolution de la règle de l'AC.



Contrôlabilité régionale frontière

La dynamique linéaire globale de l'AC qui régit l'évolution de l'AC à partir d'une configuration initiale donnée sur des pas de temps discrets est associée à une fonction de transition locale linéaire qui calcule l'état $(t+1)^{\text{ème}}$ de la cellule c_i par :

$s_{t+1}(c_i) = f(s_t(c_{i-k}), \cdots, s_t(c_{i+k})) = a_{-k}s_t(c_{i-k}) \oplus a_{-k+1}s_t(c_{i-k+1}) \cdots \oplus a_k s_t(c_{i+k})$, $a_i \in \{0,1\} \quad \forall i = -k, \ldots, k$ où $\oplus$ désigne l'opération XOR ou "somme mod 2" et k est le rayon de voisinage ($r = 2k+1$ désignant la taille du voisinage).

Définition 15 *Une fonction f définie par*

$$s' = f(s_{i-k}, s_{i-k+1}, \ldots, s_i, s_{i+k-1}, s_{i+k})$$

est linéaire par rapport aux variables périphériques si

$$\frac{\partial s'}{\partial s_{i+k}} = 1 \quad OU \quad \frac{\partial s'}{\partial s_{i-k}} = 1$$

on dit qu'il est doublement linéaire par rapport aux variables périphériques si

$$\frac{\partial s'}{\partial s_{i+k}} = 1 \quad ET \quad \frac{\partial s'}{\partial s_{i-k}} = 1$$

Contrôlabilité en dimension finie

Considérons un système linéaire qui peut s'écrire comme suit :

$$\begin{cases} z'(t) = Az(t) + Bu(t); 0 < t < T \\ y(t) = Cz(t) \end{cases} \tag{1.7}$$

Où $z \in \mathbb{R}^n$ est le vecteur d'état, $y \in \mathbb{R}$ est le vecteur de sortie et $u \in L^2[0, t; \mathbb{R}^m]$ est le vecteur de contrôle. Les matrices A, B et C sont de dimension $n \times n$, $n \times m$ et $p \times n$, respectivement, et représentent la dynamique, le contrôle et l'observation. L'étude du problème de la contrôlabilité consiste à trouver une réponse à la question suivante : Est-il possible d'atteindre tous les états désirés à partir de n'importe quel état initial. En dimension finie, la caractérisation de la contrôlabilité est liée à la détermination de la matrice de contrôlabilité qui peut être définie comme suit :

Définition 16 *Considérons le système 1.7, la matrice de contrôlabilité est une matrice de dimension $n \times nm$ définie par :*

$$\mathcal{C}_K = [B, AB, \ldots, A^{n-1}B]. \tag{1.8}$$

La détermination de la matrice donne une information indiquant si le système est contrôlable ou non. Nous avons le théorème suivant.

Théorème 8 *(La condition de Kalman)*

Le système 1.7 est contrôlable si et seulement si la matrice de contrôlabilité est de rang plein, autrement dit:

$$rang(\mathcal{C}_K) = n. \tag{1.9}$$

Résultat de caractérisation

Considérons un automate cellulaire linéaire qui peut s'écrire comme suit:

$$s^{t+1} = Js^t \oplus BU^t,$$

où $s_\omega^{t+1} = \{s^{t+1}(c_1), s^{t+1}(c_2), \ldots, s^{t+1}(c_n)\}$, $s_\omega^t = \{s^t(c_1), s^t(c_2), \ldots, s^t(c_n)\}$

$B_{|\omega}$ est une matrice de taille $(n \times 2)$ et U^t est le vecteur contrôle de taille (2×1):

$$B_{|\omega} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \qquad U^t = \begin{pmatrix} u^t(c_0) \\ u^t(c_{n+1}) \end{pmatrix} \tag{1.10}$$

où $u^t(c_0)$ est le contrôle appliqué sur la cellule c_0 et $u^t(c_{n+1})$ est le contrôle appliqué sur la cellule c_{n+1}.

Théorème 9 *Un Automate cellulaire unidimensionnel linéaire est régionalement contrôlable en exerçant des actions frontières si et seulement si :*

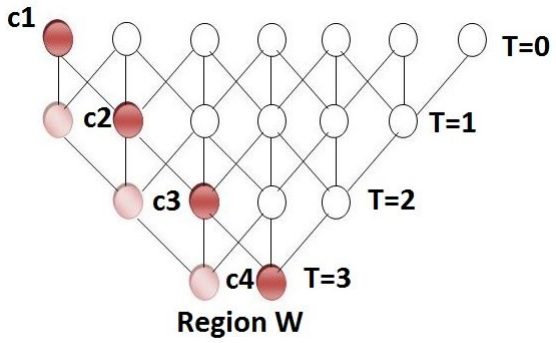
$$Rang(\mathcal{C}_K) = Rang(B \quad JB \quad J^2B \ldots J^{T-1}B) = T = n - 1$$

où T est l'horizon de temps, n est la taille de la région contrôlée et J est la matrice jacobienne.

Nouvel algorithme pour la contrôlabilité régionale frontière d'un Automate Cellulaire linéaire

Pour contrôler les règles linéaires par rapport aux variables périphériques, il suffit d'appliquer le contrôle d'un des deux côtés [12]. Il est donc possible d'atteindre l'état désiré dans une région en appliquant le contrôle à partir d'une seule frontière comme

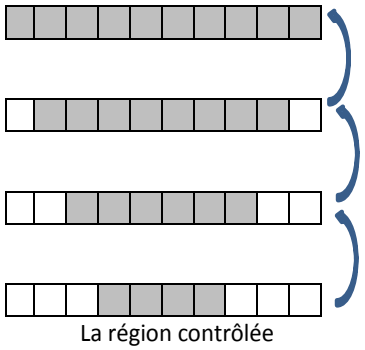
illustré dans la figure suivante qui illustre le contrôle de la région ω en appliquant une action sur la frontière c_1. Le changement dans la configuration initiale sur la cellule contrôlée c_1 va se propager à la cellule c_4 au temps $T = 3$.:



Nous avons proposé un algorithme qui permet de trouver le contrôle requis sur les frontières de l'AC afin d'atteindre un état désiré sur une région cible. Cet algorithme permet de contrôler aussi le complément des règles linéaires.

Approche préimages pour le contrôle régional des Automates Cellulaires non linéaires

Le comptage des préimages de l'AC unidimensionnel a été étudié dans plusieurs ouvrages par de nombreux auteurs comme Jen voir [66] en 1989 et Voorhees en 1993 [92] et autres dans [67, 81, 82] en utilisant l'algorithme de Brujuin. Dans cette section, nous étudions le problème de la contrôlabilité régionale en exerçant des actions frontières sur l'AC non linéaire, en proposant un nouvel algorithme pour générer toutes les préimges afin de tester numériquement la contrôlabilité. Il faut vérifier en principe la possibilité d'atteindre la configuration initiale à partir de la configuration désirée en recherchant les séquences requises de contrôles : $(u_l^0, u_l^1, \ldots, u_l^{T-1})$ et $(u_r^0, u_r^1, \ldots, u_r^{T-1})$ qui génèrent la configuration cible dans la région contrôlée.



Chapitre 7: Contrôle régional des Automates Cellulaires probabilistes

Dans ce chapitre, nous étudions le problème de la contrôlabilité régionale des automates cellulaires probabilistes unidimensionel en agissant sur la frontière de la région contrôlée. Nous utilisons l'approche des chaînes de Markov.

Les automates cellulaires probabilistes peuvent être considérés comme une extension des automates cellulaires déterministes où la fonction de transition donne la probabilité que le noeud cible se trouve dans un certain état. Si toutes ces probabilités sont égales à zéro ou à un, l'AC probabiliste se réduit à un automate cellulaire déterministe.

Position du problème

Nous traiterons ici principalement le problème de contrôle régional en agissant sur les frontières de la région contrôlée.

Considérons maintenant le problème du calcul de la probabilité $\mathcal{C}_{sy}(a,b) = \mathcal{C}(s|y;a,b)$ qui est la probabilité conditionnelle qui permet d'obtenir la configuration s au moment $t+1$ en connaissant la configuration y au moment t et les frontières a et b. La matrice de contrôle de Markov $\mathcal{C}(a,b)$ est donnée par

$$\mathcal{C}_{\boldsymbol{sy}}(a,b) = \tau(s_1|a,y_1,y_2)\tau(s_2|y_1,y_2,y_3)\ldots\tau(s_n|y_{n-1},y_n,b),$$

où n indique la taille de la région cible et $\tau(s_1|a,y_1,y_2)$ est la probabilité d'atteindre s_1 partant du voisinage (a,y_1,y_2).

Pour une séquence de contrôle donnée, $\boldsymbol{a} = a_1,\ldots,a_T$ et

$\boldsymbol{b} = b_1,\ldots,b_T$, la matrice de contrôle de Markov qui en résulte pour le temps T est

$$\mathcal{C}(\boldsymbol{a},\boldsymbol{b}) = \prod_{t=1}^{T}\mathcal{C}(a_t,b_t).$$

Nous pouvons définir plusieurs problèmes de contrôle. Un premier concerne l'ergodicité : quelle est la meilleure séquence de contrôle a et b de sorte que $\mathcal{C}_{sy(a,b)} > 0$ pour toutes les paires s,y et en temps minimal T ? Un autre problème est celui de trouver la meilleure séquence de contrôles $\boldsymbol{a}$ et $\boldsymbol{b}$ qui maximise $M_{\boldsymbol{sy}}(\boldsymbol{a},\boldsymbol{b}) > 0$?

Évidemment, on peut aussi être intéressé par éviter certaines configurations, par exemple, si $s_i = 1$ représente la présence d'un animal ou d'une plante en position i au temps t, on peut être intéressé à concevoir un contrôle qui empêche l'extinction des animaux, c'est à dire, éviter le symbole $\boldsymbol{s} = 0$.

on pourrait être plutôt intéressé par déterminer *l'existence* d'une telle séquence, pour un certain intervalle de temps T, ou par trouver le temps minimum T pour lequel une séquence optimale existe.

En particulier, ce dernier problème peut être résolu avec moins d'effort de calcul que la recherche de la séquence réelle pour le meilleur contrôle. Si l'on considère la matrice

$$Q = \frac{1}{4}\sum_{a,b}\mathcal{C}(a,b) = \frac{1}{4}\big(\mathcal{C}(0,0)+\mathcal{C}(0,1)+\mathcal{C}(1,0)+\mathcal{C}(1,1)\big),$$

et qu'on calcule ensuite sa puissance Q^T, toutes les séquences de contrôle possibles de longueur T sont contenues dans une telle puissance. Par conséquent, le problème de l'existence d'une séquence de contrôle pour un temps donné T se réduit à vérifier si $(Q^T)_{\boldsymbol{xy}} > 0$.

On peut aussi quantifier l'efficacité du contrôle en calculant le rapport η entre les valeurs minimale et maximale de $\mathcal{C}$. Si ce rapport est nul, cela signifie qu'il y a

certaines paires de configurations qui ne peuvent être connectées par aucune séquence de contrôle, tandis que $\eta = 1$ signifie que toutes les paires de configurations peuvent être connectées avec la même facilité.

Le problème est que ces contrôles optimaux ont été obtenus par le balayage de toutes les séquences possibles. Pour un AC unidimensionnel avec deux frontières libres, on a un contrôle possible de 4 valeurs, et la recherche exhaustive implique la vérification des combinaisons de 4^T, où T est le temps de contrôle estimé, proportionnel à L. Une approximation du contrôle optimal peut être obtenue au moyen de la matrice Q, qui correspond à la superposition de tous les contrôles possibles. L'idée est de calculer la quantité $Q^{T-1}(\boldsymbol{s}|\boldsymbol{y})$ qui donne la possibilité d'obtenir $\boldsymbol{s}$ à partir de la configuration $\boldsymbol{y}$ en T pas de temps, pour certains contrôles, et ensuite rechercher le meilleur contrôle à l'étape finale T, soit maximiser

$$\mathcal{C}(a_T, b_T)Q^{T-1}$$

pour chaque paire $(\boldsymbol{s}, \boldsymbol{y})$ sur a_T et b_T, obtenant $\tilde{a}_T$ et $\tilde{b}_T$.

La procédure est ensuite répétée pour

$$\mathcal{C}(\tilde{a}_T, \tilde{b}_T)\mathcal{C}(a_{T-1}, b_{T-1})Q^{T-2}$$

et ainsi de suite.

Nous traitons aussi le problème d'éviter certaines configurations. Par exemple, dans le cas d'une population d'organismes, on pourrait être intéressé par éviter leur extinction, c'est à dire à indiquer $\mathbf{0}$. Dans ce cas, la quantité à maximiser est la suivante

$$\mathcal{C}(\boldsymbol{a}, \boldsymbol{b}) = \prod_{t=1}^{T} [\mathcal{C}^*(a_t, b_t) \odot \overline{Z}(a_t, b_t)]$$

où $\mathcal{C}^*$ est égal à $\mathcal{C}$ mais dans la colonne zéro ($\mathcal{C}^*_{\boldsymbol{s},\mathbf{0}} = 0$, c'est à dire, la configuration 0 a été rendue absorbante, indépendamment des contrôles), et $\overline{Z}_{\boldsymbol{s},\boldsymbol{y}} = 1 - \mathcal{C}_{\mathbf{0},\boldsymbol{y}}$,

soit la probabilité de ne pas atteindre $\mathbf{0}$ partant de $\boldsymbol{y}$. Le symbole $\odot$ indique le produit Hadamard (composante par composante).

De cette façon, nous maximisons la probabilité de passer d'une configuration $\boldsymbol{s}$ à une configuration $\boldsymbol{y}$ durant T pas de temps, sans passer par la configuration $\mathbf{0}$.

CHAPTER 2

INTRODUCTION

Control theory is a branch of applied mathematics which deals with the behaviour of dynamical systems. It starts to emerge in 1960 when firstly introduced by Kalman and Bellman [16, 69] and followed in several works by Butkovskiy [19] and Lions [76, 77]. It considers the problem of forcing the appearance of a desired output of dynamical systems within a finite or infinite time horizon. So given an intended objective, the raised question is how can we act on the domain (or its boundary) of a dynamical system in order to guarantee to achieve this objective.

One of the basic ingredient for control theory problems is the establishment of an adequate model that constitutes the main prerequisite for the understanding of the behaviour of many real phenomena. The derivation of such a model generally requires considerable physical insight and could be helped by observations and data.

For several theoretical and computational issues in control theory, one needs to write a mathematical model based on ordinary differential equations, partial differential equations, depending on the considered phenomena. The control theory has applications in different fields: astronomy, mechanical engineering, social science, medicine, physics, biology. [64, 83].

Among the most prominent and main considered issues in control theory there

are the notions of controllability and observability which were introduced by Kalman and developed during the last two centuries. The controllability played an important role through the history of control theory. It deals with the ability of designing a control input so as to lead the system to a desired output within a predefined time interval $[0, T]$. The concept of controllability has been widely studied for finite dimensional systems [78, 86]. Another important notion is the observability which refers to the possibility to reconstruct the initial state given a knowledge of the outputs [75]. Controllability and observability have a dual relationship and constitute the basic concepts in control theory.

When the studied systems take into account the spatial and temporal changes and interact with their environment through inputs and outputs variables depending on space and time, we speak about Distributed Parameter Systems (DPS). They are usually described using partial differential equations (PDE) with various boundary and initial conditions and also various kinds of inputs and outputs. The research on DPS has received extensive attention as many industrial processes are considered DPS as they exhibit spatial variations in their underlying physical phenomena, such as diffusion, convection, phase-dispersion and flow.

A wide variety of works related to controllability and observability of distributed systems parameters has been studied [39, 76, 77]. The study of these notions on DPS via the structure of actuators and sensors were the subject of intense activities [14, 17, 18, 39, 41, 42].

In the last three decades, the general control and analysis problems on DPS have been considered for a particular situation encountered in many real world problems where one is interested in achieving some objectives only on specific parts of the domain. So it is no need to spend efforts on the whole domain when only a subregion is concerned with the objective. It turns to be necessary to reformulate all the classical notions related to systems analysis and control for regional considerations. In particular, the so-called regional controllability and observability have been introduced by El Jai and Zerrik [102] and well studied in several following works [40, 43]. For DPS, the term regional has been used to refer to control problems in which the de-

sired state is only defined and may be reachable on a portion of the spatial domain. In many physical problems the regional controllability occurs naturally in order to model a natural phenomenon just in a sub region of the whole domain. For instance, we may be interested in keeping the temperature of a device under a specific degree on a part of the domain, it is better and cheaper to control the system only on this region. Furthermore, the system may be not controllable in the whole domain but regionally controllable. In the same way, the concept of regional observability has been studied in [7, 44] and also a deep study on the regional analysis via actuators and sensors has been developed particularly within the Moroccan systems theory network coordinated by El jai and Zerrik for continuous as well as discrete time systems [3, 4, 100, 101].

Despite the importance in terms of quality and quantity of the work carried out in this field, the PDEs based models used so far, have shown significant limitations regarding the highly complex behaviour of nonlinear dynamical systems. The investigation of new modelling approaches for DPS turns to be necessary. In this context, cellular automata appear to be a good alternative to the classical models described by PDEs. They were studied for the first time in the context of systems theory by El Yacoubi and El Jai in the late 90s [45, 47, 49, 51, 52].

Cellular Automata (CA for short) are the simplest models of spatially extended systems which provide a good tool for describing a complex phenomena. They are completely discrete dynamical systems where space and time are discrete but also the physical quantities (states) take values in a finite set. The whole evolution is governed by a set of a predefined local transition functions describing the microscopic dynamics. Although these dynamics are interpreted in terms of simple rules, CA are able to exhibit very complex and unpredictable phenomena. Beyond their relevance as general models of complexity, CA have proved useful as more specific representations of non-linear dynamics in a variety of scientific fields. An extensive literature has been dedicated to applications in biology, chemistry, physics, ecology for which CA models have been successfully used [23, 24, 61, 88, 96]. They present an ideal framework for studying complex systems related to cross-cutting issues.

CA have been investigated in the context of systems theory in the late 90s where control and observation variables were introduced in an appropriate way in CA models in order to make them more useful in control theory. Some results related to additive CA can be found in [46, 48, 53].

Some regional control problems using CA have been carried out, see for instance [12, 15, 46]. In [49], regional control problem has been studied for 1D and 2D deterministic additive CA, by exploring a numerical approach based on genetic algorithms. A first approach of boundary regional control of 1D Boolean CA was investigated in [12].

The results obtained by El Yacoubi and Bagnoli in [12] will constitute a starting point for the developed work in this thesis. The idea is to explore different formalisms and approaches, some of which are specific to PDEs, that can prove the regional controllability of CA. Focusing first on Boolean deterministic CA that are spatially-extended highly non-linear dynamical systems, we investigate the possibility of controlling a subdomain by acting on its boundaries and using the concept of Boolean derivatives. Both the linear and nonlinear CA rules were considered. The classical Kalman rank condition for the finite dimension was generalised to CA and allowed to prove the regional controllability. Another approach based on graph theory was also investigated and necessary and sufficient conditions were obtained for 1D and 2D CA [37]. Finally the case of probabilistic CA was examined considering its actual stochastic dynamics. In this case one cannot guarantee to be able to drive the system towards a given state in a given time, but one can increase the probability of reaching that state. The evolution of a probabilistic CA can be seen as a Markov chain, where the elements of the transition matrix are given by the product of the local transition probabilities [10, 11]. The Markov chain approach was also extended to deterministic CA as they are the limits of probabilistic CA when the transition probabilities go to zero or one [35, 36].

This thesis is divided into 6 chapters. While chapter 2 gives a general introduction of the thesis with an overview on the basic literature in the subject, chapter 3 will give the bases of control theory and present CA as a counterpart of PDE for studying spatio-temporal systems. Chapter 4 is devoted to study the problem of

regional controllability of CA by exploring a new approach based on Markov chains. In chapter 5, another approach based on a graph theory has been proposed for proving the regional controllability of CA. Chapter 6 discusses the problem of boundary regional controllability of one dimensional CA where we have proposed some algorithms and used the Kalman condition. Original results on probabilistic CA will be presented in chapter 7 and finally some conclusions and an outlook for future work are given in chapter 8.

Let us give an overview of the thesis:

Chapter 3 In this chapter, we recall some notions of control theory which will be used in the work performed in the following chapters. We have started by a short overview on distributed parameter systems described by partial differential equations, then we have introduced cellular automata which are considered as a good alternative of PDE for studying DPS. We give some examples which illustrate that Cellular automata can describe many real problems.

Chapter 4 This chapter is devoted to study the problem of regional controllability of one dimensional and two dimensional cellular automata by acting on the boundaries of the controlled region. This study strives to highlight a new approach that is of Markov chains. We have proved the regional controllability by using the definition of regular and ergodic Markov chain. At the end, we have proved that for a given time t if the constructed matrix contains an absorbing state then the system is not regionally controllable.

Chapter 5 This chapter is dedicated to the regional controllability problem of one dimensional and two dimensional cellular automata by exploiting the link between CA and graphs. A new approach based on graph theory has been proposed. We have proved the regional controllability through the notions of Hamiltonian circuit and strongly connected component. We finished by proposing an algorithm for creating the preimages in order to find the required control that allows to reach the desired configuration.

Chapter 6 We tackled in this chapter, the problem of boundary regional controllability of one dimensional cellular automata by exploring new tracks of proofs

and included some algorithms. We have started by the linear case where we have proposed an algorithm which allows to find the required control on the boundaries in order to force the appearance of the desired configuration on the controlled region. Another algorithm has been proposed for the non linear case. We finished by a characterisation result based on the well known Kalman condition.

Chapter 7 This chapter is devoted to the regional controllability problem of probabilistic cellular automata using the same approach based on the Markov chains. We have proved the regional controllability of probabilistic CA. Moreover, We have proposed a method to find the optimal control and the faster suboptimal control. We finished by an example of optimal control in the case where one wants to avoid some configurations.

Chapter 8 This chapter concludes the work carried out in this thesis and gives some future directions.

From Distributed Parameter Systems to Cellular Automata

CHAPTER 3

FROM DISTRIBUTED PARAMETER SYSTEMS TO CELLULAR AUTOMATA

Résumé:

Dans ce chapitre, on a commencé par introduire les systèmes à paramètres distribués décrits par des équations aux dérivés partielles notés (SDP). On a rappelé brièvement quelques notions de la théorie du contrôle en considérant le cas linéaire d'un SDP. On a donné une vue d'ensemble sur les notions utilisées dans cette thèse telles que la contrôlabilité et la contrôlabilité régionale.
Au vu des difficultés rencontrée lors de l'analyse des systèmes à paramètres distribués il est devenu nécessaire de chercher des approches alternatives et ainsi les automates cellulaires se sont imposés comme alternatives aux équations aux dérivées partielles (EDP). Une introduction des automates cellulaires ainsi que leurs propriétés générales sont présentées. Quelques exemples pour mettre l'accent sur l'efficacité des AC sont ensuite donnés. À la fin, nous présentons l'essentiel d'une étude faite par El Yacoubi dans [48] sur les automates cellulaires en relation avec la théorie des systèmes.

3.1 Overview on Distributed Parameter Systems

Definition 1 *Distributed parameter systems (DPS) are systems in which the variables depend on time and space. They are defined by operators (A, B, C) where A defines the dynamics of the system, B is the control operator describing the inputs and C is the observation operator in terms of outputs. They are usually described by partial differential equations that can be linear or nonlinear, discrete or continuous, deterministic or stochastic. The study of DPS by means of PDE requires some mathematical tools that will be presented in this chapter. Let us start first with the notion of semi-group in terms of which the solution of PDEs will be expressed.*

3.1.1 Semi-group

Definition 2 [5] *A strongly continuous semi-group is a family $(\phi(t))_{t\geq 0}$ of operators in $\mathcal{L}(Z)$ (where Z is the state space) satisfying the following properties:*

1. $\phi(0) = I$.
2. $\phi(t+s) = \phi(t)\phi(s)$, *for all* $t, s \geq 0$.
3. $\lim \mid\mid \phi(t)z - z \mid\mid \longrightarrow 0$ *when* $t \longrightarrow 0^+$, $\forall z \in Z$.

Thus $\phi : \mathbb{R}^+ \to \mathcal{L}(Z)$ associates to every $t \geq 0$, an operator $\phi(t) : Z \to Z$. The family of operators $(\phi(t))_{t\geq 0}$ depends obviously on A describing the dynamics. Additionally we have the following definition:

Definition 3 [5]

The infinitesimal generator of the semi-group $\phi(t)_{t\geq 0}$ is the operator A defined by

$$Az = \lim_{t\to 0^+} \frac{\phi(t)z - z}{t} \tag{3.1}$$

when the limit exists.
The domain of A, denoted by $D(A)$, is the set of $z \in Z$ such as the previous limit exists.

$$D(A) = \left\{ z \in Z \quad / \quad \lim_{t\to 0^+} \frac{\phi(t)z - z}{t} \quad exists \right\}$$

Some properties of semi groups are given in the following proposition.

Proposition 3.1.1 *[5] Let $\phi(t)_{t\geq 0}$ be a strongly continuous semi-group on Z with an infinitesimal generator A. Then for all $z_0 \in D(A)$, we have the following proprieties.*

1. *$\phi(t)z_0 \in D(A)$, for all $t \geq 0$.*
2. *$\frac{d}{dt}(\phi(t)z_0) = A\phi(t)z_0 = \phi(t)Az_0$, for all $t > 0$.*
3. *$\frac{d^n}{dt^n}(\phi(t)z_0) = A^n\phi(t)z_0 = \phi(t)A^n z_0$, for all $t > 0$, $z_0 \in D(A)$ and $n \geq 0$.*
4. *$\phi(t)z_0 - z_0 = \int_0^t \phi(s)Az_0 \, \mathrm{d}s$, for all $t > 0$.*
5. *For each $n \geq 1$, $D(A^n)$ is dense in Z if A is closed.*

Let A be a bounded operator $A \in \mathcal{L}(Z)$, one can define a strongly continuous semi-group $\phi(t)$ such that A is its infinitesimal generator:

$$\phi(t) = e^{At} = I + At + \frac{A^2t^2}{2!} + \cdots + \frac{A^nt^n}{n!} + \ldots$$

In the unbounded case, we need A to be closed and thus its domain should satisfy $\overline{D(A)} = Z$. More precisely, we have the following theorem [5]:

Theorem 3.1.2 *If an operator A is closed, with a domain $D(A)$ dense in Z, and satisfies the conditions 1) and 2)*

1. *$(\lambda I - A)^{-1}$ exists for $\lambda > \alpha$*
2. $$|| \, (\lambda I - A)^{-m} \, || \leq \frac{M}{(\lambda - \alpha)^{-m}}, m = 1, 2, \ldots \tag{3.2}$$

   then A generates a strongly continuous semi-group $(\phi(t))_{t\geq 0}$ satisfying

$$|| \, \phi(t) \, || \leq Me^{\alpha t}, t > 0 \tag{3.3}$$

In the particular case where A has a complete set of orthogonal eigen-functions (φ_{n_j}) in Z associated to the eigenvalues (λ_n), and λ_n is of multiplicity r_n, then the semi group $(\phi(t))_{t\geq 0}$ generated by A can be expressed, for all $t > 0$ and $z \in Z$, by

$$\phi(t)z = \sum_n e^{\lambda_n t} \sum_{j=1}^{r_n} < z, \varphi_{n_j} > \varphi_{n_j} \tag{3.4}$$

Similar result is obtained for its adjoint A^*:

Proprety 3.1.3 *[5] The adjoint A^* of A generates the semi-group $(\phi^*(t))_{t\geq 0}$ adjoint of $(\phi(t))_{t\geq 0}$ which is also strongly continuous on the dual Z' of Z.*

3.1.2 Linear distributed system

In the linear case, a DPS is described by the following state equation:

$$\begin{cases} z'(t) = Az(t) + Bu(t); 0 < t < T \\ z(0) = z_0 \in D(A) \end{cases} \tag{3.5}$$

augmented with the output equation:

$$y(t) = Cz(t) \tag{3.6}$$

where

- The operator A generates a strongly continuous semi-group $(\phi(t))_{t\geq 0}$ on the state space Z,
- The operator $B \in \mathcal{L}(\mathbb{R}^{n_a}, Z)$ is the input operator,
- $C \in \mathcal{L}(Z, \mathbb{R}^{n_s})$ is the output operator.

n_a and n_s are the number of controls and measures respectively.
The semi-group plays a prominent role in the determination of the solution of an abstract differential equation. Particularly the solution of the system 3.5 is given by

$$z_u = \phi_A(t)z_0 + \int_0^t \phi_A(t-s)Bu(s)ds$$

3.1.3 Controllability

Introduction

and observability are among the two most fundamental concepts in control theory, Controllability plays an important role. Roughly speaking, controllability generally means, that it is possible to steer a dynamical system from an arbitrary initial state to an arbitrary final state using a set of admissible controls.

As opposed to a lumped parameter system, a DPS is a system whose state space is infinite-dimensional. Such systems are therefore also known as infinite-dimensional systems and usually described by partial differential equations or by delay differential equations.

In that case, we distinguish between two types of non-equivalent definitions of controllability: exact and approximate or weak controllability. It follows directly from the fact, that in infinite-dimensional spaces there exist linear subspaces which are not closed.

Definition 4 *[5] Let z_0 and z_d be given states in Z. The system 3.5 is said to be exactly controllable (or the pair (A, B) is controllable) on $[0, T]$ if for any states z_0 and z_d in Z, there exists a control $u \in L^2(0, T; \mathcal{U})$ such that $z(T) = z_d$.*

Figure .3.1 : Exact controllability

Definition 5 *[5] The system is said to be weakly controllable on $[0, T]$ if for all z_0 and $z_d \in Z$ and for all $\varepsilon > 0$, there exists a control $u \in L^2(0, T; U)$ such that*

$$|| \, z_u(T) - z_d \, ||_Z < \varepsilon$$

Characterisation result

Let H be the operator defined in $L^2(0, T, \mathcal{U})$ with its values taken in $Z = L^2(\Omega)$:

Figure .3.2 : Weak controllability

$$\begin{aligned} H : L^2(0,T;\mathcal{U}) &\rightarrow Z \\ u &\rightarrow Hu = \int_0^T \phi(T-s)Bu(s)ds \end{aligned}$$

Proposition 3.1.4 *[5] There is an equivalence between the two following assertions:*

- *The system 3.5 is exactly controllable,*
- $Im(H) = Z$.

Proposition 3.1.5 *[5] The three followings properties are equivalent:*

- *The system 3.5 is weakly controllable,*
- $\overline{Im(H)} = Z$,
- $Ker(H^*) = \{0\}$.

The proofs of these propositions can be found in [5] and the references therein.

More specifically to DPS analysis, abstract concepts of actuators and sensors were introduced and applied to the controllability and observability of systems described by PDEs [38]. Actuators and sensors are ubiquitous notions for describing the exchanges between the system and its environment. For the excitation of the system, one needs to define the actuators while to obtain an information on the system and its state evolution necessitates the definition of sensors. There are several types of actuators and sensors: zone or pointwise, internal or boundary.

Definition 6 *[5] An actuator is a couple (D, g) where $D \subset \Omega$ represents the domain that supports the actuator, and g is its spatial distribution.*

Definition 7 *[5] An actuator is said to be:*

- *zone actuator if D is a nonempty subregion of Ω. In this case, we assume that $g \in L^2(D)$.*
- *pointwise actuator if D is reduced to a point $b \in \Omega$. In this case, we have $g \equiv \delta_b$, where δ_b is a Dirac function concentrated at b. The actuator is denoted by (b, δ_b).*

Definition 8 *[5] An actuator (zone or pointwise) is said to be a boundary actuator if its support $D \subset \Gamma$. where Γ is the whole domain boundary.*

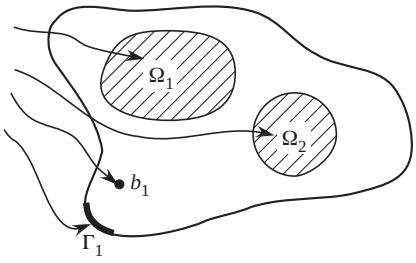


Figure .3.3 : Types of actuators

3.1.4 Regional controllability of distributed parameter systems

The regional controllability problem has been introduced by El jai and Zerrik in 1994 and has been studied afterward in several works [3, 4, 7, 43, 99]. In regional controllability problem we pay attention only on a region ω of the whole domain, see Figure (.3.4).
For regional controllability problem, one considers a system and asks if it is possible to steer it from an initial state to a predefined target state in the part ω of the domain. As we have seen before we distinguish two types of regional controllability.

Definition 9 *[5] The system 3.5 is said to be exactly regionally controllable on ω (or exactly ω-controllable) if, for all $z_d \in L^2(\omega)$, there exists a control $u \in L^2(0, T; \mathcal{U})$*

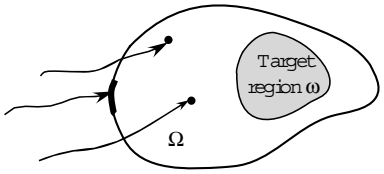


Figure .3.4 : Regional controllability

such that

$$p_\omega z_u = z_d$$

at time T *and with the application of the control* u.

$p_\omega z_u$ *is the restriction of the state* z_u *on* ω *which is defined as follows:*

$$\begin{aligned} p_\omega : L^2(\Omega) &\rightarrow L^2(\omega) \\ z &\rightarrow p_\omega z = z_{|\omega} \end{aligned}$$

Definition 10 *[5] The system 3.5 is said to be weakly regionally controllable on* ω *(or weakly* ω*-controllable) if, for all* $z_d \in L^2(\omega)$ *and* $\varepsilon > 0$*, there exists a control* $u \in L^2(0, T; \mathcal{U})$ *such that*

$$|| \, p_\omega z(T, u) - z_d \, ||_{L^2(\omega)} \leq \varepsilon$$

Beyond controlability, several aspects of regional analysis have been extensively studied for systems described by PDEs [4, 102, 103, 104].

However, the investigation of more innovative approaches with which natural systems that exhibit highly complex behaviours can be modelled and analysed, becomes clearly desirable. Even if sometimes there exist nonlinear or stochastic PDEs models for such complex dynamical systems, they remain very hard to analyse even through simulations. For this purpose, cellular automata seem to be good candidates.

3.2 Cellular Automata approach

3.2.1 A brief introduction

A cellular automaton is a discrete dynamical system in which space, time and the states are discrete. It is composed of adjacent elements (simple units) which called cells, arranged as a regular d-dimensional lattice evolving in discrete time steps. Each cell is characterised by an internal state whose value belongs to a finite set (this set can be numbers or properties). The updating of these states is done in general, simultaneously according to a common local transition rule that involves only a neighbourhood of each cell, [91].

In the one-dimensional case, a CA consists of a linearly connected array of cells denoted by $c_i, i \in \mathbb{Z}$, each of which takes the value in a finite set $S = \{0, 1, \cdots, q-1\}$ and a local transition function $f : S^r \longrightarrow S$ that acts on the state of a neighbourhood configuration described by $r = 2k+1$ variables where k defines the automaton range (neighbourhood radius). At each discrete time step t, all the cell state values denoted by s_i^t are updated in parallel as: $s_i^{t+1} = f(s_{i-k}^t, \cdots, s_{i+k}^t)$. The function f can be deterministic or probabilistic, synchronous or asynchronous, linear or nonlinear. For a given integer r and state number q, the possible configurations and the total number of rules are q^r and q^{q^r} respectively.
In the simplest case, each cell can have the binary states 0 or 1 so the Boolean cellular automata is defined. The cells are arranged in a regular paving called the lattice. Its topology usually corresponds to one dimensional line or two dimensional grid. All cells change their state simultaneously, using the so called transition function or the update rule (which can be the same in the case of uniform cellular automata) and according to the states of its nearby neighbouring cells states. The process is repeated at discrete time steps.

Before going in a deep study of CA, it may be worth to give some definitions of CA and their prominent features. For that, we start with the formal following definitions.

3.2.2 Formal definitions

Definition 11 *A cellular automaton (CA for short) is defined by a tuple* $\mathcal{A} = (\mathcal{L}_c, S, \mathcal{N}, f)$ *where:*

1. $\mathcal{L}_c$ *is a cellular space which consists in a regular paving of a domain* $\Omega \subset \mathbb{R}^d$, $d = 1, 2$. *Its geometry depends on the dimension and the form of its cells.*

2. S *is a finite discrete set of possible states.*

3. $\mathcal{N}$ *is a function that defines the neighbourhood of a cell* c *which locally determines the interaction of* c *with its environment. It is defined as follows:*

$$\begin{aligned} \mathcal{N} : \mathcal{L} &\rightarrow \mathcal{L}^r \\ c &\rightarrow \mathcal{N}(c) = (c_1, c_2, \ldots, c_r) \end{aligned}$$

   where c_i, $i = 1, \ldots, r$ *are the neighbouring cells and* r *is the size of* $\mathcal{N}(c)$.

4. f *is the transition function that allows to compute the state of a cell at time* $t+1$ *according to the state of its neighbourhood at time* t. *It is defined as follow:*

$$\begin{aligned} f : S^r &\rightarrow S \\ s_t(\mathcal{N}(c)) &\rightarrow f(s_t(\mathcal{N}(c))) = s_{t+1}(c) \end{aligned}$$

   where $s_t(c)$ *is the state of a cell* c *at time* t *and* $s_t(\mathcal{N}(c)) = \{s_t(c^{'}), c^{'} \in \mathcal{N}(c)\}$ *is the state of the neighbourhood of* c.

- There exist different types of cellular space as illustrated in Figure (.3.5) and Figure (.3.6):

- The neighbourhood can be connexe or not either it can change over the time according to the system. Furthermore, the neighbourhood may be punctured, i.e $c \notin \mathcal{N}$. For the two-dimensional CA we have different types of neighbourhood, see Figure (.3.7).

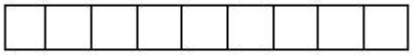
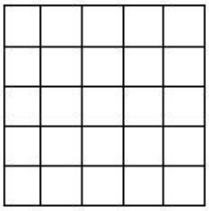
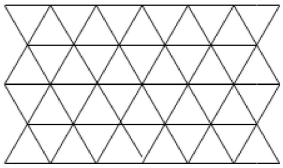


Figure .3.5 : One-dimensional rectangular lattice

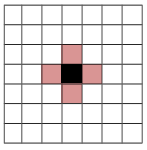


Two-dimensional rectangular lattice

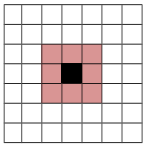


Triangular lattice

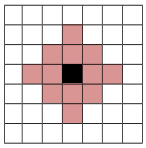
Figure .3.6 : Typical 2-D cellular space shapes.



Von Neumann



Moore



Modified Von Neumann

Figure .3.7 : Different types of neighbourhood

Definition 12 • *The configuration of a CA at time t corresponds to the set* $\{s_t(c), c \in \mathcal{L}_c\}$*, it is given by the following function:*

$$\begin{aligned} s_t : \mathcal{L}_c &\rightarrow S \\ c &\rightarrow s_t(c) \end{aligned}$$

- *The global dynamics of a CA is given by the function:*

$$\begin{aligned} F : S^{\mathcal{L}_c} &\rightarrow S^{\mathcal{L}_c} \\ s_t &\rightarrow s_{t+1} \end{aligned}$$

F maps to the configuration of CA at time t a new configuration at time $t+1$. The function F is related to f by the relation:

$$F(s_t) = f \circ s_t \circ \mathcal{N}$$

The global evolution of CA is given by the couple (s_0, F) where s_0 is the initial configuration and $s_t = F^t(s_0)$ is the configuration of CA at time t such as: $F^t = \underbrace{F \circ F \circ \ldots F}_{t \quad times}$

Among the crucial elements to define CA, we have the notion of the transition function or the update rule f which governs the evolution of the CA system. This function can be expressed by an analytical function, a table of cell-state transitions.

3.2.3 Initial and Boundary Conditions

In order to complete the spatial characterization of CA and as for each evolution system the initial and boundary conditions deserve to be specified. One needs to define them.

Firstly, we shall start with boundary conditions. To simulate the behaviour of CA, we must consider a finite number of cells and as we know that the cardinal of cellular space is finite the addressed question is: how can we define the neighbourhood of the cells at borders. This is because it is clear that they do not possess the same neighbourhood as other internal cells. For that purpose, we shall give the most commonly approaches used in the following figures.

1- Periodic conditions: In this case the two cells at extreme left and right ends to be neighbours of each other.

f	a	b	c	d	e	f	a

2- Reflective conditions: Here we consider the cells at the extremities left and right for which the states are repeated in order to provide neighbouring cell values for them.

a	a	b	c	d	e	f	f

3- Fixed conditions: They correspond to Dirichlet boundary conditions known in solving partial differential equations (PDE). The idea is to impose states values to a virtual cells such as these values (along the boundary) remain constant throughout the simulation.

1	a	b	c	d	e	f	1

Secondly, For doing a simulation, the starting point is the initial configuration which influence the CA evolution. It can be random or given (according to the problem). The initial condition can be expressed by the following function:

$$\begin{aligned} s_t : \mathcal{L}_c &\rightarrow S \\ c &\rightarrow s_0(c) \end{aligned}$$

3.2.4 Transition functions

Among the most prominent aspects in CA modelling we have the transition rule which stewards the transition between states and the evolution of the whole system. It can be expressed by an analytical function or a table of cell-state transitions, or a set of transition rules. It is called local transition function because it uses only the neighbourhood as inputs. Of course, it depends on the neighbourhood, the state set and also the lattice geometry. It can be deterministic or probabilistic.

- Deterministic Rules:

  Let $\mathcal{L}_c$ be a cellular space and $\mathcal{N}$ a type of neighbourhood. A function f describes a deterministic rule, if it is defined by:

$$\forall t \in I, \exists! \quad s_i \in S \quad such \quad as \quad f(s_t(\mathcal{N})) = s_i$$

where I is an interval of discrete time.

Some particular forms of deterministic rules encountered in real problems are as worth considering.

i. Additives rules:

A transition rule is said to be additive if

$$f(s_t(\mathcal{N}(c))) = \sum_{1\leq i\leq r} a_i s_t(c_i)$$

for some scalar $a_1, \ldots, a_r$ which are called the weights or coefficients of cells in the neighbourhood $\mathcal{N}$. The sum $\sum$ is considered modulo q, the cardinal of $\mathcal{S}$.

ii. Totalistic rules:

A transition rule f is totalistic if $a_i = 1, \quad \forall 1 \leq i \leq r$ or f can be written as the following form:

$$f(s_t(\mathcal{N}(c))) = \varphi(\sum_{c'\in\mathcal{N}} s_t(c')) = s_{t+1}(c')$$

Such functions give equal weight to all the cells in the neighbourhood which imply that the value of a cell state depends only on the sum of all previous state values of the neighbourhood cells.

- Probabilistic Rules:

They form an important class of transition rules. In this case the function f has more than one result for each type of neighbourhood i.e, f provides one or many possible values with associated probabilities. A probabilistic rule may be specified by a function.

$$f(s_t(\mathcal{N}(c))) \begin{cases} s^1 & \text{with} \quad \text{probability} \quad p(s^1|s_t(\mathcal{N}(c))), \\ \vdots & \vdots \\ s^m & \text{with} \quad \text{probability} \quad p(s^1|s_t(\mathcal{N}(c))), \end{cases}$$

where the probabilities are positive and satisfy the normalization condition

$$\sum_{s \in S} p(s|s_t(\mathcal{N}(c))) = 1$$

Generalized Rules

1. Non stationary rules:
   A rule is said to be non stationary if the cells update their states according to a rule which has the following form

   $$s_{t+1}(c) = g_t(s_t(\mathcal{N}_t(c)))$$

   where g_t and $\mathcal{N}_t$ depend on time. $g_t : S \times I \to S$, $\mathcal{N}_t : \mathcal{L}_c \times I \to \mathcal{L}_c^r$, I is a given interval, see [2] for more details.

2. Asynchronous rules:
   A rule is said to be asynchronous when the update of their states is not simultaneous. For more details see [20].

3. Rules with memory:
   The transition function is of type:

   $$s_{t+1}(c) = f(s_t(\mathcal{N}(c)), s_{t-1}(\mathcal{N}(c)), ..., s_{t-m}(\mathcal{N}(c)))$$

   where m is called the capacity of memory of the cell, with $m \geq 1$. This type of rules can be used for modelling the systems with delay.

3.2.5 Wolfram Rules

The most well known CA are the Boolean one-dimensional CA where there are two possible states such as $S = \{0, 1\}$ and the rules depend only on the current state of cell and the states of its nearest neighbours. The evolution of CA can be expressed by a table specifying the resulting state of a given cell according to the possibilities of states of it self and its neighbours (left and right).

In this case, we have 2^3 possible configurations of neighbourhood $000, 001, \ldots, 111$ and 2^8 different possible rules $00000000, \cdots, 11111111$. Each rule is labelled with a number between 0 and 255 that when written in base 2, gives an eight bit sequence corresponding to the outputs of the transition function. These CA are called elementary CA (ECA) and were extensively studied and empirically classified by Wolfram who proposed a scheme that has since became standard.

Examples of ECA:

Example 3.2.1 *Considering Wolfram rule 30 and the 8 bits binary expansion of the decimal number* 30 *which is* 00011110*. The ECA rule* 30 *can be represented in the following table:*

Current pattern (t)	*111*	*110*	*101*	*100*	*011*	*010*	*001*	*000*
Central cell State at $t+1$	*0*	*0*	*0*	*1*	*1*	*1*	*1*	*0*

and its illustrated evolution starting with the initial state (generation zero) in the first row, the first generation on the second row, and so on. Figure (.3.8) shows the first 50 generations of the rule 30 starting with an arbitrary initial condition.

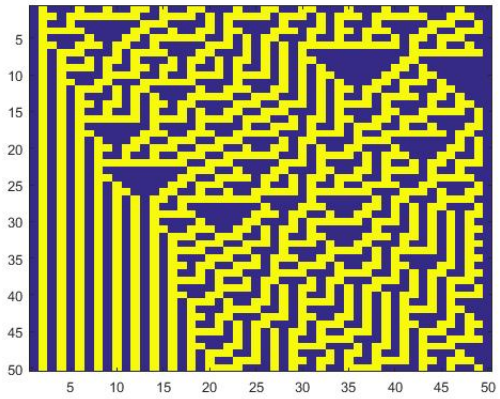


Figure .3.8 : Rule 30

It should be noted that ECA can also be expressed as a transition function that is defined in the case of rule 30 by:

$$s_{t+1}(c_0) = f(s_t(c_{-1}), s_t(c_0), s_t(c_{+1})) =$$
$$s_t(c_{-1}) + s_t(c_0) + s_t(c_{+1}) - s_t(c_0)s_t(c_{+1}) - 2s_t(c_{-1})s_t(c_0)s_t(c_{+1})$$

In fact, rule 30 is of special interest because it is chaotic and used as the random number generator.

Example 3.2.2 *Considering now Wolfram rule 90 where the 8 bit binary expansion of the decimal number* 90 *is* 01011010. *So the corresponding ECA has the local update rule:*

$$s_{t+1}(c_i) = s_t(c_{i-1}) \oplus s_t(c_{i+1})$$

Where $\oplus$ *is the XOR operator which means sum modulo 2. We can express the transition function of this rule either by a table:*

current pattern (t)	*111*	*110*	*101*	*100*	*011*	*010*	*001*	*000*
new state for center cell	*0*	*1*	*0*	*1*	*1*	*0*	*1*	*0*

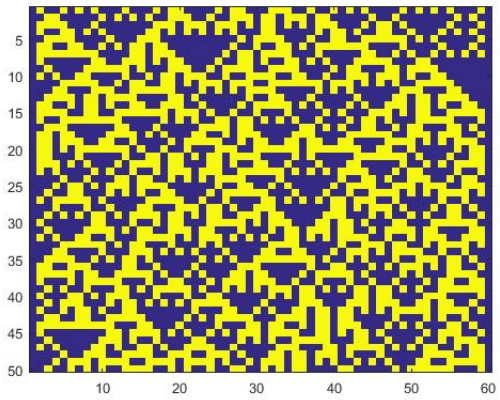


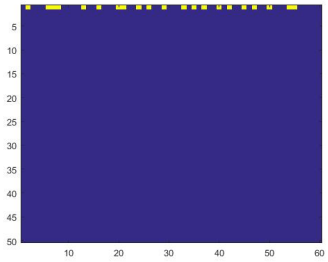
Figure .3.9 : Rule 90

Rule 90 *is an additive CA rule.*

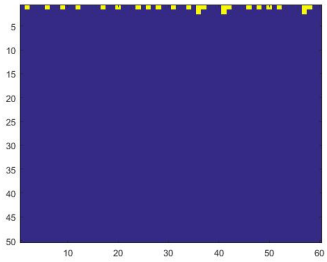
Classification of Wolfram The dynamical behaviour of CA sparked the interest of researchers regarding the complexity and the properties that generate. The study of CA as dynamical systems was pioneered by wolfram [95]. He has studied CA

extensively and their relationship to dynamical systems then he identified the classes of CA behaviour:

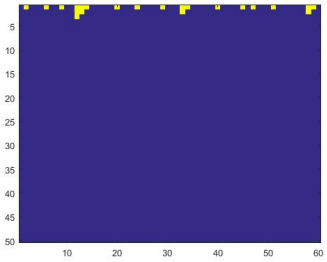
1. Class 1: Systems whose evolution relaxes to homogeneous states (fixed point). Among the rules of this class we have rule 0, rule 8 and rule 136.



Rule 0



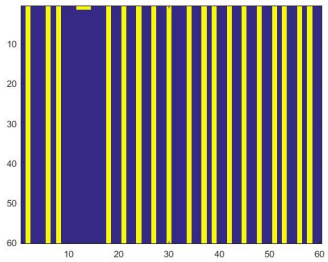
Rule 8



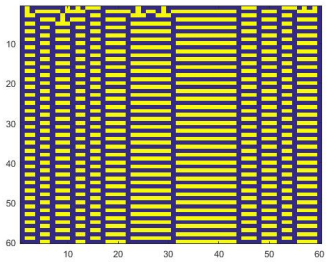
Rule 136

Figure .3.10 : Class 1 of Cellular automata, after a time evolution and starting from a random initial configuration, the evolution leads to a homogeneous state.

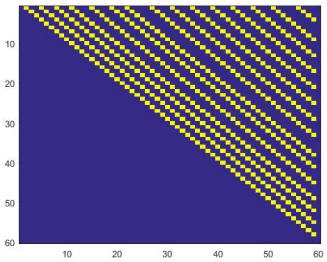
2. Class 2: Systems which converge to simple separated periodic structures (limit cycles). Among them we can cite rule 4, rule 37 and rule 56.



Rule 4



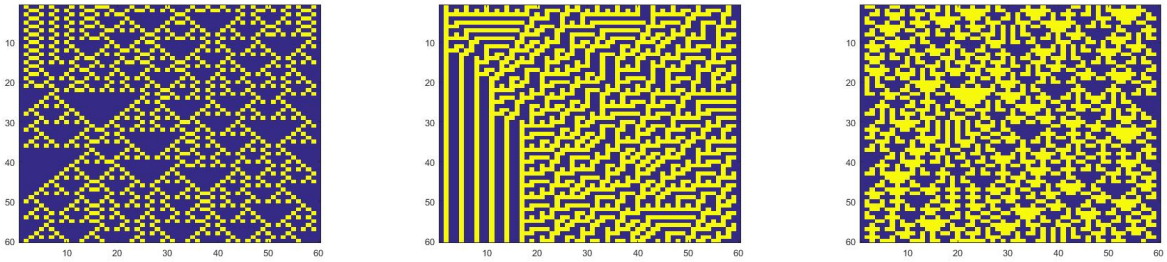
Rule 37



Rule 56

Figure .3.11 : Class 2 of Cellular automata, after a time evolution and starting from a random initial configuration, the evolution leads to a set of separated simple stable or periodic structures.

3. Class 3: Systems whose evolution yields chaotic periodic patterns (chaotic behaviour associated to strange attractors).
   Among the rules which are in this class we have rule 18, rule 45 and rule 150.



Rule 18 **Rule** 45 **Rule** 150

Figure .3.12 : Class 3 of Cellular automata, after a time evolution and starting from a random initial configuration, the evolution leads to a chaotic pattern.

4. Class 4: Systems whose evolution leads to a complex behaviour (no apparent analogues in continuous dynamical systems). We can cite rule 110.

3.2.6 Properties of Cellular Automata

Self reproduction It is an important propriety of life, the goal of Von Neumann when he has designed the principle of CA, was designing a mechanism which made

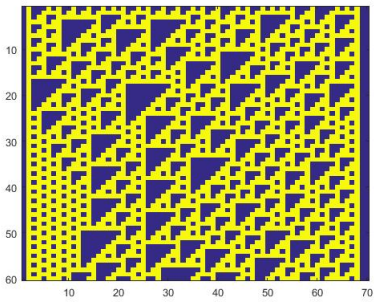


Figure .3.13 : Class 4 of Cellular automata, the evolution leads to a complex behaviour

a copy of life in a sens it can reproduce a copy of itself. This deals with the so called self similarity exhibited in some CA patterns.

Invertibility A CA is invertible if it possesses at least a configuration which has no ancestors. In terms of CA global dynamics F, the invertibility is equivalent to the injectivity and surjectivity of the map F.

Irreversibility We say that cellular automata is reversible if its transition rule allows to return to precedent states. In order to prove that CA has no inverse, one needs to find two different configurations which lead to the same configuration.

Undecidability: This propriety describe the impossibility to find a program which allows to test if a propriety is true. There is no algorithm which is able to predict how a CA is going to evolve. Also to determine if a CA possesses an inverse is undecidable: it will be never possible to write a program that takes as parameters any automata and be able to decide if this automata has an inverse or not. In the same way the future of CA is undecidable. We have no general method which allows to predict the evolution of any CA without performing simulations.

Simple Example of Binary Cellular Automata Rules

Example 3.2.3 *In this paragraph, we will introduce one of the simple binary cellular automata which called the parity rule, and either well known as the XOR (exclusive OR) rule. It has been introduced by E. Fredkin in 1970 [57]. We consider a CA* $\mathcal{A} = (\mathcal{L}_c, \mathcal{S}, \mathcal{N}, f)$ *where:*

- $\mathcal{L}_c$ *two dimensional cellular space.*
- $S = \{0, 1\}$ *a set of states.*
- $\mathcal{N}$ *a Von Neumann neighbourhood.*
- f *is transition function, which is given:*

$$s_{t+1}(c_{i,j}) = s_t(c_{i+1,j}) \oplus s_t(c_{i-1,j}) \oplus s_t(c_{i,j+1}) \oplus s_t(c_{i,j-1})$$

Where $\oplus$ is the XOR operator or in other words sum modulo 2.

We observe that this CA has the self replicate feature. In other words, any arbitrary pattern in the initial configuration is capable to replicate itself, Figure (.3.14) shows this feature.

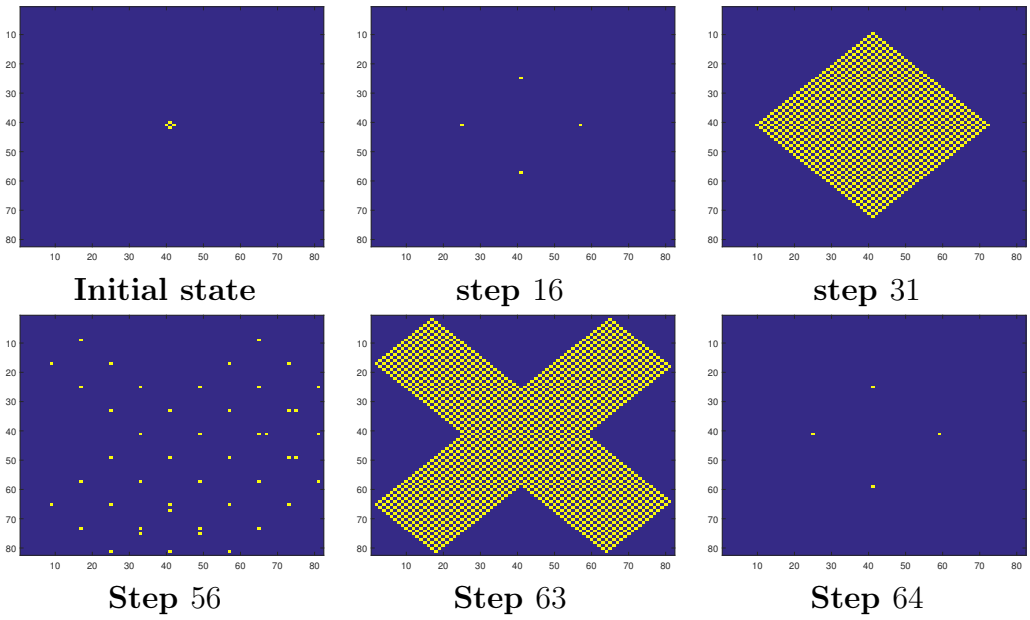


Figure .3.14 : Parity rule of size 80×80.

3.2.7 Linear Cellular Automata

Definition 13 *[60] A global dynamics F is linear if for every pair of configurations $s_1, s_2 \in S^{\mathcal{L}_c}$*

$$\forall \mu \in S, F(\mu s_1 + s_2) = \mu F(s_1) + F(s_2)$$

For the Boolean linear CA, the previous definition holds by using the XOR $\oplus$ instead of using plus $(+)$ thus we have the following definition:

Definition 14 *[74] If in a CA, the neighbourhood dependence is on EX-OR or EX-NOR only, then the CA is called an additive CA. Specifically, a linear CA employs XOR rules only.*

Example 3.2.4 *Consider a one-dimensional linear CA. The state of a cell at time* $t+1$ *depends on its neighbouring cells states at time* t *which can be given by:*

$$s_{t+1}(c_i) = (s_t(c_{i-1}) + s_t(c_i) + s_t(c_{i+1})) \quad mod \quad 2$$

or

$$s_{t+1}(c_i) = s_t(c_{i-1}) \oplus s_t(c_i) \oplus s_t(c_{i+1})$$

where $\oplus$ *denotes the XOR operator. This example is related to the Wolfram Rule* 150.

Remarque 3 *If the transition function of a CA rule is expressed by the XNOR operator only it is said to be the complement of a linear rule.*

Example 3.2.5 *Consider the Wolfram Rule* 105 *which is the complement of the rule* 150 *(their sum is* $255 = 2^8 - 1$*). The state of a cell at time* $t+1$ *depends on its neighbouring cells states at time* t *which can be given by:*

$$s_{t+1}(c_i) = \overline{s_t(c_{i-1}) \oplus s_t(c_i) \oplus s_t(c_{i+1})}$$

3.2.8 Applications of Cellular Automata

Cellular automata can be viewed as simple computational programs but have a remarkable ability to create complex behaviour. They have many applications in different fields : Cryptography and Random Number Generation, Implementing Parallel Computers, [28, 55, 60]. Since the 80s, the use of CA approach for modelling and simulation of physical or biological complex systems becomes more and more popular. Several examples have been successfully accomplished in this way: forest fire propagation, urban development, immunology and biological ageing, turbulence in fluid flow, Ising model dynamics, traffic models, etc. [50, 61, 62, 65]. The "Game of

Life" constructed by Conway [29] constitutes a famous example of cellular automata which may be easily simulated on a home computer. We shall present in this section two examples from the above mentioned applications of CA.

Traffic model

In our daily life the role of cars becomes necessary for travelling. In order to simplify the circulation, one needs to improve the traffic. Cellular automata can be useful for modelling traffic which can be described by a line using CA see [31, 84] where:

1. One dimensional array represent the road.

2. The value 1 means that the car occupies the cell.

3. The value 0 means that the cell is free.

The state of each cell depends on the state of the cell in front and behind the cell which occupies the car. The cars are moving to the right as in Figure (.3.15). If the car has a free cell in front of it, it will move to the right and in case where the cell in front of it occupies a car then it stays in its position. In notation introduced by Wolfram, this rule is written as 184, Figure (.3.16) [84]. This kind of dynamics already captures one essential characteristic of traffic flow, namely the traffic congestion. Maximum traffic flow is reached when only every second lattice cell is occupied. When traffic density becomes larger, the stop-and-go behaviour can be observed [25].

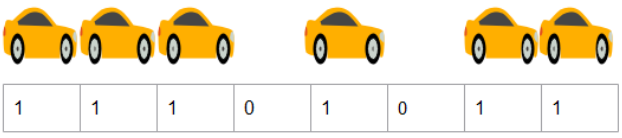


Figure .3.15 : traffic cars

More interesting behaviours can be observed by simulating traffic model in two dimensions. However, we will consider in the following the very popular Game of life example due to its computational capabilities.

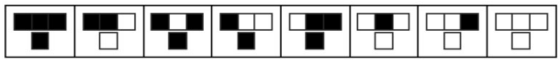


Figure .3.16 : Rule 184

Game of life The game of life is the best-known two-dimensional cellular automaton, invented by John H. Conway and popularized by Martin Gardner in the late 70s. It is not a real game as there are no players, and no winning or losing. Once the pieces are placed in the starting position, the rules extracted by the life principle determine everything that happen later. In most cases, it is impossible to look at a starting position (or pattern) and see what will happen in the future. The only way to find it out is to follow the rules of the game."[21]

Game of life CA is represented in a two-dimensional lattice with periodic boundary conditions. Each cell can take one of the states: dead if its value is 0 or alive if its value is 1, and change in each step of time its state depending on the states of its eight neighbours (Moore neighbourhood). The evolution of the game of life is given as follows:

1. A living site value with fewer than two live neighbours dies.

2. A living site with two or three living neighbours stays alive.

3. A living site with more than three living neighbours dies.

4. A dead cell with exactly three living neighbours becomes alive.

Game of life is an irreversible CA. In other words, knowing the actual configuration, it is impossible to determine the precedent configuration; It exists some configurations which have more than one antecedent. Game of life is also undecidable which means that there is no algorithm which can predict the evolution of CA starting from a certain configuration.

Life would definitely be considered in the class 4 as stated by Wolfram's classification. In this simple setting, periodic structures, stable blocks and complex moving patterns come into existence, even starting from a very simple initial configuration.

What we can learn from this dissertation is that CA models are able to describe the complexity of real world spatio-temporal systems despite their apparent simplicity.

However our main interest is to study CA in relation with the field of systems theory. And since the considered systems are distributed parameter systems, we needed to define CA in the context of DPS.

3.3 Cellular Automata as Distributed Parameter Systems

CA in its classical form is a closed system. In order to define them as DPS, inputs and outputs have been introduced in [46, 48]. Let us start by the classical autonomous CA.

3.3.1 Autonomous Cellular Automata

Definition 15 *[48] The local evolution of an autonomous cellular automaton is defined by the couple:*

$$A_c = (f, \mathcal{N}) \tag{3.7}$$

where f defines the local dynamics and $\mathcal{N}$ describes the local interactions between the cells of $\mathcal{L}_c$.
The state set is the set of all the configurations and will be denoted by X with

$$X = \mathcal{S}^{\mathcal{L}_c} \tag{3.8}$$

This set plays a similar role to what the state space is for DPS.

Definition 16 *[48] Given an initial configuration s_0, an autonomous CA can be defined in terms of the global dynamics by the state equation*

$$\begin{cases} s_{t+1} = Fs_t \\ s_0 \in X \end{cases} \tag{3.9}$$

and then completely determined by the couple (s_0, F).

This presentation is similar to that of discrete DPS where the state equation is often stated by

$$\begin{cases} z_{k+1} = Az_k \\ z_0 \in Z \end{cases} \tag{3.10}$$

and A *is the dynamics operator defined on the state space.*

It is easy to see that the solution of system 3.9 given by

$$s_t = F^t(s_0)$$

has the same form than the solution of DPS in autonomous case 3.10 given by

$$z_t = \phi_t(z_0)$$

where $(\phi_t)_{t\geq 0}$ *is the semi group generated by the operator* A.

Table 3.1: Analogies between DPS and CA [6]

Analogies between DPS and CA	
Distributed Parameter Systems	Cellular Automata
Dynamics of the system A generates a semi group $\phi_A(t)$	Dynamics of the system F generates a family F^t
State space $Z = L^2(\Omega)$	State space $X = \mathcal{S}^{\mathcal{L}_c}$
State Equation $\begin{cases} z_{k+1} = Az_k \\ z_0 \in Z \end{cases}$	State Equation $\begin{cases} s_{t+1} = Fs_t \\ s_0 \in X \end{cases}$
Solution of the system $z_t = \phi_t(z_0)$	Solution of the system $s_t = F^t(s_0)$

Let us now define the control in CA.

3.3.2 Control in Cellular Automata

Consider the following assumptions:

- $\mathcal{L}_c$ is a cellular space,
- $I = \{0, 1, \ldots, T\}$ is a discrete time horizon,
- $\mathcal{L}_{c_p}$, is a subdomain of $\mathcal{L}_c$ formed by p cells.

Let us consider the control space given as follows:

$$\mathcal{U} = C(I \times \mathcal{L}_{c_p}, \mathbb{R})$$

It consists of all the bounded controls defined as:

$$\begin{aligned} u : I \times \mathcal{L}_{c_p} &\rightarrow \mathbb{R} \\ (t, c_i) &\rightarrow u_t(c_i) \end{aligned}$$

Thus, the control operator G is given by:

$$\begin{aligned} G : \mathcal{U} &\rightarrow \mathcal{S}^{\mathcal{L}_c} \\ u &\rightarrow Gu \end{aligned}$$

This operator gives the way the control is applied to the CA on the p cells in $\mathcal{L}_{c_p}$.

The maps G associates to every control variable u a local control function $g : \mathbb{R} \rightarrow \mathcal{S}$ such that:

$$Gu_t(c) = \begin{cases} g(u_t(c)) & \text{for} \quad c \in \mathcal{L}_{c_p} \\ 0 & \text{elsewhere} \end{cases} \tag{3.11}$$

Introducing the characteristic function

$$\chi_{\mathcal{L}_{c_p}} = \begin{cases} 1 & for \quad c \in \mathcal{L}_{c_p} \\ 0 & \text{elsewhere} \end{cases} \tag{3.12}$$

one can write,

$$Gu_t = g(u_t(.))\chi\mathcal{L}_{c_p}$$

and so the obtained CA can be considered as a controlled system.

Definition 17 *[48] A controlled CA can be locally defined by the triple*

$$A_c = (f, g, \mathcal{N}) \tag{3.13}$$

where f and g are respectively the local transition and control functions and $\mathcal{N}$ is the neighbourhood.

The global evolution of a controlled CA denoted by A_c is described in the linear case by the following state equation:

$$\begin{cases} s_{t+1} = F(s_t + Gu_t) \\ s_0 \in X \end{cases} \tag{3.14}$$

where F is the global dynamics of the autonomous CA and G is the global control function.

Table 3.2: Introduction to control and observation [6]

Introduction to control and observation	
Distributed Parameter Systems	Cellular Automata
Dynamics of the system A	Dynamics of the system F
Control space $\mathcal{U} = \mathbb{R}^{n_\alpha}$	Control space $\mathcal{U} = l^2(\mathcal{T}^{n_\alpha}, \mathbb{R})$
Observation space $\mathcal{O} = \mathbb{R}^{n_c}$	Observation space $\mathcal{O} = l^2(\mathcal{T}^{n_c}, \mathbb{R})$
Control operator B	Control operator G
Observation operator B	Observation operator H
State equation $\begin{cases} \dot{z} = Az + Bu \\ z_0 \in Z \end{cases}$	State equation $\begin{cases} s_{t+1} = F(s_t + Gu_t) \\ s_0 \in X \end{cases}$
augmented by the output y=Cz	augmented by the output $\theta_t = H(s_t)$

3.4 Conclusion

This chapter launched the basic notions of CA and some notions of control theory which will be used in the following chapters. In addition to this, the chapter discusses CA in the context of system theory. We have introduced CA as distributed parameter systems with inputs and outputs, then we will deal with the controllability in CA models. We have shown through some examples that many spatio-temporels phenomena can be expressed by CA.

In the following chapter, we will study one of the most prominent problems in control theory that is of controllability of CA by using a new approach based on Markov chains.

Markov Chains approach for regional controllability of Cellular Automata

CHAPTER 4

MARKOV CHAINS APPROACH FOR REGIONAL CONTROLLABILITY OF CELLULAR AUTOMATA

Résumé:

Ce chapitre [1], est consacré à l'étude du problème de la contrôlabilité régionale des Automates Cellulaires en agissant sur la frontière de la région contrôlée. Cette étude s'efforce de mettre en lumière une nouvelle approche, celle des chaînes de Markov en établissant une matrice similaire à la matrice de transition d'une chaîne de Markov. Le but de la contrôlabilité régionale est d'amener l'état du système à un état désiré dans une région contrôlée à un temps donné T. Premièrement, nous avons démontré la contrôlabilité régionale des automates cellulaires en utilisant la définition d'une chaîne régulière. Cette preuve assure qu'à un certain temps donné T tous les états sont atteignables dans une région. En outre, cette preuve donne une information sur le temps nécessaire pour assurer que le système est régionalement contrôlable. Deuxièmement, nous avons démontré la contrôlabilité régionale en utilisant la définition

[1] Cette partie a fait l'objet des travaux suivants:
S. Dridi, S. El Yacoubi, F. Bagnoli (2019)
Boundary regional controllability of linear boolean cellular automata using Markov chains. Advances in Intelligent Systems and Computing.
S. Dridi, S. El Yacoubi, F. Bagnoli (2019)
Markov chains for regional controllability of cellular automata. Journal of Cellular Automata.

d'une chaîne ergodique. À la fin, nous avons démontré que, pour un temps donné T si la matrice de transition construite contient un état absorbant, alors le système est non régionalement contrôlable.

4.1 Introduction

Markov chains were first introduced in 1906 by Andrey Markov. They are considered as useful mathematical models which represent a class of stochastic processes of great interest which are used in a wide spectrum of practical applications. Markov chains use concepts from probability to describe how a system changes from one state to another one. A Markov chain process satisfies the Markov property, which means that the past and the future are independent when the present is known. In other words, if one knows the current state of the Markov process, then no supplementary information of its past states is required to predict its future. In this chapter, we will not delve to deep into mathematical aspects for Markov chains. We will focus on delivering a more general understanding which serve to solve the problem of regional controllability of deterministic cellular automata by acting on the boundaries of the controlled region.

The evolution of a controlled CA can be seen as a walk of Markov chain where the states are the possible configurations which can be represented in the region ω. A couple of configurations (s, s') (CA configurations restricted to ω) are related to each other if it exists a boundary control (ℓ, r) such that $p_{s,s'} = 1$ where $p_{s,s'}$ denotes the probability of jumping from a state configuration s to a state configuration s' in one step. Sometimes it is impossible to reach a state configuration in one step, but in q steps .i.e s' is reachable from s in q steps if $p^q_{s,s'} > 0$.

In this chapter, we prove the regional controllability by establishing a new relation between deterministic cellular automata and Markov chains by checking the existence of a power P^q which has all positive entries (i.e. strictly greater than zero) which allows us to give a sufficient and necessary condition. Another necessary and sufficient condition is obtained by using the definition of ergodic Markov Chain. The results have been extended to the two dimensional CAs. We start by giving a simple example of Markov chains.

Example 4.1.1 *Bilo is a child lost in a labyrinth, where reigns a total dakrness. Starting from the cave A, he chooses equiprobably one of the galleries from this cave and continues in this way until he reaches the cave C where he finds his parents or the cave E where he finds the exit.*

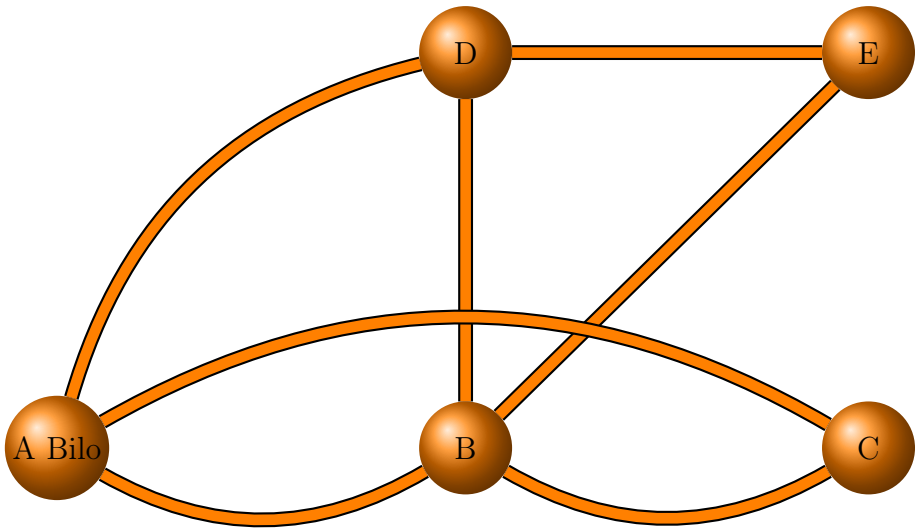


Figure .4.1 : Example of a Markov Chain

In mathematical terms, the definition of a Markov chain can be expressed as follows:

Definition 18 *A Markov Chain (MC) is a sequence of random variables* $X = \{X_0, X_1, X_2, \dots\}$ *with the following proprieties. For each* $k \in \{0, 1, 2, \dots\}, X_k$, *is defined on the sample space and* Ω *take values in a finite set* K. *Thus,* $X_k : \Omega \to M$. *Also for* $k \in \{0, 1, 2, \dots\}$, *and* $\{i, j, i_{k-1}, i_{k-2}, \dots, i_0\} \subseteq M$:

$$P\{X_{k+1} = j | X_k = i, X_{k-1} = i_{k-1}, X_{k-2} = i_{k-2}, \dots, X_0 = i_0\}$$

and the transition probabilities

$$P\{X_{k+1} = j | X_k = i\} = p_{i,j}$$

$p_{i,j}$ *are independent of* k.

For simplicity, we will give a simple definition in less technical terms.

We describe a Markov Chain with a finite number of states as follows: We have a system which changes from a state to another one. In other words, a system which starts from a state and moves successively from a state to another. Each move is called a step. A chain starts in a state s_i then it moves to a state s_j at the next step with probability denoted by $p_{i,j}$. We denote by the probability $p_{i,i}$ the case where the process remains in the same state in one step. The probability of being in a particular state at step t+1 depends only on the state occupied at step t. To describe a Markov Chain one needs to define the so-called transition matrix which is a prominent tool for analysing a Markov Chain.

4.1.1 Transition Matrix

Definition 19 *We often list the transition probabilities in a matrix. The matrix is called the state transition matrix or transition probability matrix and is usually shown by P. Assuming the states are* $1, 2, \ldots, M$, *then the state transition matrix is given by:*

$$P = p_{ij} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,M} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,M} \\ p_{(M-1),1} & p_{(M-1),2} & \cdots & p_{(M-1),M} \\ p_{M,1} & p_{M,2} & \cdots & p_{M,M} \end{pmatrix} \tag{4.1}$$

Note that P *is a non negative matrix in which each row sums to 1.*

We shall now give an example of a Markov chain with its transition matrix:

Example 4.1.2 *[1]*

According to Kemeny, Snell, and Thompson, the Land of Oz is blessed by many things, but not by good weather. They never have two nice days in a row. If they

have a nice day, they are just as likely to have snow as rain the next day. If they have snow or rain, they have an even chance of having the same the next day. If there is change from snow or rain, only half of the time is this a change to a nice day. With this information we form a Markov chain as follows. We take as states the kinds of weather R, N, and SN. From the above information we determine the transition probabilities. These are mostly represented in a square array as follows:

$$P = \begin{array}{c} \\ R \\ N \\ SN \end{array} \begin{array}{c} \begin{array}{ccc} R & N & SN \end{array} \\ \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{pmatrix} \end{array}$$

4.2 Evolution as walk on a graph or as a deterministic limit of a Markov chain

Markov chain is a dynamic process describing a system whose states change over time. The changes of states can be described by a transition matrix as follows:

$$P = p_{ij} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,M} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,M} \\ p_{(M-1),1} & p_{(M-1),2} & \cdots & p_{(M-1),M} \\ p_{M,1} & p_{M,2} & \cdots & p_{M,M} \end{pmatrix} \tag{4.2}$$

This probability transition matrix is an $M \times M$ matrix whose elements p_{ij} are all positive and whose rows sum to 1. Each element of this matrix p_{ij} is a conditional probability which represents the probability of change from a state into another one. For instance, given that the chain is in a state i at time t, the chain changes its state and jumps to the state j at time $t + 1$.

The evolution of CA is given as follows:

$$\boldsymbol{s}' = F(\boldsymbol{s})$$

It can be seen as a path connecting configurations, i.e. One can introduce a matrix $A_d(\boldsymbol{s}'|\boldsymbol{s})$ such that

$$A_d(\boldsymbol{s}'|\boldsymbol{s}) = p_{\boldsymbol{s},F(\boldsymbol{s})}$$

This matrix can be seen either as an adjacency matrix connecting configurations which can be represented by a graph see Figure (.4.2), or as a deterministic limit of a Markov chain.

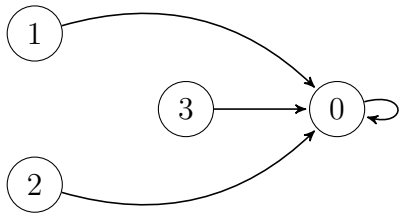


Figure .4.2 : Transition graph for a CA of size 2 with a periodic boundary condition rule 90 such that $F(0) = F((0000)_2) = (00)_2 = 0$, $F(2) = F((1010)_2) = (00)_2 = 0$, $F(1) = F((0101)_2) = (00)_2 = 0$, $F(3) = F((1111)_2) = (00)_2 = 0$.

The Boolean conversion of vertex 3 is 11, and for vertex 2 is 01.

Remark 1 *Note that we have taken the binary representation for the controlled region in a reverse order (the least significant bit is the first one). For instance, for a controlled region of size 3, we note:* $(100)_2 = 1$, $(110)_2 = 3$ *and* $(001)_2 = 4$

4.3 Regional controllability of one-dimensional linear CA:

4.3.1 Control problem:

We shall deal with the problem of regional controllability via boundary actions on the target region. The objective is to lead a dynamical system from any initial state to any desired state in a finite time, see Figure (.4.3). This problem can be defined in this way: Let us consider a Boolean one-dimensional CA and a target region $\omega = \{c_1, \ldots, c_n\}$. One asks if it is possible to drive the initial configuration $s_0(\omega) = \{s_0(c_1), \ldots, s_0(c_n)\}$ to a given desired configuration at time T by actions performed on the boundary $\{c_0, c_{n+1}\}$ of the region ω in the time interval $[0, T-1]$ such as:

$$s_T(c_i) = s_d(c_i) \quad \forall i = 1, \ldots, n$$

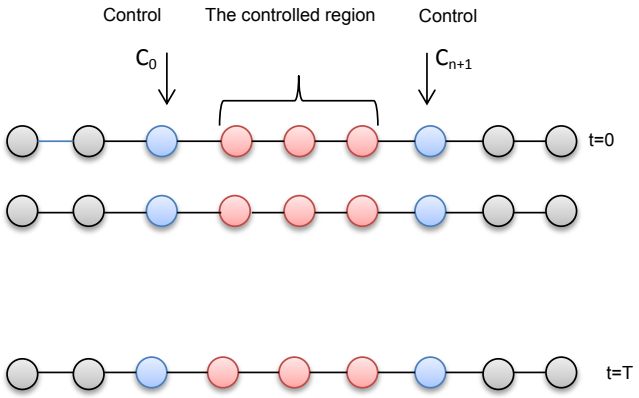


Figure .4.3 : Control of one-dimensional CA.

As for continuous systems, control problems that deal with a global behaviour study generally need the use of derivatives. To approach this issue for cellular automata dynamics, one attempts to define a linear operator that acts as some sort of derivative that should be defined for Boolean functions. As a local CA rule is a function of several Boolean variables, a derivative as an instantaneous rate of change cannot work. A more appropriate notion is that of Boolean derivative that produces values 0 or 1 since a Boolean variable can only be varied by flipping its value. Before going on, let us give some definitions and related examples:

Definition 20 *[89] The partial Boolean derivative of $f : S^r \longrightarrow S$ is defined as:*

$$\frac{\partial f_i}{\partial s_j} \overset{not}{=} \frac{\partial f}{\partial s_j} = f(s_i, \ldots, s_j \oplus 1, \ldots, s_{i+r}) \oplus f(s_i, \ldots, s_j, \ldots, s_{i+r})$$

Where $\oplus$ is the XOR Boolean operation and $s_j = s(c_j)$, $j = i, \cdots, i+r$ represent the values associated to the neighbourhood cells states.

For instance, the Boolean derivative with respect to s_1, for a local rule defined by $f(s_1, s_2) = s_1 \oplus s_2$ is:

$$\frac{\partial f}{\partial s_1} = s_1 \oplus s_2 \oplus (s_1 \oplus 1) \oplus s_2 = 1$$

Analogously, for $f(s_1, s_2) = s_1 s_2$, the Boolean derivative with respect to s_1 is:

$$\frac{\partial f}{\partial s_1} = s_1 s_2 \oplus (s_1 \oplus 1) s_2 = s_1 s_2 \oplus s_1 s_2 \oplus s_2 = s_2$$

Definition 21 (Boolean Derivative) *[89] The Boolean derivative of F is the Jacobian matrix defined as:*

$$J = \frac{\partial f_i}{\partial s_j}|_{i,j=1,\ldots,N}$$

Where N is the lattice number of cells.

Example 4.3.1 *Let $S = \{0, 1\}$ and $\mathcal{N}(c) = (c_{i-1}, c_i, c_{i+1})$. Let us consider the rule* 150 *defined by:*

$$f_i \overset{not}{=} s_i^{t+1} = f(s_{i-1}^t, s_i^t, s_{i+1}^t) = s_{i-1}^t \oplus s_i^t \oplus s_{i+1}^t$$

The partial derivative of this rule with respect to s_{i-1} is

$$\frac{\partial f_i}{\partial s_{i-1}} = \frac{\partial s_i^{t+1}}{\partial s_{i-1}} = (s_{i-1} \oplus 1 \oplus s_i \oplus s_{i+1}) \oplus (s_{i-1} \oplus s_i \oplus s_{i+1})$$

and then:

$$\frac{\partial s_i^{t+1}}{\partial s_{i-1}} = 1$$

This Boolean derivative measures whether s^{t+1} changes when changing s_{i-1}.

For a CA with zero boundary conditions, the Jacobian matrix takes the form:

$$J = \begin{pmatrix} 1 & 1 & \ldots & \ldots & \ldots & 0 & 0 & 0 \\ 1 & 1 & 1 & \ldots & \ldots & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & 1 & 1 & 1 & .. & \ldots & \\ \ldots & \ldots & \ldots & 1 & 1 & 1 & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & \ldots & \ldots & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{4.3}$$

According to the Jacobian, the evolution of linear CA will be expressed in the form:

$$s^{t+1} = Js^t \tag{4.4}$$

where s^t is the CA global state or configuration at time t.

The controllability was identified by R. Kalman for finite dimensional linear systems and characterised by the Kalman condition [69]. In order to prove the controllability of linear CA, we propose an alternative method by exploring the approach based on Markov Chains that could be applicable also for nonlinear rules.

The Kalman rank condition states that if the controllability matrix defined in [69] is of full rank then the system is controllable.

This criterion has already been extended to regional controllability in [98]. The Markov Chains approach allows not only to prove the regional controllability of linear CA systems but also to find the required time T for which the system is regionally controllable by using the definition of regular Markov chain. One can find also the required control on the boundaries $\{c_0, c_{n+1}\}$ in the time interval $[0, T-1]$ to reach the desired configuration s_d on ω given by $\{s_d(c_1), s_d(c_2), \ldots, s_d(c_n)\}$. Starting from an initial configuration $\{s_0(c_1), s_0(c_2), \ldots, s_0(c_n)\}$ the CA configuration will evolve to a final state at time T according to the application of a transition matrix defined in the following paragraph.

The evolution of linear cellular automata can be expressed as

$$s^{t+1} = Js^t,$$

where s^{t+1} is the configuration at time $t+1$ and s^t the configuration at time t.
A controlled cellular automata can be defined by the following state equation:

$$s^{t+1} = Js^t + BU^t,$$

where $s^{t+1} = \{s^{t+1}(c_1), s^{t+1}(c_2), \ldots, s^{t+1}(c_n)\}$, $s^t = \{s^t(c_1), s^t(c_2), \ldots, s^t(c_n)\}$

B is an $(n \times 2)$ matrix and U^t is the control vector (2×1) :

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \qquad U^t = \begin{pmatrix} u^t(c_0) \\ u^t(c_{n+1}) \end{pmatrix} \tag{4.5}$$

where $u^t(c_0)$ is the control applied on the cell c_0 and $u^t(c_{n+1})$ is the control applied on the cell c_{n+1}.

Definition 22 *[49] The CA is said to be regional controllable for $s_d \in S^\omega$ if there exists a control sequence $(U^0, \ldots, U^{T-1})$ where $U^i = (u^i(c_0), u^i(c_{n+1})), \quad i = 0, \ldots, T-1$ such as*

$$s_T = s_d \quad on \quad \omega,$$

where s_T is the final configuration at time T and s_d is the desired configuration.

Notation 1 *Let us introduce the following notations*

- $\overline{\omega} = \{c_0, c_1, \ldots, c_n, c_{n+1}\} = \omega \cup \{c_0, c_{n+1}\}$ *where $\{c_0, c_{n+1}\}$ are the boundary cells of ω where we apply control,*
- *We denote by $|\omega|$ the number of cells in the controlled region,*
- $a = u^0(c_0)$ *and*
- $b = u^0(c_{n+1})$.

Let us now consider the problem of regional controllability of linear CA. The evolution of CA by applying the control on the boundaries can be seen as a Markov Chain. One needs to define a square matrix $2^{|\omega|} \times 2^{|\omega|} \quad \mathcal{C}$ such that, each element $\mathcal{C}_{(s),(s')}$ can be represented as the conditional probability $p_{(s'|\mathbf{a}s\mathbf{b})}$.
For each pair $(\mathbf{a}s\mathbf{b}, s')$, s' is reachable from s by applying the control a, b on the cells $\{c_0, c_{n+1}\}$ and we denote it as $F(\mathbf{a}s\mathbf{b}) = s'$. In terms of graph theory, if it exists an

arc starting from s to s', in other words it is possible to go from the state s to s' by applying the control $\{\mathbf{a}, \mathbf{b}\}$ but sometimes not necessarily in one move, i.e it exists a path between s and s' for that the definition of regular Markov chain is welcomed. A Markov Chain is called regular if some power of the transition matrix has only positive elements.

Let us consider the one-dimensional CA rule 150; we focus on reaching a given configuration in a region $\omega = \{c_1, \ldots, c_n\}$.

The problem of proving if a CA is regional controllable by acting on its boundaries reduces to checking the existence of $\mathcal{C}^T > 0$.

For instance for a CA whose controlled region is of size $|\omega| = 3$, we define a square matrix $2^3 \times 2^3$ $\mathcal{C}$, 2^3 referring to all the possible configurations can be represented in the controlled region $\omega = \{c_1, c_2, c_3\}$. Assume that the initial state in the controlled region $(000)_2 = 0$ ($(000)_2$ is the binary representation of 0), and then use all the possible controls on the boundaries to calculate s' (by using the jacobian matrix) we get:

$F_{S^{\overline{\omega}}}(00000) = (000)_2 = 0,$

$F_{S^{\overline{\omega}}}(00001) = (001)_2 = 4,\ F_{S^{\overline{\omega}}}(10000) = (100)_2 = 1,\ F_{S^{\overline{\omega}}}(10001) = (101)_2 = 5,$

which means that the probability of reaching one of the configurations $0, 1, 4, 5$ starting from 0 is one, i.e, $p_{0 \rightsquigarrow 0,1,4,5} = 1$.

The adjacency matrix $\mathcal{C}$ can be reduced to a Markov transition matrix after normalisation. We already know that we can associate a graph to the transition matrix of a Markov Chain, where the vertices of the graph are the different states. If $p_{i,j} > 0$ ($p_{i,j}$ is the transition probability from a state i to a state j, we have an arrow from i to j. For the transition matrix $\mathcal{C}$ of deterministic cellular automata, the vertices are the configurations; if $p_{i,j} > 0$, so the configuration j is reachable from i.

Algorithm: *Transition Matrix $\mathcal{C}$ one-dimensional linear CA*

loltab $\leftarrow$ zeros$(4, (n+2))$; loltab(4,1) $\leftarrow$ 1; loltab is the matrix of all possibilities of applying the control on the boundaries.

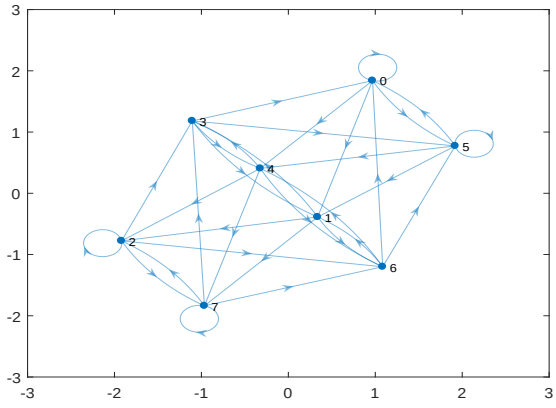
loltab(2,(n+2))$\leftarrow$ 1;

loltab(3,1) $\leftarrow$ 1; loltab(4,(n+2))$\leftarrow$ 1;

for $i \leftarrow 1$ **to** $2^{|\omega|}$ **do**
convert (i-1) to binary and save the values in vector initialconfiguration of size n.
for $l \leftarrow 1$ **to** 4 **do**
d $\leftarrow$ 1;
for $m \leftarrow 2$ **to** $(n+1)$ **do**
loltab(l,m)$\leftarrow$ initialconfiguration(d);
d $\leftarrow$d+1;
end for
end for
do the evolution of each line of the matrix loltab (by using the formula $X^{t+1} = JX^t$ and save it in the matrix binaryconf of size $4\times(n+2)$, we consider that each line of the matrix loltab is the configuration at time t by applying the control on the boundaries.
for $h \leftarrow 1$ **to** 4 **do**
did $\leftarrow$1;
for $r \leftarrow 2$ **to** $(n+1)$ **do**
table(h,did) $\leftarrow$mod(binaryconf(h,r),2);
did$\leftarrow$ did+1;
end for
end for
convert each line of the matrix table to decimal and save it in vector resultdecimal.
for $j \leftarrow 1$ **to** $2^{|\omega|}$ **do**
for $l \leftarrow 1$ **to** 4 **do**
if $(j-1) = resultdecimale[l]$ **then**
$\mathcal{C}$(i,j) $\leftarrow$ 1;
end if
end for
end for
end for

Example 4.3.2 ***Rule*** 150***, the controlled region is of size***=3

$$\mathcal{C}_{150} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$



Graph of the matrix $\mathcal{C}$.

We recall some definitions of Markov Chains:

Definition 23 *A state j is said to be reachable from a state i and we denote it as $i \rightsquigarrow j$, if it exists a strictly positive probability to reach a state j from a state i in a finite number of transitions.*

In terms of graph, if it exists a path between i and j.

Definition 24 *A Markov chain is said to be ergodic if $i \rightsquigarrow j$ $\forall i, j \in S$, where S is the set of states. A Markov chain is said to be regular if it exists a power P^q of P such as all the components of P^q are strictly positive, where P is the transition matrix of Markov chain.*

Remark 2 *The matrix P^q gives the probability that the Markov chain, starting in state M_i, will be in state M_j after q steps.*

Example 4.3.3 *In the regular chain defined by the dip graph in Figure (4.4), the transition matrix is given by:*

$$P = \begin{array}{c} \\ M_1 \\ M_2 \end{array} \begin{array}{c} \begin{array}{cc} M_1 & M_1 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1/2 & 1/2 \end{pmatrix} \end{array}$$

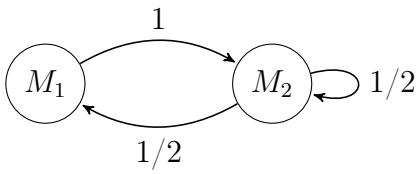


Figure .4.4 : A regular chain

since all pairs of vertices can be joined by a path of length 2, we find that:

$$P^2 = \begin{array}{c} \\ M_1 \\ M_2 \end{array} \begin{array}{c} \begin{array}{cc} M_1 & M_1 \end{array} \\ \begin{pmatrix} 0.5 & 0.5 \\ 0.25 & 0.75 \end{pmatrix} \end{array}$$

We notice that P^2 has all positive elements.

By using the last definition, we get the following theorem:

Theorem 4.3.4 *A linear Cellular Automaton is regional controllable iff it exists a power $\mathcal{C}^T$ such as all the components are strictly positive.*

Proof 4.3.5 *The matrix $\mathcal{C}$ is similar to the transition matrix of Markov Chains, $\mathcal{C}^T$ is the matrix at time T.*
For a pair of states (k_1, k_2), the state k_2 is reachable from k_1, if there exist an integer $T > 0$ such that $\mathcal{C}^T_{k_1 k_2} > 0$. We assume that $\mathcal{C}^T_{i,j} > 0 \quad$ for all $i, j \in \{0, \ldots, 2^{|\omega|} - 1\}$. Then all configurations in the controlled region are reachable at time T and the CA related to this transition matrix is regional controllable.

Let us assume that the CA is regional controllable.Then it exists $T > 0$ such as for each $i, j \in \{0, \ldots, 2^{|\omega|} - 1\}$ $i \rightsquigarrow j$ in time T which implies that for each $i, j \in \{0, \ldots, 2^{|\omega|} - 1\}$, $\quad \mathcal{C}^T_{i,j} > 0$. Thus the theorem holds.

Remark 3 *The matrix $\mathcal{C}$ is a square matrix $(|V| \times |V|)$. We denote by V the vertices, in term of graph theory we call it the adjacency matrix where $|V| = 2^{|\omega|}$ (all possible configurations in the controlled region) such that its element $C_{ij} = 1$ when there is*

an arrow from a vertex i to a vertex j and zero otherwise. $\mathcal{C}$ can be represented by a finite graph.

Example 4.3.6 *Consider a linear CA governed by Wolfram rule* 150 *[97], the controlled region is of size $|\omega| = 4$. It is clear from the graph correspond to the square $2^4 \times 2^4$ matrix $\mathcal{C}$ that some configurations are not reachable (there is no arrow from 9 to 12).*

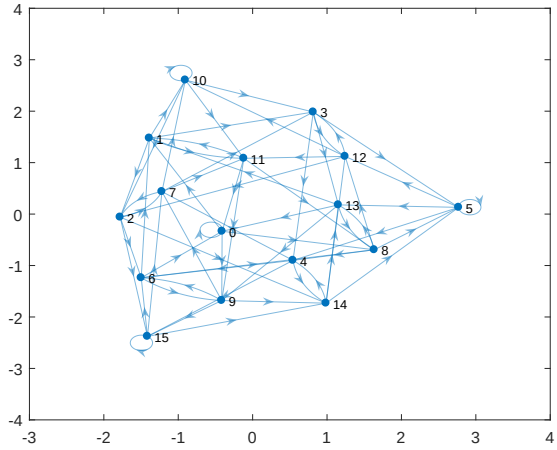


Figure .4.5 : Graph of the matrix $\mathcal{C}$ of rule 150 at time $t = 1$.

At time $t = 2$, the matrix $\mathcal{C}^2 > 0$ i.e, all configurations are reachable. The CA is regional controllable by acting on the boundaries of the target region at time $t = 2$.

We have chosen CA of small size to compare between the two graphs in instant $t = 1$ and $t = 2$. The theorem holds for CA's of larger sizes.

Remark 4 *One can also find the required controls on the boundaries $\{c_0, c_{n+1}\}$. For instance, starting from the configuration $(011)_2 = 6$ one can achieve the desired configuration $(001)_2 = 4$ at time $t = 2$. The sequence of required controls is $\{1, 1\}$ at time $T = 0$ and $\{1, 0\}$ at time $t = 1$.*

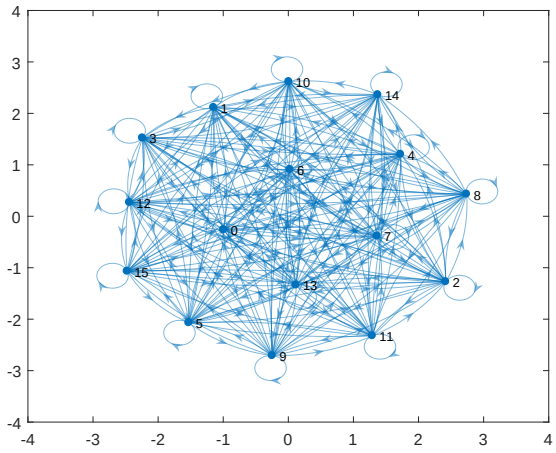


Figure .4.6 : Graph of the matrix $\mathcal{C}^2$ of rule 150 at time $t = 2$.

Remark 5 *The matrix $\mathcal{C}$ is an adjacency matrix in terms of graph theory. It does not satisfy the property of the transition matrix of Markov chain which states that the sum of each line of the transition matrix P equals one. To obtain the matrix satisfying this propriety, one can normalise the matrix $\mathcal{C}$ in the following way: For instance, for the CA rule 150, the controlled region is of size $|\omega| = 2$. The obtained matrix is given as follows:*

$$\mathcal{C}_{150} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

to which one can associate a graph.

The normalised matrix is given as follows:

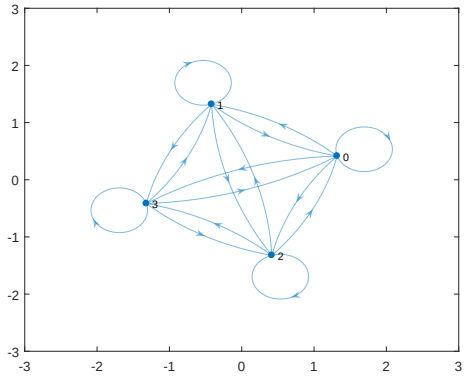


Figure .4.7 : The graph of the matrix $\mathcal{C}$ rule 150

$$\mathcal{C}_{150}^{norm} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} \end{array}$$

However and for more simplicity, we choose to work directly with the non normalised matrix $\mathcal{C}$.

Remark 6 *The required time T to reach the regional controllability of linear CA can also be obtained by investigating the biggest cost route between the vertices.*

To check the controllability in minimal time, we can use this definition.

Definition 25 *Markov chain is called an ergodic chain if it is possible to go from any state to any other one (not necessarily in one step).*

To complete this concern, we just give the following result without proof:

Theorem 4.3.7 *A linear CA is regionally controllable if it is ergodic.*

4.4 Regional Controllability of two-dimensional linear CA:

4.4.1 Control problem:

Let us consider a linear two-dimensional CA of size $\mu \times \eta$, the idea is to impose the control on the boundaries of a controlled region ω in order to force the appearance of a given configuration in the target region ω, see Figure (.4.8):

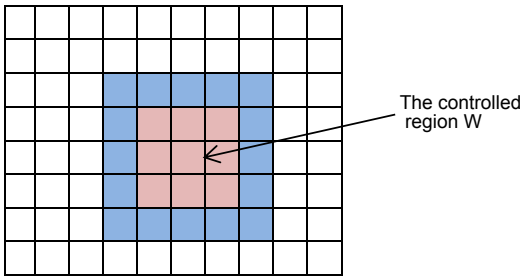


Figure .4.8 : Control of two-dimensional CA in the interval $[0, T]$.

Remark 7 *We can define asymmetric controls i.e, we keep a part of the boundaries fixed (for instance at 0) and act on a subset of the boundary of the controlled region (pink cells) in order to get the desired state see Figure (.4.9):*

Lemma 4.4.1 *[27] The equivalent one-dimensional map matrix for any rule R can be represented as a tri-diagonal binary matrix as follows:*

$$\mathcal{T}_R = \begin{pmatrix} D & U & ... & ... & ... & 0 & 0 & 0 \\ L & D & U & ... & ... & 0 & 0 & 0 \\ 0 & L & D & U & ... & 0 & 0 & 0 \\ ... & ... & L & D & U & .. & ... & \\ ... & ... & ... & L & D & U & ... & ... \\ ... & ... & ... & ... & L & D & U & ... \\ 0 & 0 & 0 & ... & ... & L & D & U \\ 0 & 0 & 0 & 0 & 0 & 0 & L & D \end{pmatrix}_{\mu\eta \times \mu\eta} \tag{4.6}$$

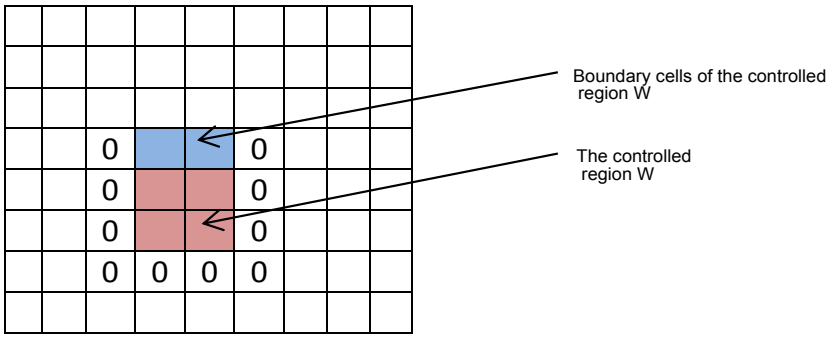


Figure .4.9 : Regional control of two dimensional CA with asymmetric controls.

where D, L and U are one of the following matrices of the order of $\mu \times \mu$: $[0], [I], [\mathcal{M}_1], [\mathcal{M}_2], [I + \mathcal{M}_1], [I + \mathcal{M}_2], [\mathcal{S}]$ *and* $[I + \mathcal{S}]$*, and $\mathcal{S}$ is the sum of $\mathcal{M}_1$ and $\mathcal{M}_2$, $\mathcal{M}_1$ and $\mathcal{M}_2$ are defined as follows:*

$$\mathcal{M}_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{4.7}$$

and

$$\mathcal{M}_2 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{4.8}$$

The rule matrix $\mathcal{T}_R$ has dimension $(\mu\eta \times \mu\eta)$. The state of the system s' obtained by multiplying $\mathcal{T}_R \times s$, such as s can be represented by a 1D column matrix of order $(\mu\eta \times 1)$ obtained from the binary 2D $(\mu \times \eta)$ matrix in a row major order as reported in [49].

The 2D configuration:

$$\begin{pmatrix} s(c_{11}) & s(c_{12}) & \dots & s(c_{1\eta}) \\ s(c_{21}) & s(c_{11}) & \dots & s(c_{2\eta}) \\ \vdots & \vdots & \dots & \vdots \\ s(c_{\mu 1}) & s(c_{\mu 2}) & \dots & s(c_{\mu\eta}) \end{pmatrix}$$

can be transformed to a 1D configuration consisting of a column vector
$\rightarrow (s(c_{11}), s(c_{12}), \dots, s(c_{\mu 1}), \dots, s(c_{\mu\eta}))^t$.

By using the transformation matrix $\mathcal{T}_R$, we can calculate the next state by operating it on the current CA states (the binary matrix of dimension $(\mu \times \eta)$) generates the next state $[s^{t+1}]_{\mu\times\eta}$, we consider in this study only null of fixed (0th state) valued boundary condition.

$$\begin{pmatrix} s^{t+1}(c_{11}) \\ \vdots \\ s^{t+1}(c_{1\eta}) \\ \vdots \\ s^{t+1}(c_{\mu 1}) \\ \vdots \\ s^{t+1}(c_{\mu\eta}) \end{pmatrix} = (\mathcal{T}_R)_{\mu\eta\times\mu\eta}. \begin{pmatrix} s^{t}(c_{11}) \\ \vdots \\ s^{t}(c_{1\eta}) \\ \vdots \\ s^{t}(c_{\mu 1}) \\ \vdots \\ s^{t}(c_{\mu\eta}) \end{pmatrix} \tag{4.9}$$

For a two-dimensional cellular automata, one wants to control the region ω by applying the control from one side of its boundaries, see Figure (.4.9).
The evolution of 2D cellular automata is given by:

$$s^{t+1} = \mathcal{T}_R s^t$$

Where s^{t+1} is the configuration at time $t+1$ and s^t the configuration at time t.

Firstly, for a two-dimensional CA, $|\omega| = m \times n$ we define the controlled region by a vector of size $1 \times nm$. For that purpose, one needs to calculate all the possibilities of applying the control from one boundary of the target region which is $2^{columns} = 2^n$. We define a square matrix $2^{n\times m} \times 2^{n\times m}$ $\mathcal{C}$.

Algorithm: *Transition Matrix $\mathcal{C}$ two-dimensional linear CA*

Define a matrix loltab of dimension $2^n \times ((m+1) \times n)$.
Enter all the possibilities of applying control and save them in the boundary of loltab (the boundary is constituted by n cells of each line of the matrix loltab).
for $i \leftarrow 1$ **to** $2^{|\omega|}$ **do**
  convert (i-1) to binary and save the values in vector tab of size $1 \times (n \times m)$.
  for $k \leftarrow 1$ **to** 2^n **do**
    r ←1;
    for $y \leftarrow (n+1)$ **to** $(n \times (m+1))$ **do**
      loltab(k,y) ←tab(1,r);
      r← r+1;
    end for
  end for
do the evolution of each line of the matrix loltab (by using the formula $X^{t+1} = \mathcal{T}_R X^t$) and save it in a matrix matc of size $2^n \times (n \times (m+1))$, we consider that each line of the matrix loltab is the configuration at time t by applying the control on the boundaries.
for $h \leftarrow 1$ **to** (2^n) **do**
  di←1;
  for $r \leftarrow (n+1)$ **to** $(n \times (m+1))$ **do**
    table(h,di) ← mod(matc(h,r),2);
    di←di+1;
  end for
end for
convert to decimal each line of the matrix table and save it in vector resultdecimal.
for $j \leftarrow 1$ **to** $2^{|\omega|}$ **do**
  for $l \leftarrow 1$ **to** 2^n **do**
    if $(j-1) = resultdecimale[l]$ **then**
      $\mathcal{C}$(i,j) ← 1;
    end if
  end for
end for

end for

Notation 2

- *Cont is the set of the cells where we apply the control (the pink cells) see Figure (4.4.9),*
- $\overline{\omega} = \omega \cup Cont$.

Example 4.4.2 *Let a two-dimensional cellular automaton, the controlled region is of size* $|\omega| = 2 \times 2$*, the local evolution of this rule is given as follows:*

$$s^{t+1}(c_{i,j}) = s^t(c_{i-1,j}) \oplus s^t(c_{i+1,j}) \oplus s^t(c_{i,j-1}),$$

and the transformation matrix is given as follows:

$$\mathcal{T} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \tag{4.10}$$

We impose asymmetric controls by setting all cells on the boundaries to 0 *except for the two cells (in blue color) as in Figure (4.4.9). The CA is regional controllable at time* $T = 2$.

For instance for the vertex 0*, its binary conversion is* $(0000)_2$*, by adding the possible controls we get:*

$$F_{S^{\overline{\omega}}}(000000) = (0000)_2 = 0,\ F_{S^{\overline{\omega}}}(010000) = (0100)_2 = 2,$$
$$F_{S^{\overline{\omega}}}(100000) = (1000)_2 = 1,\ F_{S^{\overline{\omega}}}(110000) = (1100)_2 = 3,$$

For the vertex 6*, its binary conversion is* $(0110)_2$*, by adding the possible controls we get:*

$$F_{S^{\overline{\omega}}}(000110) = (1000)_2 = 1,\ F_{S^{\overline{\omega}}}(010110) = (1100)_2 = 3,$$
$$F_{S^{\overline{\omega}}}(100110) = (0000)_2 = 0,\ F_{S^{\overline{\omega}}}(110110) = (0100)_2 = 2,$$

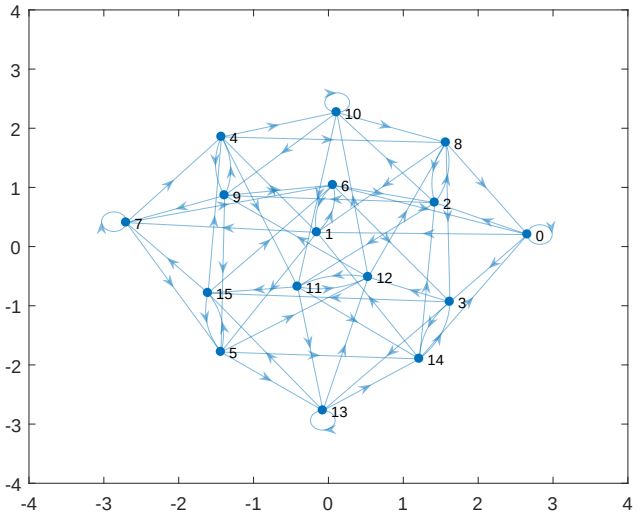


Figure .4.10 : The evolution of the controlled two dimensional CA at time $t = 1$ for a controlled region of size 2×2.

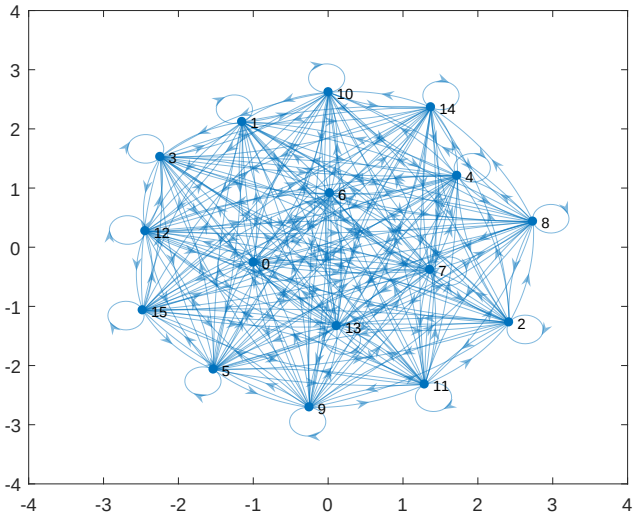


Figure .4.11 : The evolution of the controlled two dimensional CA at time $t = 2$ for a controlled region of size 2×2.

It is clear from the graph that all configurations are reachable in the controlled region.

The theorem also holds for non linear cellular automata.

Theorem 4.4.3 *A non linear Cellular Automata is regional controllable via boundary actions iff it exists a power $\mathcal{C}^T$ such as all the components are strictly positive.*

Example 4.4.4 *The local evolution rule is expressed either as a table, mapping the next states from all possible combinations of inputs. For instance, consider the transition function corresponding to rule 22 in Wolfram's notation (cf. [94]) where the neighbourhood consists of adjacent cells and the cell itself:*

$$\begin{array}{rccc} f_{22}: & 000 & \mapsto & 0 \\ & 001 & \mapsto & 1 \\ & 010 & \mapsto & 1 \\ & 011 & \mapsto & 0 \\ & 100 & \mapsto & 1 \\ & 101 & \mapsto & 0 \\ & 110 & \mapsto & 0 \\ & 111 & \mapsto & 0 \end{array}$$

Let us consider a non linear CA rule 22, $|\omega| = 4$ cells, i.e. the controlled region is of size 4. This CA is regional controllable by acting on the boundaries of the target region ω at time $t = 4$.

Theorem 4.4.5 *A deterministic CA is not regional controllable if the transition matrix related to the CA admits a state configuration i for which $\mathcal{C}_{ij} = 0\ \forall j = 0, \ldots, 2^{|\omega|} - 1$.*

Proof 4.4.6 *Let us consider the transition matrix $\mathcal{C}$ of size $2^{|\omega|} \times 2^{|\omega|}$, the controlled region of size $|\omega|$ and $z = 2^{|\omega|} - 1$*

$$\mathcal{C} = \begin{bmatrix} p_{0,0} & p_{0,1} & p_{0,2} & \cdots & p_{0,z} \\ p_{1,0} & p_{1,1} & p_{1,2} & \cdots & p_{1,z} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{z-1,0} & p_{z-1,1} & p_{z-1,2} & \cdots & p_{z-1,z} \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}$$

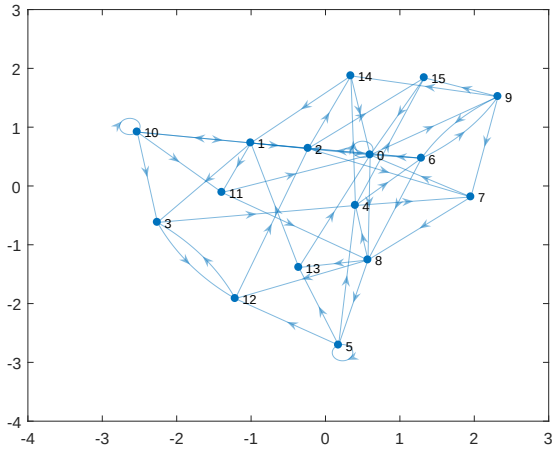


Figure .4.12 : Graph of the matrix $\mathcal{C}$ time $t = 1$ of the rule 22.

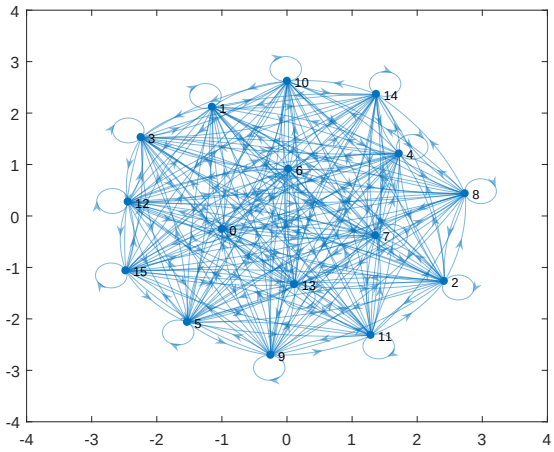


Figure .4.13 : Graph of the matrix $\mathcal{C}^4$ of the rule 22, at time $t = 4$.

where $p_{q,y} > 0 \quad$ *for* $q = 0, \ldots, z-1 \quad$ *and* $\quad y = 0, \ldots, z$, $p_{q,y} > 0$ *is the probability to reach the state* y *from the state* q. *Let us assume that the CA related to* $\mathcal{C}$ *is regional controllable at time* T, *for a given configuration* i. *Since we have* $p_{2^{|\omega|}-1,i} = 0 \quad for \quad i = 0, \ldots, 2^{|\omega|} - 1$, *there exists no power* C^T *such as the configuration* i *is reachable from the configuration* $2^{|\omega|} - 1$, *therefore The CA related to the transition matrix* $\mathcal{C}$ *is not regional controllable.*

We have another type of Markov Chains which called an Absorbing Markov Chain. This type is used to describe many phenomena in the real world. An absorbing Markov Chain is a Markov Chain in which some states cannot be left once achieved. These states are called absorbing states in which are reached from any state after some number of steps. For instance, to model a living organisms, the death is considered as an absorbing state. It is impossible to leave this state once the organism enters it. The associate transition matrix can be the defined matrix as follows:

$$P = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0.2 & 0.7 & 0.1 \\ 0 & 1 & 0 \\ 0.6 & 0.2 & 0.2 \end{pmatrix} \end{array}$$

This matrix shows the probability $p_{i,j}$ of moving from a state i to the state j. The probability of staying in state 2 is 1. Thus, once state 2 is entered it is not possible to leave it. We call the state 2 an absorbing state. The related diagram to the transition matrix shows that is impossible to leave the state 2.

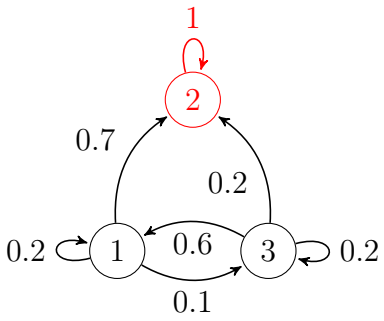


Figure .4.14 : Diagram related to the transition matrix

Definition 26 *[63] A state $i \in S$ is said to be absorbent if $p_{ii} = 1$ and then necessarily $p_{ij} = 0$ for each $j \neq i$.*

Theorem 4.4.7 *A Deterministic Cellular automata is not regional controllable, for*

a given time t, if the transition matrix $\mathcal{C}^t$ related to the CA admits an absorbent state.

Proof 4.4.8 *Let $\mathcal{C}^t$ the transition matrix of size $2^{|\omega|} \times 2^{|\omega|}$,* $z = 2^{|\omega|} - 1$

$$\mathcal{C}^t = \begin{bmatrix} p_{0,0} & p_{0,1} & p_{0,2} & \cdots & p_{0,z} \\ p_{1,0} & p_{1,1} & p_{1,2} & \cdots & p_{1,z} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{z-1,0} & p_{z-1,1} & p_{z-1,2} & \cdots & p_{z-1,z} \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

At time t, the state $2^{|\omega|} - 1$ is an absorbent state, it is not reachable at time t by any other configuration, so the CA is not regional controllable at time t.

4.5 Conclusions

This work launched the problem of regional controllability via boundary actions on the target region of CA. We have proposed a new approach to characterize the regional controllability of Boolean one-dimensional and two-dimensional deterministic CA, based on Markov chains.
We have shown that the evolution of a controlled CA can be seen as a walk of Markov Chain which allow to build a matrix similar to the transition matrix of Markov Chains. We have proved the regional controllability by using the definitions of ergodic and regular Markov chain. This approach allows not only to prove the regional controllability of CA systems but also to find the required time T for which the system is regionally controllable by using the definition of regular Markov chain. Furthermore, the use of this approach allows to find the required control to reach a given desired state configuration. Finally, we have equally used the definition of absorbent state to prove the non regional controllability of CA.

As Markov chains can be described using directed graphs where the nodes represent the different possible states and the edges represent the probability of the system moving from one state to the other in the next time instance. It seems that the graph theory is a good and appropriate tool to solve the problem of regional

controllability of CA. In the following chapter, we will study the same problem and we will give some necessary and sufficient conditions based on the notions of graph theory.

Graph theory approach for regional controllability of Cellular Automata

CHAPTER 5

GRAPH THEORY APPROACH FOR REGIONAL CONTROLLABILITY OF CELLULAR AUTOMATA

Résumé:

Dans ce chapitre [1], on a étudié le problème de la contrôlabilité régionale des Automates Cellulaires en agissant sur la frontière de la région contrôlée. La nouveauté la plus importante réside en l'approche originale est celle de la théorie des graphes. Rappelons qu'un système est dit contrôlable, si pour tout état désiré il existe un contrôle qui permet de transférer le système depuis un état initial vers un état désiré. Premièrement, on a obtenu la contrôlabilité régionale dont la preuve repose sur l'existence d'un Circuit Hamiltonien qui assure qu'à un certain temps donné T tous les états sont atteignables. D'autre part, on a démontré la contrôlabilité d'une autre façon en se basant sur l'existence d'une composante fortement connexe. Cette condition peut être considérée comme un critère de décidabilité. Des exemples illustratifs sont donnés dans le cas unidimensionnel et bidimensionnel et diverses situations ont été examinées. À la fin, on a démontré comment trouver le contrôle qui assure la contrôlabilité régionale en utilisant l'approche des préimages.

[1] Cette partie a fait l'objet de travail suivant:
S. Dridi, S. El Yacoubi, F. Bagnoli and A. Fontaine (2019)
A graph theory approach for regional controllability of Boolean cellular automata. International Journal of Parallel, Emergent and Distributed Systems, 1-15.

Begin at the beginning, the King said, gravely, and go on till you come to the end; then stop. — Lewis Carroll, Alice in Wonderland

5.1 Introduction

Graph theory is a branch of mathematics which deals with problems that have a fairly natural graph. It is a mathematical study of connections between nodes (any kind of entities) and edges which represent the relationships between nodes. Graph theory has been invented in 1736 by the great Leonhard Euler, who used it to solve the Königsberg Bridge Problem. The branch of Graph theory began to blossom in the twentieth century and after that the growth continued thus it became in nowadays a prominent field which can be applied in a wide variety of subjects as operational research, chemistry and biology [13, 22, 58]. Many real-world problems or situations can be conveniently be described by the mean of graphs [73]. The evolution in the case of Boolean deterministic CA can be represented by an oriented graph where the vertices correspond to the configurations obtained from the binary representation converted to decimal. There is an arc between two vertices v_1 and v_2 if the configuration corresponding to v_2 can be obtained from the configuration obtained from v_1 where the local transition function is applied. The evolution of a controlled Cellular automata in a region ω either can be described by an oriented graph, where the vertices represent the configurations in the controlled region and they are related to each other by arc if there is a boundary controls (l, r) such that $F(l.\lambda(v_1).r) = \lambda(v_2)$ where $\lambda(v_1)$ is the Boolean conversion of the vertex v_1 and F is the global dynamic function.

In this chapter, we study the problem of regional controllability of deterministic cellular automata by exploring an original approach that is of graph theory. This chapter is organized as follows: First, we start by the necessary notions of graph theory. In section 5.2, we give an overview of the problem of regional controllability. Section 5.3, is dedicated to the formulation of the problem using transition graphs and section 5.4 gives necessary and sufficient conditions for regional controllability. It first deals with the existence of a Hamiltonian Circuit in the graph representing the Boolean CA global evolution and then the decidability criterion of

regional controllability by establishing a relation with strongly connected component is given. According to this criterion, we give a classification of selected rules in the one-dimensional CA case. In section 5.5, we introduce a method to trace the configurations where a regional control is possible using a method based on preimages. Finally, a conclusion will be given in section 5.6.

First of all, we start by establishing some notions of graph theory which will be used further in this chapter:

Definition 27 *A graph G consists of a pair $(V(G), AR(G))$ where $V(G)$ is a non empty finite set whose elements are called points or vertices and $AR(G)$ is a set of unordered pairs of distinct elements of $V(G)$. The elements of $AR(G)$ are called lines or edges of the graph G.*

Example 5.1.1 *$G = \{V(G), AR(G)\}$ is a graph where:*
$V(G) = \{a, b, c, d, e\}$ $AR(G) = \{\{a, b\}, \{a, c\}, \{b, d\}, \{c, e\}, \{a, e\}, \{a, d\}\}$. Below is the graph with 5 vertices and 6 edges. Notice that each element of V is represented by a circle and that each element of AR is represented by a line drawn between the corresponding two elements.

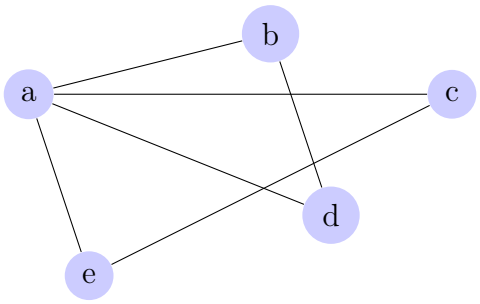


Figure .5.1 : Graph G

Definition 28 *The adjacency matrix, sometimes also called the connection matrix, of a simple labeled graph is a matrix with rows and columns labeled by graph vertices, with a 1 or 0 in position (v_i, v_j) according to whether v_i and v_j are adjacent or not. i.e, there is a link between v_i and v_j .*

Definition 29 *A directed graph is a graph in which the edge set consists of ordered pairs. The term directed graph is often abbreviated as digraph.*

Example 5.1.2 ***Graph with its adjacency matrix***

$$A_d = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

To which one can associate a graph:

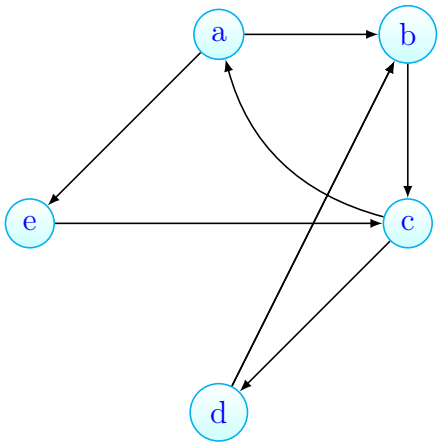


Figure .5.2 : Graph of the adjacency matrix A_d

5.2 Problem Statement

Let us consider:

- a 1D-cellular domain $\mathcal{L}_c$ of N cells,
- a discrete time horizon $I = \{0, 1, ..., T\}$,

- a sub-domain ω that defines the controlled region where we want to drive the CA towards a given configuration.

  It will contain n cells denoted by c_i, $i = 1, 2, \cdots, n$, $n < N$.

- $\overline{\omega} = \{c_0, c_1, \ldots, c_n, c_{n+1}\} = \omega \cup \{c_0, c_{n+1}\}$ where $\{c_0, c_{n+1}\}$ are the boundary cells of ω where we apply control.

We are interested in the problem of regional controllability defined in the last chapter.

Notation 3 *Let us introduce the following notations:*

$$\begin{aligned} \ell^t &= u_t(c_0) \\ r^t &= u_t(c_{n+1}) \\ s_i^t &= s_t(c_i) \end{aligned}$$

$\forall\ i.\ 1 \leq i \leq n$.

$(l \cdot x \cdot r)$ *is the concatenation operation describing the CA state on* $\overline{\omega}$ *where* $x = s(c_1), \ldots, s(c_n)$, $\ell = u(c_0)$ *and* $r = u(c_{n+1})$.

Problem 1 *Starting from an initial condition and for a given desired configuration* s_d*, the considered problem of regional controllability consists in finding the control required on the boundaries* $\{c_0, c_{n+1}\}$*, in order to get at time* T*, the configuration* s_d *in the controlled region* $\{c_1, \ldots, c_n\}$*, such that* $s_d(c_i) = s_T(c_i)\ \forall i = 1, \ldots, n$*, for a given time horizon* T*.*

Example 5.2.1 *Consider the Wolfram's rule* 90 *for which the evolution can completely be described by a table mapping the next state from all possible combinations of three inputs* (s_{-1}, s_0, s_{+1}) *according to the sum modulo 2 of the state values of the cells to its left and to its right* $s_{-1} \oplus s_{+1}$*:*

0000	011100	0000
0000	110110	0000
0000	110111	0000
	110101	

0000	011100	1000
0001	110111	0100
0001	010101	0010
	000000	

Figure .5.3 : The evolution of CA Wolfram rule 90 on the region $\omega = \{c_1, \cdots, c_6\}$ starting with the same initial configuration; on the left without control and on the right with control.

$$
\begin{array}{rlcl}
f_{90}: & 111 & \mapsto & 0 \\
 & 110 & \mapsto & 1 \\
 & 101 & \mapsto & 0 \\
 & 100 & \mapsto & 1 \\
 & 011 & \mapsto & 1 \\
 & 010 & \mapsto & 0 \\
 & 001 & \mapsto & 1 \\
 & 000 & \mapsto & 0
\end{array}
$$

For instance, with $n = 6$ (cf. Figure (.5.3)), if we assume starting at time 0 with an initial configuration $\{s_1^0, s_2^0, s_3^0, s_4^0, s_5^0, s_6^0\} = \{011100\}$ on $\omega = \{c_1, \cdots, c_6\}$ and given a desired null state on ω, there exists a control $u = (u_0, u_1, u_2)$ where $u_0 = (\ell^0, r^0) = (0, 1), u_1 = (\ell^1, r^1) = (1, 0), u_2 = (\ell^2, r^2) = (1, 0)$ that are applied on cells c_0, c_7, such that the final CA configuration on ω obtained at time $T = 3$ from the evolution of rule 90, is $\{s_1^3, s_2^3, s_3^3, s_4^3, s_5^3, s_6^3\} = \{000000\}$.

Remark 8 *The same problem can be defined on two-dimensional CA. For example, we can apply the control on one side of the boundary or on the whole boundary cells of the controlled region ω in order to get the desired state inside ω.*

Example 5.2.2 *Consider the following local evolution rule of a two dimensional CA given by the function:*

$$s_{t+1}(c_{i,j}) = s_t(c_{i-1,j}) \oplus s_t(c_{i+1,j}) \oplus s_t(c_{i,j-1}) \oplus s_t(c_{i,j+1})$$

We consider a controlled region given by the square $\omega = \{c_{1,1}, c_{1,2}, c_{2,1}, c_{2,2}\}$.

For a given initial configuration given by $\{s^0_{1,1}, s^0_{1,2}, s^0_{2,1}, s^0_{2,2}\} = \{1, 0, 0, 0\}$ *on* ω*, we first let the system evolve without applying controls and get the final configuration* $\{s^1_{1,1}, s^1_{1,2}, s^1_{2,1}, s^1_{2,2}\} = \{0, 1, 1, 0\}$ *at time* $T = 1$*.*

When looking for controls applied on the boundary cells of ω *in order to obtain a desired configuration consisting of 1s on* ω*, we obtain the values illustrated in red in Figure (.5.4).*

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$T = 0$ $\qquad$ $T = 0$

$$\begin{pmatrix} - & - & - & - \\ - & 0 & 1 & - \\ - & 1 & 0 & - \\ - & - & - & - \end{pmatrix} \qquad \begin{pmatrix} - & - & - & - \\ - & 1 & 1 & - \\ - & 1 & 1 & - \\ - & - & - & - \end{pmatrix}$$

$T = 1$ $\qquad$ $T = 1$

Figure .5.4 : Evolution of the CA rule .5.4 on ω in the autonomous and controlled cases on the left and right matrices respectively.

The above examples of Boolean CA in one and two dimensional cases show that it is possible to steer a system from an initial state to a desired target on a subregion of the domain by acting on its boundary. Our aim in this chapter is to generalise this results and find necessary and sufficient conditions for the regional controllability of Boolean CA. The proposed method will be based on transition graphs.

5.3 Transition graph approach and regional controllability problem

In this section, we recall the transition matrix described in the last chapter which is the main tool on which our work is based.

5.3.1 Transformation matrix

Recall that the evolution of controlled CA for one step can be represented by a directed graph where the vertices represent the configurations and the arcs represent the transition from a configuration to another one in one step *i.e.* by applying the global transition function F. Consider an Elementary CA where the controlled region ω is of size $|\omega|$ and controls are applied on its two boundary cells $\{c_0, c_{n+1}\}$. When considering the restriction of F on $\mathcal{S}^{|\omega|}$, there exists a bijection between $\mathcal{S}^{|\omega|}$ and the set of integers $[0 : 2^{|\omega|} - 1]$ that represents CA configurations on ω as $|\omega|$-bit binary numbers. Let λ be a vertex labelling such that for every vertex v, $\lambda(v)$ is the Boolean conversion of vertex v.

We define the transition graph $\Upsilon = (V, AR)$ as follows where the vertices V corresponds to each possible configuration of the region ω and AR is the set of arcs. Let v_1 and v_2 be two vertices in V, there is an arc from the vertex v_1 to the vertex v_2 if there exists a control $u = (\ell, r) \in \{(0,0); (0,1); (1,0); (1,1)\}$ such that $\lambda(v_2)$ is equal to $F_{|_{\mathcal{S}^{\overline{\omega}}}}(\ell \cdot \lambda(v_1) \cdot r)$, where the $\lambda(v_1)$ denotes the configuration in the controlled region at time t and $\lambda(v_2)$ denotes the configuration in the controlled region at time $t + 1$.

We denote by $\mathcal{C}$ the transition matrix which is the associate adjacency matrix of the graph Υ. The transition matrix is built as a Boolean matrix of size $2^{|\omega|} \times 2^{|\omega|}$. There is a 1 at position (i, j), the ith row and jth column, if there is an arc between vertices i and j for all i, j in $[0 : 2^{|\omega|} - 1]$. Otherwise it stays at 0.

We proceed as follows to construct the transition graph Υ (the algorithm is given at the end of this chapter). For each vertex v, we compute the four configurations (represented by u_1, u_2, u_3, u_4) obtained by the application of the global transition function $F_{|_{\mathcal{S}^{\overline{\omega}}}}$ to the four possible configurations obtained by the concatenation of the controls $((0,0); (0,1); (1,0); (1,1))$ on the extremities of v. Then we add an arc from v to each of the four u_i. In total, the time complexity to build Υ is $O(|V|)$ *i.e.* $O(2^{|\omega|})$ where $|\omega|$ is the size of the controlled region ω in the CA. The space complexity is the size of the Boolean matrix $\mathcal{C}$: $O(|V| \times |V|) = O(2^{|\omega|} \times 2^{|\omega|})$. Note that the number of arcs is at most $4 \times |V|$.

Example 5.3.1 *For instance, consider the rule* 30 *where the controlled region is of size* $|\omega| = 2$ *for more simplicity. The corresponding graph is represented in Figure (.5.5).*

The corresponding table for rule 30 is:

$$\begin{array}{lccc} f: & 111 & \mapsto & 0 \\ & 110 & \mapsto & 0 \\ & 101 & \mapsto & 0 \\ & 100 & \mapsto & 1 \\ & 011 & \mapsto & 1 \\ & 010 & \mapsto & 1 \\ & 001 & \mapsto & 1 \\ & 000 & \mapsto & 0 \end{array}$$

Consider vertex 2 *whose binary conversion is* 01*. We have* $F_{|_{\mathcal{S}^{\overline{\omega}}}}(0010) = F_{|_{\mathcal{S}^{\overline{\omega}}}}(0011) = 11$ *and* $F_{|_{\mathcal{S}^{\overline{\omega}}}}(1010) = F_{|_{\mathcal{S}^{\overline{\omega}}}}(1011) = 01$*. Therefore there are two arcs from* 2*:* $(2, 2)$ *and* $(2, 3)$ *as the binary conversion of* 3 *is* 11*.*

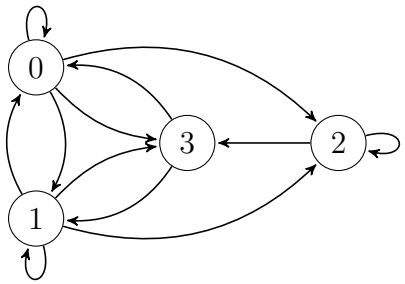


Figure .5.5 : Transition graph Υ for the CA rule 30 where the region to be controlled is of size 2 and $\lambda(0) = 00$, $\lambda(1) = 10$, $\lambda(2) = 01$, $\lambda(3) = 11$.

5.4 Characterising regional controllability for Boolean deterministic CA

5.4.1 Necessary and Sufficient Condition

A Hamiltonian circuit in a graph G is defined as a graph cycle that traverses every vertex exactly once and returns to the starting vertex. In other words, the Hamiltonian circuit is a path that starts from an original vertex and passes to all the vertices once, except the original vertex. The Hamiltonian circuit is named after inventing a puzzle game in 1857 by the mathematician William Rowan Hamilton, which involved searching for a Hamiltonian cycle. The Hamiltonian Circuit problem is whether there is a Hamiltonian circuit in graph. This problem turned out to be hard to solve. It is known that this problem belongs to the class of NP Complete problem [59, 68]. This was shown by Karp in [71]. The search of a Hamiltonian circuit has real applications in a wide range fields. For instance, pizza delivering use Hamiltonian cycles to plan the best route to deliver pizza to the customers houses. Here the customers houses are considered as nodes, the path between them represents edges and the bike will go to each customers houses exactly once.
In this section, we prove the regional controllability for one-dimensional and two-dimensional CA using a method based on the existence of a Hamiltonian circuit. The CA is regionally controllable if all the states are reachable in the target region (starting from each vertex we can reach another vertex in finite number of steps). The existence of a Hamiltonian circuit ensures that all vertices (configurations) are visited once and ensures that it exists a time T such as all the configurations are reachable.

Definition 30 *[85] A Hamiltonian circuit of a graph $G = (V, AR)$ is a simple directed path of G that includes every vertex exactly once.*

Example 5.4.1 *The following graph contains a Hamiltonian circuit* $1 \to 3 \to 2 \to 4 \to 6 \to 5 \to 1$.

Notation 4 *We introduce new notation $a \rightsquigarrow b$, this notation means that there is a directed path between a and b. In other words, there exists vertices $v_1, v_2, \ldots, v_i$ such that $(a, v_1), (v_1, v_2), \ldots, (v_{i-1}, v_i), (v_i, b)$ are arcs in AR.*

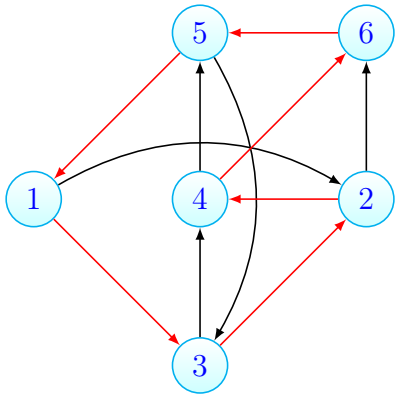


Figure .5.6 : Hamiltonian Circuit

Theorem 5.4.2 *A Cellular Automaton is regionally controllable iff there exists a t such that the graph associated to the transformation matrix $\mathcal{C}^t$ contains a Hamiltonian circuit.*

Proof 5.4.3 *Let us start with the first implication. Let $\Upsilon = (V, AR)$ be the transition graph built in Section 5.3 for a CA with a controlled region of size $|\omega|$ and $V = \{v_1, \ldots, v_{2^{|\omega|}}\}$. The graph Υ will be represented by an adjacency matrix $\mathcal{C}$. Let G_1 be the transition graph associated to the matrix $\mathcal{C}^t$. The proof is based on the following property in graph theory: the (i, j)th entry of the matrix $\mathcal{C}^t$ corresponds to the number of paths of length t from vertex i to j.*

Assume that the CA is regionally controllable at time $T \geq t$. Then some configurations can be reached in less than T steps from any other one. That means that each pair of vertices are linked by a directed path of length at most equal to T. Therefore, $\mathcal{C}^T$ will be strictly positive as reported in the theorem in [36] which states that the CA is regionally controllable if there exists a power T such that $\mathcal{C}^T > 0$. The associated graph G_T to the matrix $\mathcal{C}^T$ is therefore a complete graph and it is trivial to find a Hamiltonian circuit in a complete graph which implies that there exists $t \leq T$ such that the graph related to the matrix $\mathcal{C}^t$ contains a Hamiltonian Circuit and the direct implication holds.

To prove the converse one, let us assume that G_1 contains a Hamiltonian circuit. This means that there is a directed path that goes through all the vertices

once. Therefore there exists an order $i_1, i_2, \ldots, i_{2^{|\omega|}}$ *such that:* (v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), $\ldots$, $(v_{i_{2^{|\omega|}-1}}, v_{i_{2^{|\omega|}}})$, $(v_{i_{2^{|\omega|}}}, v_{i_1})$ *are arcs in A. And then we have:*

$$v_i \rightsquigarrow v_j \quad \forall\ i, j \in \{1, \ldots, 2^{|\omega|}\} \text{ and } i \neq j$$

Thus, $\exists T \geq t$ *such that all the vertices (configurations) are reachable. And the theorem holds.*

Example 5.4.4 *The associated graph to the rule* 150 *where* $|\omega| = 2$. *This CA is controllable at time* $T = 1$ *the graph related to the adjacency matrix* $\mathcal{C}$ *contains a Hamiltonian Circuit.* $(2 \to 0 \to 1 \to 3 \to 2)$.

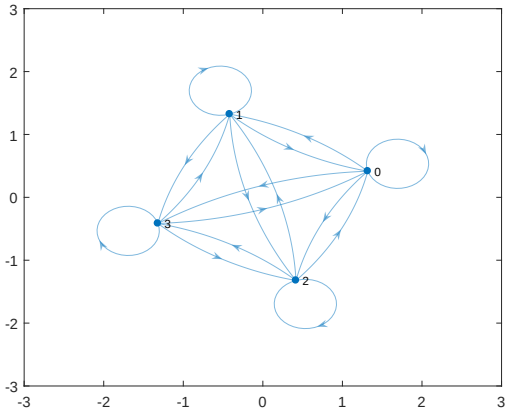


Figure .5.7 : Graph of $\mathcal{C}$ contains a Hamiltonian Circuit

As the problem of proving the existence of a Hamiltonian circuit in a graph is NP-complete, the time complexity can be exponential in the number of vertices of the transformation graph. We improve this criterion in the next section with a solution in polynomial time that gives a necessary and sufficient condition.

5.4.2 Necessary and Sufficient Condition (in polynomial time)

In graph theory, a directed graph is composed by nodes. These nodes are usually not connected together by edges but in some cases there is a path which links each

two vertices of the graph for that the notion of strongly connected component is confirmed.

In this section, we give a necessary and sufficient condition for the problem of regional controllability by using the strongly connected component.

Definition 31 *[85] A strongly connected component (SCC for short) of a directed graph G is a maximal set of vertices $C \subset V$ such that for every pair of vertices v_1 and v_2 in C, there is a directed path from v_1 to v_2 and a directed path from v_2 to v_1.*

Example 5.4.5 *We give a graph which contains 3 strongly connected component $\{\{a, b, c\}, \{d, e, f\}, \{h, g\}\}$ which are represented in the following graph:*

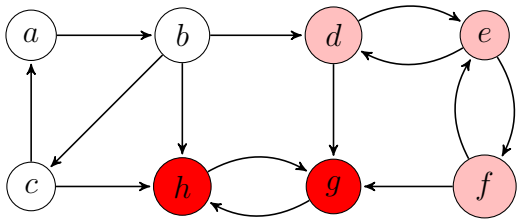


Figure .5.8 : Graph with a marked strongly connected components

Theorem 5.4.6 *A CA is regionally controllable for a given rule iff the transition graph Υ associated to the rule has only one SCC.*

Proof 5.4.7 *Let $\Upsilon = (V, AR)$ be the transition graph built in Section 5.3 from a controlled region of size $|\omega|$.*

Assume that the graph contains only one SCC. There exists a directed path which relates each pair of vertices of the graph. Hence there is a sequence of controls that permits to go from every configuration to any other one. The CA is then controllable on ω and the direct implication holds.

Assume now that the graph Υ contains more than one SCC, let us say that it contains two. Then, the set of vertices can be divided in two sets related to each SCC such as:

$$V_1 = \{v_1, \ldots, v_k\} \quad V_2 = \{v_{k+1}, \ldots, v_{2^{|\omega|}}\}$$

and there is no arc between V_1 and V_2. Therefore, there is no control that allows to obtain a configuration represented in V_1 from a configuration in V_2 according to the construction of Υ. It is impossible since the CA is regionally controllable and so the converse implication holds.

Time complexity To find the SCCs, we have used Tarjan's algorithm [87] which has a linear time complexity: $O(|V| + |AR|)$ on the graph $\Upsilon = (V, AR)$. If we consider a controlled region ω of size $|\omega|$ and since in that case $|AR| \leq 4|V|$, then the time complexity is $O(|V|) = O(2^{|\omega|})$.

Remark 9 *The regional controllability depends on the rule and the size of the controlled region. The size of the controlled region for the same rule has an impact on the number of SCCs. According to Theorem 5.4.6, by changing the size of CA, a rule can be sometimes regionally controllable and sometimes not.*

In Table 5.4.2 the results of our simulations are highlighted.

Classification of some rules of one-dimensional CA

Rules	Decidability Criterion	number of SCC
0,255	not cont	64 for $\|\omega\| = 6$ and 16 for $\|\omega\| = 4$
1	not cont $\|\omega\| = 4$, cont $\|\omega\| = 2$	8 for $\|\omega\| = 4$ and 1 for $\|\omega\| = 2$
60,90,102,150,170	cont	1
204	not cont	1
2,4,8,16	not cont	1
105,195,165,153,85	cont for all the sizes of CA	1
22	cont for $\|\omega\| = 2$, not cont $\|\omega\| = 5$	1 for $\|\omega\| = 2$ and 1 for $\|\omega\| = 5$
26	cont $2 \leq \|\omega\| \leq 3$, not cont $\|\omega\| = 4$	1 for $2 \leq \|\omega\| \leq 3$, 2 for $\|\omega\| = 4$
233	not cont	1
3	cont $\|\omega\| = 2$, otherwise not cont	1 for $\|\omega\| = 2$, otherwise 1

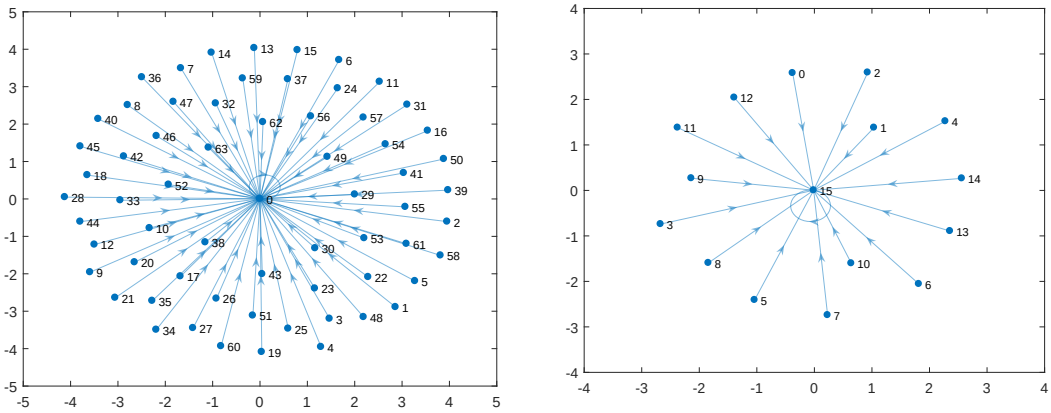
To illustrate the obtained results, we shall give in the following section, some examples in both one and two dimensional cellular automata.

We give some examples of one dimensional and two-dimensional cellular automata.

5.4.3 Examples

According to the Wolfram classification of one dimensional CA, we give some examples:

Example 5.4.8 *Wolfram Rule* 0 *is not controllable neither its Boolean complement rule* 255 *as they converge to a fixed point (they belong to the class 1 according to the Wolfram classification). The graph of their matrix* $\mathcal{C}$ *contains more then one strongly connected component and the previous theorem states that these rules are not regionally controllable for every region* ω*, see Figure (.5.9).*

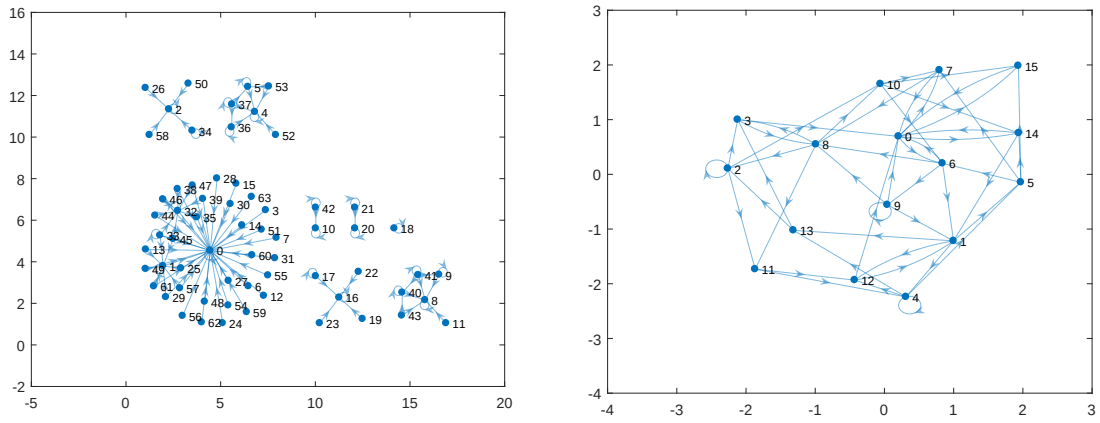


(a) rule 0, $|\omega| = 6$.

(b) rule 255, $|\omega| = 4$.

Figure .5.9 : Graphs related to the matrices $\mathcal{C}_0$ and $\mathcal{C}_{255}$ respectively.

Example 5.4.9 *Let us consider the two Wolfram rules* 4 *and* 37 *[94] which belong to the class* 2 *whose rules converge to simple separated periodic structures. For a controlled region of* $|\omega| = 6$ *and* $|\omega| = 4$*, the graph of the matrix* $\mathcal{C}$ *associated to these two rules is illustrated in Figure (.5.10). The associated graph to the rule* 4 *contains more then one strongly connected component while the graph associated to the rule* 37 *contains one strongly connected component which means that there exists a time* T *where each configuration is reachable.*



(a) More than one SCC associated to the graph of rule 4.

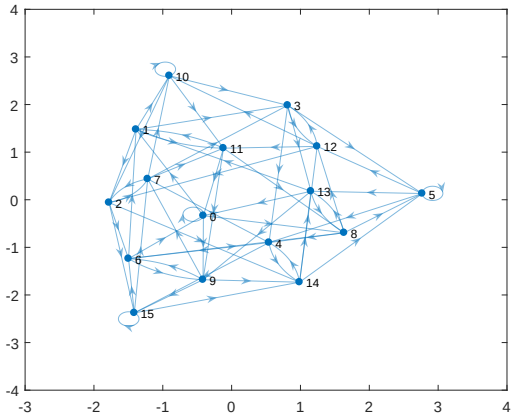
(b) One SCC associated to the graph of rule 37.

Figure .5.10 : Graphs of the matrices $\mathcal{C}_4$ and $\mathcal{C}_{37}$ respectively.

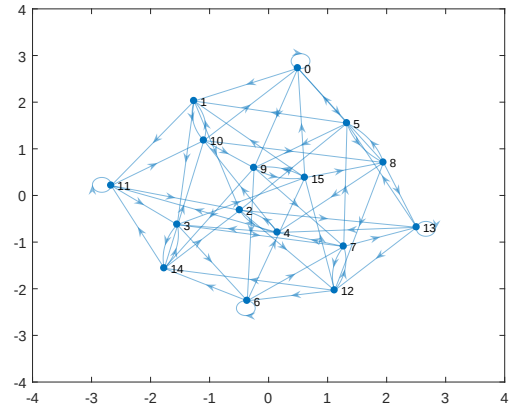
Example 5.4.10 *Let us consider the two linear Wolfram rules* 150 *and* 90 *[94] which belong to the class* 3 *the class of rules whose evolution yields chaotic periodic patterns (chaotic behaviour). For a controlled region of* $|\omega| = 4$*, the graph of the matrix* $\mathcal{C}$ *associated to these two rules is illustrated in Figure (.5.11). It contains one strongly connected component which means that there exists a time* T *where each configuration is reachable.*

Example 5.4.11 *Let us consider the Wolfram rule* 110 *[94] which belongs to the class* 4 *which is the class of rules whose evolution leads to a complex behaviour. For a controlled region of* $|\omega| = 6$*, the graph of the matrix* $\mathcal{C}$ *associated to this two rule is illustrated in Figure (.5.12). It contains more than one strongly connected component.*

Example 5.4.12 *Finally, an example with rule* 1 *is given to show that the decidability criterion for regional controllability may change for the same rule, according to the size of* ω*. With a region of size* $|\omega| = 6$*, the graph of the matrix* $\mathcal{C}_1$ *contains more*



(a) One SCC associated to the graph of rule 150.



(b) One SCC associated to the graph of rule 90.

Figure .5.11 : Graphs of the matrices $\mathcal{C}_{150}$ and $\mathcal{C}_{90}$ respectively.

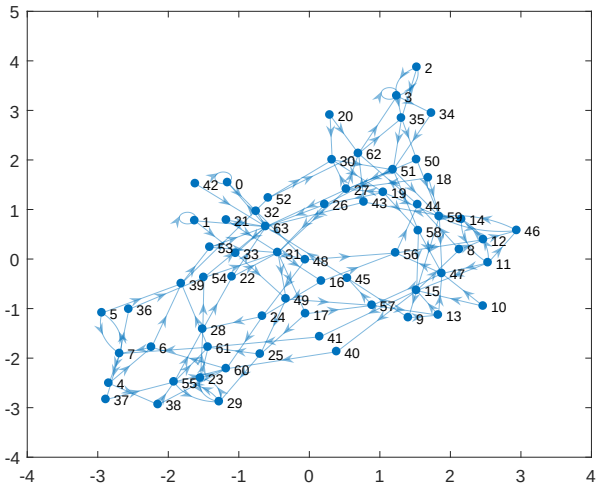
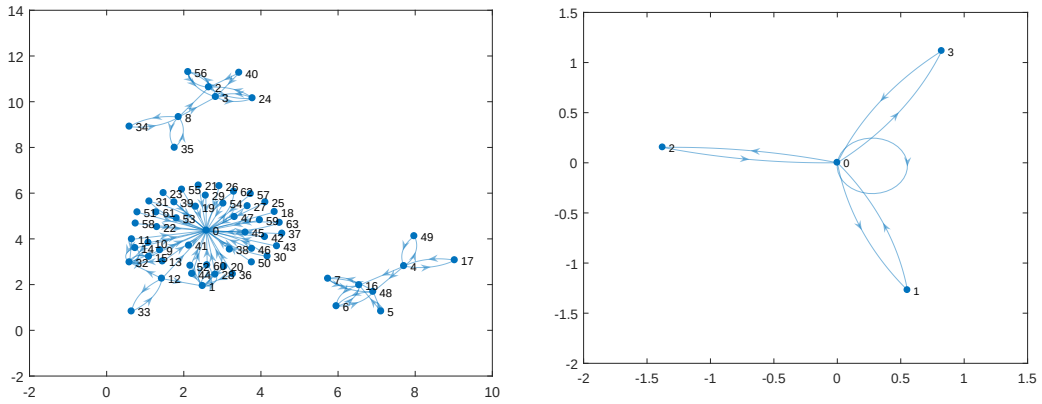


Figure .5.12 : Graph of the matrix $\mathcal{C}_{110}$.

than one SCC while for $|\omega| = 2$, it contains only one strongly connected component. Consequently, the CA is not regionally controllable in the first case and regionally controllable in the second one.

Example 5.4.13 *Let us consider now a two-dimensional cellular automaton. Its*



(a) rule 1, $|\omega| = 6$.

(b) rule 1, $|\omega| = 2$.

Figure .5.13 : Graphs of the matrix $\mathcal{C}_1$ for two sizes of ω.

local evolution is given by the transition function:

$$s^{t+1}(c_{i,j}) = s^t(c_{i-1,j}) \oplus s^t(c_{i+1,j}) \oplus s^t(c_{i,j-1}) \oplus s^t(c_{i,j+1})$$

that is also denoted by rule 170 *using Wolfram's formalism. We impose asymmetric controls by setting all cells on the boundaries to* 0 *except for the two cells on the top of the controlled region as illustrated in Figure (.4.9).*

The obtained graph of the matrix $\mathcal{C}_{170}$ *for* $|\omega| = 2 \times 2, \quad \omega = \{c_{1,1}, c_{1,2}, c_{2,1}, c_{2,2}\}$, *contains one strongly connected component and so the CA is regionally controllable.*

5.5 Pre-images of a regional controlled area

The famous criterion used to decide whether a system is controllable or not is the Kalman rank condition. This criterion does not give any information about the controls used to obtain a given desired state starting from an initial state. The approach used in this chapter gives an information about the controls. In this part, we give an efficient algorithm to create the preimages.

Let $\{s_1^i, s_2^i, \ldots, s_n^i\}$ be the configuration at time i of the region to be controlled. The idea is to find a boundary control given as a sequence $(\ell^0, r^0), (\ell^1, r^1), \ldots, (\ell^{T-1}, r^{T-1})$

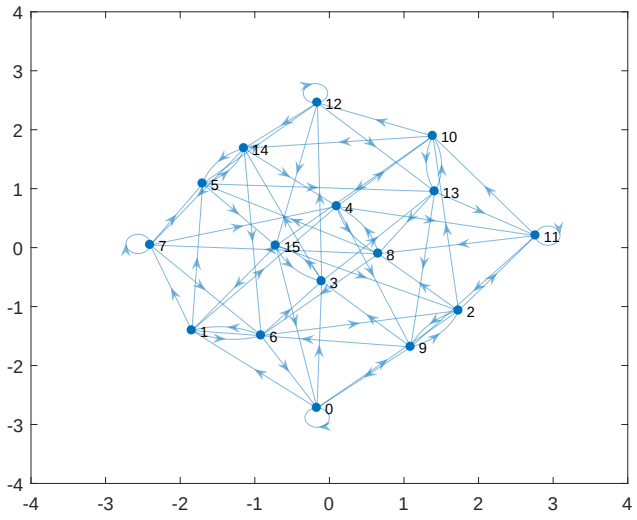


Figure .5.14 : Graph of the matrix $\mathcal{C}_{170}$ in two dimensional CA

so as to obtain a desired configuration $\{s_1^T, s_2^T, \ldots, s_n^T\}$ at time T from an initial one $\{s_1^0, s_2^0, \ldots, s_n^0\}$.

Let us define in what follows some needed notions and present the data structure required to solve the problem.

5.5.1 Distance function

We define the distance function

$$\Delta_i : \text{vertex} \mapsto \text{list of vertices}$$

that associates to each vertex $v \in \Upsilon$, the list of vertices from which v can be reached within a path of length exactly i.

Where Υ is the transition graph introduced in Section 5.3.

Therefore $\Delta_i(v)$ gives all initial configurations from which the desired configuration v can be reached in exactly i steps with the application of control.

Let T be the time at which we want to reach a desired state. Representing Δ as a map, we shall consecutively construct the functions Δ_i by searching the predecessors

from Δ_{i-1}. We start by Δ_1 and go until Δ_T. The algorithm is given at the end of this chapter.

If δ is the maximum number of predecessors of a vertex, the time complexity is $O(T \times \delta \times |V|) = O(T \times \delta \times 2^{|\omega|})$.

5.5.2 Path controllability

We address two problems in this section.

Problem 2 *Find one state configuration that can be driven to a desired state configuration b_f in k steps and the relative control sequence.*

To solve this problem, we construct the distance function Δ_i for $1 \leq i \leq k$. Then we consider the ancestors b_i at distance k of b_f (stored in $\Delta_k(b_f)$). If there is no ancestor, that means that it is not possible to reach this state in k steps. Otherwise, we can find the path of length k with end extremity b_f. To do so, we pick one predecessor of b_f, say b_{k-1}, that can be reached in $k-1$ steps, *i.e.* one among the vertices in the list $\Delta_{k-1}(b_f)$. Then we search the first one among the predecessors of b_{k-1}, say b_{k-2}, that can be reached in $k-2$ steps, *i.e.* one among the vertices in the list $\Delta_{k-2}(b_f)$ and so forth until we find b_1. We obtain the path of configurations $b_1, \ldots, b_{k-1}, b_f$. It just remains to find the appropriate control by applying the rules to each configuration extended with the boundaries $(0,0), (0,1), (1,0), (1,1)$.

In total, once the distance function is built, it takes $O(k)$ time.

Problem 3 *Find all the needed controls and the intermediate states of the controlled region required to obtain a desired state b_f in exactly k steps (*i.e.*) from a state configuration b_1.*

For this problem, instead of checking if the list of ancestors is not empty (and taking one among the vertices), we need to check if among the ancestors there is the initial configuration b_1.

Therefore, the time complexity is $O(k \times \delta)$.

We present; the algorithms to construct the data structures used in the chapter.

Construction of the transition graph Υ.

```
Algorithm transGraph(d_ω, F)
    d ← d_ω
    d_Υ ← 2^d
    Υ ← [0]_{|d_Υ × d_Υ}                          (zero matrix of size d_Υ
    forall 0 ≤ i < d_Υ                             for every configuration i)
        λ(v_1) ← F(0 · ĭ|_d · 0)
        λ(v_2) ← F(0 · ĭ|_d · 1)
        λ(v_3) ← F(1 · ĭ|_d · 0)                   (add the boundary controls
                                                    and apply the rule)
        λ(v_4) ← F(1 · ĭ|_d · 1])
        forall 0 ≤ j < d_Υ                         (for every configuration j)
            if λ(v_1), λ(v_2), λ(v_3) or λ(v_4) equal j̆_{|d}   (add an edge if there
                                                    is a boundary control)
                Υ(i, j) ← 1                        for i that leads to j)
    return Υ
```

Construction of the distance function.

```
Algorithm distanceFunction(G_Υ, k, v)
    Let d_Υ be the number of vertices of G_Υ
    Let Δ be an empty matrix of size d_Υ × k
    for each vertex v in G_Υ
        Δ(v, 1) ← predecessors(G_Υ, v)
    for 1 < ℓ ≤ k
        for each vertex v in G_Υ
            for each vertex u in Δ(v, ℓ − 1)
                add(Δ(v, ℓ), predecessors(G_Υ, u))
    return Δ
```

Finding the control in k steps to reach the desired state.

```
Algorithm pathControllability(G_Υ, k, v_init, v_desired)
    p ← [v_desired]
    Δ ← distanceFunction(G_Υ, k, v_desired)
    if v_init ∉ Δ(v, k)
        return p
    pred ← v_desired
    for i from k to 1
        predList ← predecessors(G_Υ, pred)
        pred ← find(x ∈ predList.v_init ∈ Δ(x, i))
        add(pred, p)
    return p
```

5.6 Conclusions

In this chapter, we considered the problem of regional controllability of cellular automata focusing on actions performed on the boundary of a target region. We established some necessary and sufficient conditions using graph theory tools.

This work is based on the transition matrix defined in the previous chapter. We showed that the existence of a Hamiltonian circuit guarantees the regional controllability.

Then we proved that a CA is regional controllable if the associated graph to the transition matrix contains a strongly connected component. The obtained results based on the existence of strongly connected component were successfully tested for some cases in 1D and 2D cellular automata.

To obtain the control that allows the system to reach the desired state during a given time horizon and starting from a given initial condition, an efficient algorithm for generating preimages was used.

In the following chapter, we will study the problem of regional controllability by actions performed on the boundaries of CA. We will propose some efficient algorithms which work for big sizes of CA, the one dimensional linear CA.

Kalman condition and some algorithms for regional controllability of Cellular Automata

CHAPTER 6

KALMAN CONDITION AND SOME ALGORITHMS FOR REGIONAL CONTROLLABILITY OF CELLULAR AUTOMATA

Résumé:

Dans ce chapitre [1], on a abordé l'étude du problème de la contrôlabilité frontière des Automates Cellulaires en découvrant des nouvelles pistes de preuves incluant des algorithmes. Ce travail est une contribution à l'étude numérique du problème de la contrôlabilité frontière des automates cellulaires unidimensionnels en agissant sur la frontière du domaine pour atteindre un objectif sur une région. On a commencé par le cas linéaire où on a proposé un algorithme qui permet de trouver le contrôle requis sur la frontière du domaine afin de forcer l'apparition d'une configuration désirée dans une région contrôlée. Divers exemples illustratifs sont présentés. On s'intéresse aussi à la question: " Est ce possible de reconstruire les configurations précédentes à partir d'une configuration désirée restreinte à une région contrôlée à un temps donné". Ce qui revient à résoudre le problème connu par les préimages. à la fin, on

[1] Cette partie a fait l'objet de travail suivant:
S. Dridi, S. El Yacoubi, F. Bagnoli (to be submitted)
Boundary regional controllability of 1D cellular automata. Mathematical methods in the applied sciences.

a donné un résultat de caractérisation grâce à la condition de Kalman.

6.1 Introduction

R. Kalman has identified the notion of controllability as a one of the prominent proprieties for determining the system behaviour. The study of controllability for linear systems requires the simple rank condition which became ubiquitous in linear systems analysis. In this chapter, we study the problem of boundary regional controllability of deterministic cellular automata by exploring a new algorithms. We investigate one-dimensional linear deterministic Cellular Automata. This chapter is divided as follows: In section 6.1, we start by introducing the problem of boundary regional controllability of CA. Section 6.3 is devoted to numerical applications where a suitable algorithm is proposed and illustrated with examples. The case of the complement of linear CA is also investigated. The necessary condition is being proved by using the known Kalman criterion for the linear case. An extension to the more general affine rule is also presented In section 6.4, we have proposed an algorithm for generating the preimages. A conclusion is given in the last section.

6.2 Problem Statement

Let us consider a Boolean CA defined on a lattice $\mathcal{L}_c$ that is assumed to be finite and composed of N interior cells and 2 boundary cells. We denote by c_l and c_r the left and right boundary cells respectively. The boundary conditions for the autonomous CA are assumed to be fixed at 0 (fixed boundary conditions).

Let $\omega = \{c_{posmin}, \ldots, c_{posmax}\}$ be a subset of the 1-D lattice where the desired configuration intends to be reached. We are interested in finding the suitable sequences of controls acting on the boundary of the lattice, $(u_l^0, u_l^1, \ldots, u_l^{T-1})$ and $(u_r^0, u_r^1, \ldots, u_r^{T-1})$, so as to steer the system from a given initial state s_0 to a desired configuration s_d on the subregion ω at a given time T, such that: $s_T(c_i) = s_d(c_i) \quad \forall i = posmin, \ldots, posmax$, where $s_T(c_i)$ and $s_d(c_i)$ are the final and desired state of cell c_i at time T, respectively. The desired configuration s_d is assumed to be reachable in the evolution of the CA rule.

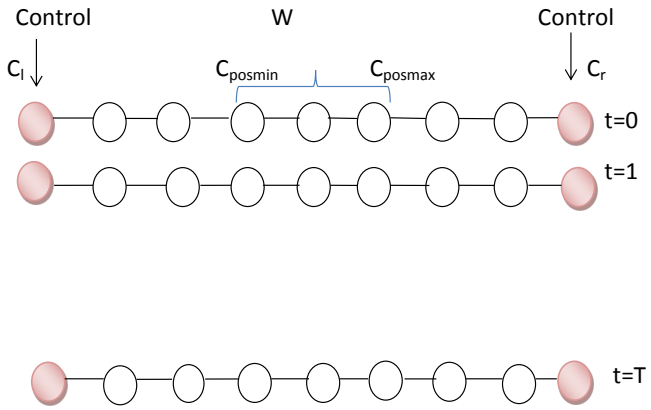


Figure .6.1 : Boundary regional control of one dimensional cellular automata.

6.3 Regional controllability via boundary actions of linear Boolean CA:

Linear global CA dynamics that governs the evolution of a given initial configuration over discrete time steps, is associated with a linear local transition function that calculates the $(t+1)^{\text{th}}$ state of cell c_i by : $s_{t+1}(c_i) = f(s_t(c_{i-k}), \cdots, s_t(c_{i+k})) = a_{-k}s_t(c_{i-k}) \oplus a_{-k+1}s_t(c_{i-k+1}) \cdots a_k s_t(c_{i+k}) \quad a_i \in \{0,1\} \quad \forall i = -k, \ldots, k$ where $\oplus$ denotes the XOR operation or "sum mod 2". f is a Boolean function of $r = 2k+1$ Boolean arguments.

Since all variables take Boolean values and because of the discreteness of all the parameters of the system, the usual differential operators cannot be applied.

For this purpose, an equivalent notion introduced in [8, 89] called Boolean derivatives was defined and used to study the changes in a global CA state according to infinitesimal changes in the cell states. It locally measures the changes of f when changing one of its arguments. This definition fulfills many of the standard properties of the derivative, and this is particularly evident if one expresses the function f using only AND (multiplication) and XOR operations. The notions of partial Boolean derivatives as well as the Jacobian matrix were defined first used for control issues in [12] and the references therein.

Let us recall in what follows, some useful definition.

Definition 32 *Let f be a local transition function of a 1D-CA defined according to a symmetric neighbourhood of size r:*
$f : S^r \longrightarrow S$ such that $s_i^{t+1} = f(s_{i-k}^t, \cdots, s_{i+k}^t)$ where $r = 2k+1$ and k is the radius. For more simplicity, as with the time assumed to be constant, the exponent t may be removed. The Partial derivative of f is defined as:

$$\frac{\partial f}{\partial s_j} = f(s_{i-k}, \ldots, s_j \oplus 1, \ldots, s_{i+k}) \oplus f(s_{i-k}, \ldots, s_j, \ldots, s_{i+k})$$

Definition 33 *[89] Let us denotes by $f_i \overset{not}{=} f(s_{i-k}, \cdots, s_{i+k})$ and let F be the associated CA global dynamics defined on S^N that updates the CA configuration at each*

time step. The Boolean derivative F' of F is the Jacobian matrix defined as:

$$J = \frac{\partial f_i}{\partial s_j}|_{i,j=1,\dots,N}$$

where N is the size of the CA lattice.

Definition 34 *[12] A function*

$$s' = f(s_{i-k}, s_{i-k+1}, \dots, s_i, s_{i+k-1}, s_{i+k})$$

is peripherally linear if

$$\frac{\partial s'}{\partial s_{i+k}} = 1 \quad or \quad \frac{\partial s'}{\partial s_{i-k}} = 1$$

it is said to be double peripherally linear if

$$\frac{\partial s'}{\partial s_{i+k}} = 1 \quad and \quad \frac{\partial s'}{\partial s_{i-k}} = 1$$

k is the neighbourhood radius.

6.3.1 Controllability in finite dimension

Let us consider a linear system which can be written as:

$$\begin{cases} z'(t) = Az(t) + Bu(t); 0 < t < T \\ y(t) = Cz(t) \end{cases} \tag{6.1}$$

Where $z \in \mathbb{R}^n$ is the state vector, $y \in \mathbb{R}$ is the output vector and $u \in L^2[0, t; \mathbb{R}^m]$ is the control vector. The matrices A, B and C are of $n \times n$, $n \times m$ and $p \times n$, respectively, and represent the dynamic, control and observation. The study of the controllability problem consists to find an answer to the following question: Is it possible to reach all the desired state starting from any initial state. In finite dimension the characterisation of controllability is related to the determination of the controllability matrix which can be defined as follows:

Definition 35 *Consider the system 6.1, the controllability matrix is a matrix of dimension $n \times nm$ defined by:*

$$\mathcal{C}_K = [B, AB, \ldots, A^{n-1}B]. \tag{6.2}$$

The determination of the matrix gives an information whether the system is controllable or not. We have the following theorem (Kalman condition):

Theorem 6.3.1 *(Kalman condition) The system 6.1 is controllable if and only if the controllability matrix is of full rank, in other words:*

$$rank(\mathcal{C}_K) = n. \tag{6.3}$$

6.3.2 A characterization result

Example 6.3.2 *Lets us start with an illustrative example of CA rule 150. The rule is defined by the local transition function:*

$$s_i^{t+1} = s_{i-1}^t \oplus s_i^t \oplus s_{i+1}^t$$

The partial derivative of this rule with respect to s_{i-1} is

$$\frac{\partial f}{\partial s_{i-1}} = (s_{i-1} \oplus 1 \oplus s_i \oplus s_{i+1}) \oplus (s_{i-1} \oplus s_i \oplus s_{i+1})$$

so on:

$$\frac{\partial f}{\partial s_{i-1}} = 1$$

For a finite chain without boundary periodic conditions, the Jacobian matrix takes the form:

$$J = \begin{pmatrix} 1 & 1 & \ldots & \ldots & \ldots & 0 & 0 & 0 \\ 1 & 1 & 1 & \ldots & \ldots & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & \ldots & 0 & 0 & 0 \\ \ldots & \ldots & 1 & 1 & 1 & .. & \ldots & \\ \ldots & \ldots & \ldots & 1 & 1 & 1 & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & 1 & 1 & 1 & \ldots \\ 0 & 0 & 0 & \ldots & \ldots & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{6.4}$$

Let us consider a linear cellular automata which can be written as:

$$s^{t+1} = Js^t \oplus BU^t,$$

Theorem 6.3.3 *A 1D linear Cellular Automata is regionally controllable via boundary actions iff:*

$$Rank(\mathcal{C}_K) = Rank(B \quad JB \quad J^2B \dots J^{T-1}B) = T = N - 1$$

Where T is the time horizon, N is the size of CA lattice and J is the Jacobian matrix such as:

$$s^{t+1} = Js^t \text{ for all } 0 \leq t \leq T - 1$$

Proof 6.3.4 *It is clear that if the rank($\mathcal{C}_K$) $= T = N - 1$, the CA is regionally controllable.*

Assume that the CA is regionally controllable, it exists a sequence of control $(u^0, u^1, \dots, u^{T-1})$ such as:

$$\begin{cases} s^1 = Js^0 + Bu^0 \\ s^2 = Js^1 + Bu^1 \\ \vdots \\ s^T = Js^{T-1} + Bu^{T-1} \end{cases}$$

We have:

$$s^2 = Js^1 + Bu^1 = J^2s^0 + JBu^0 + Bu^1$$

$$s^3 = Js^2 + Bu^2 = J^3s^0 + J^2Bu^0 + JBu^1 + Bu^2$$

Therefore:

$$s^T = J^Ts^0 + (B \quad JB \quad \dots \quad J^{T-1}B)(u^{T-1} \quad u^{T-2} \quad \dots \quad u^0)^T$$

We define the controllability matrix $\mathcal{C}_K$*:*

$$\mathcal{C}_K = (B \quad JB \quad \dots \quad J^{T-1}B)$$

We get the boundary regional controllability when:

$$Rank(\mathcal{C}_K) = T$$

For peripheral linear CA, we know also that for being able to change any state of region $\omega = \{c_1, c_2, \dots, c_N\}$ *from one boundary* c_r *(the right one) the time* T *should equal* N*, where* N *is the size of CA. This result has been verified only numerically. So we get the boundary controllability for linear CA when:*

$$Rank(\mathcal{C}_K) = T = N - 1$$

Example 6.3.5 *Let consider a 1D linear cellular automaton consisting of* $N = 8$ *cells and rule* 150 *defined by the local transition function* $s_i^{t+1} = s_{i-1}^t \oplus s_i^t \oplus s_{i+1}^t$*. The required time to control this CA from one boundary is* $T = 7$ *and the vector* B *and the Jacobian matrix* J *of this rule are given as follows:*

$$J = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

The matrix of controllability $\mathcal{C}_K$ *is given as follows:*

$$\mathcal{C}_K = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 6 \\ 0 & 0 & 0 & 0 & 1 & 5 & 20 \\ 0 & 0 & 0 & 1 & 4 & 14 & 44 \\ 0 & 0 & 1 & 3 & 9 & 25 & 69 \\ 0 & 1 & 2 & 5 & 12 & 30 & 76 \\ 1 & 1 & 2 & 4 & 9 & 21 & 51 \end{pmatrix}$$

It is clear that the matrix is composed of linearly independent vectors lines and for that we have: $Rank(\mathcal{C}_K) = 7$

We shall now consider a numerical approach of the boundary regional controllability problem of linear elementary cellular automata (ECA) which is a particular case of Boolean CA. We will consider two kinds of boundary controls, from one side and from both sides.

6.3.3 New algorithm to solve boundary regional controllability of linear ECA

For controlling peripheral linear rules, one only needs to apply the control from one side [12]. Therefore it is possible to reach a desired state in a region by applying the control from just one boundary as illustrated in Figure (.6.2). Let us propose the

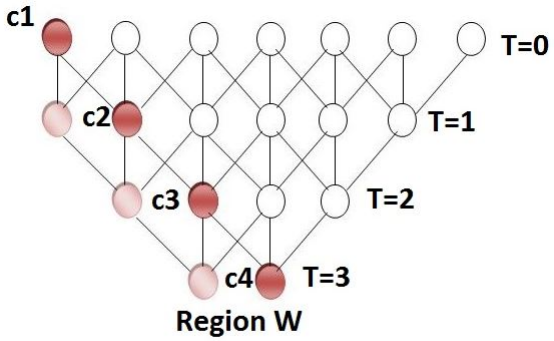


Figure .6.2 : Control of the region ω by applying the control on one boundary c_1. The change in the initial condition on the controlled cell c_1 will be propagated to the cell c_4 at time $T = 3$.

following algorithm:

6.3.4 The proposed algorithm

Input The CA size, the parameters related to the linear rule i.e a_{-1}, a_0, a_1, the initial CA configuration, time T, the desired state and the border cells of the controlled region c_{posmin}, c_{posmax}.

Step 1 According to the cell c_{posmax} and using the neighbourhood dependence at the previous time, go up to search the line of cells in the boundaries where we apply the control on which the cell indexed with *posmax* in the region depends as illustrated in Fig (.6.2).

Step 2 Do the evolution of cellular automata in the top part which does not depend on the control choice (the white cells in Fig (.6.2)).

Step 3 Determine the controls required in the boundary c_l which are depending on the desired state by using the considered rule

Step 4 Update cells state of the inferior part

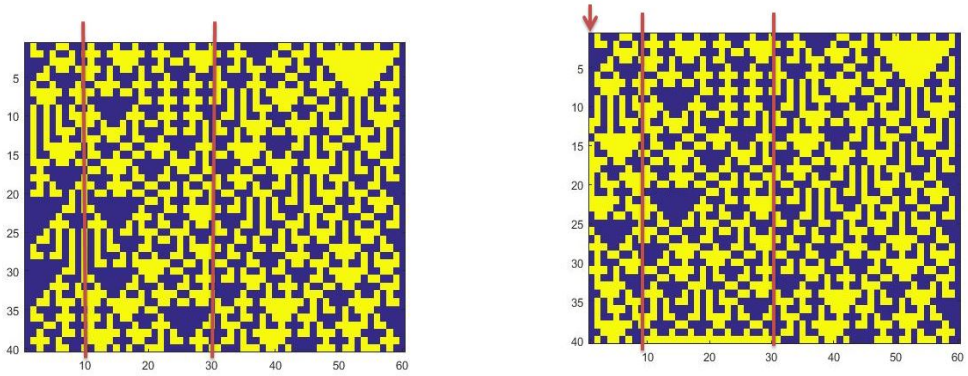
6.3.5 Simulation Examples

Example 6.3.6 *Consider the example of the linear rule f_{150}. There exists a sequence of controls $u = \{u^0_{c_l}, \ldots, u^{T-1}_{c_l}\}$ applied to the left boundary cell, that is c_l which steers the system in this region $\omega = \{c_{10}, \ldots, c_{30}\}$ from the initial configuration to a desired one $s_d = (1, \cdots, 1)$ at time $T = 39$. In the following figures, we illustrate the evolution of CA rule 150 with and without controls see Figure (.6.3).*

Double peripherally linear CA are optimally controllable [12]. In this case we prove that we can get the desired state in minimum time as illustrated below with the same rule 150.

Therefore we divide the region ω on two parts $\omega_r = \{c_{posmin}, \ldots, c_d\}$ and $\omega_l = \{c_{d+1}, \ldots, c_{posmax}\}$ see Figure (.6.4):

Example 6.3.7 *Let us consider the 1D linear CA rule f_{90} defined on a lattice of size $N = 80$. We are looking of the control sequence for getting the desired state in the region $\omega = \{c_{26}, \ldots, c_{55}\}$ such as $\forall 26 \leq i \leq 55$, $s_{T_{min}}(c_i) = 1$. We distinguish three cases as illustrated in the Figure (.6.5).*



(a) (b)

Figure .6.3 : Evolution of the CA rule 150 in the autonomous and the controlled cases ($\omega = \{c_{10}, \ldots, c_{30}\}$) illustrated in figures (a) and (b) respectively. Yellow and blue squares designate cells at state 1 and 0 respectively.

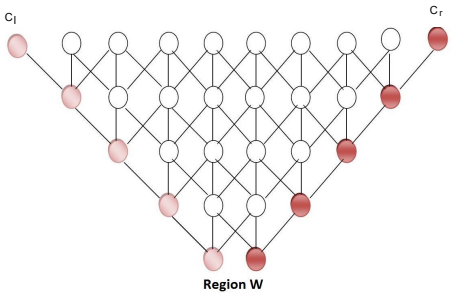


Figure .6.4 : Control the region ω by applying the control on the two boundaries

6.3.6 Extension to the case of complement of linear CA

In this part, we consider the problem of regional controllability via boundary actions for the logical complement of linear rules.

Example 6.3.8 *Consider the transition function corresponding to the rule* 105 *[94] which is the complement of rule* 150 *where the neighbourhood consists in adjacent cells and oneself, here is the difference between a rule* 150 *and its compliment* 105*:*

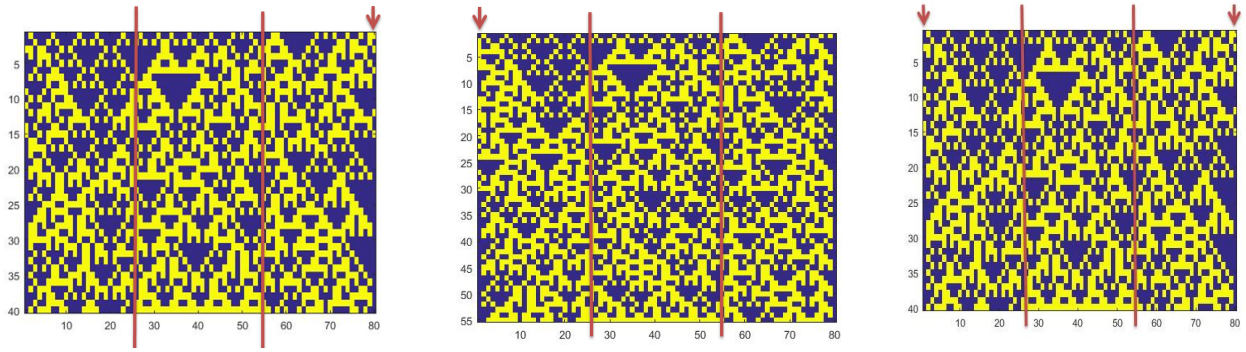


Figure .6.5 : Evolution of rule 90 from an initial configuration to a desired one at $T = 54$ by applying control on one side (right or left) at $T = 39$ and on both sides, the controlled region is $\omega = \{c_{26}, \ldots, c_{55}\}$

$$
\begin{array}{lccc}
f_{105}: & 000 & \mapsto & 1 \\
 & 001 & \mapsto & 0 \\
 & 010 & \mapsto & 0 \\
 & 011 & \mapsto & 1 \\
 & 100 & \mapsto & 0 \\
 & 101 & \mapsto & 1 \\
 & 110 & \mapsto & 1 \\
 & 111 & \mapsto & 0
\end{array}
\qquad
\begin{array}{lccc}
f_{150}: & 000 & \mapsto & 0 \\
 & 001 & \mapsto & 1 \\
 & 010 & \mapsto & 1 \\
 & 011 & \mapsto & 0 \\
 & 100 & \mapsto & 1 \\
 & 101 & \mapsto & 0 \\
 & 110 & \mapsto & 0 \\
 & 111 & \mapsto & 1
\end{array}
$$

We shall give theoretic result similar to the one obtained for linear rules using the following theorem, [26].

Theorem 6.3.9 *[26] The Jacobian matrices of two CAs of the same size, with the same boundary conditions but with different rules, are identical if and only if the rule of one of the CAs is the Boolean complement of that of the other CA.*

And then we get the immediate consequence:

Theorem 6.3.10 *The complement of linear CA is controllable via boundary actions iff*

$$Rank(\mathcal{C}_K) = T = N - 1$$

Simulation Examples

Example 6.3.11 *Consider the transition function corresponding to the rule* 105 *[94] which is the logical complement of rule* 150. *In a finite lattice of* $N = 70$, *One needs to find the sequence of controls* $\{u^0_{c_1}, \ldots, u^{T-1}_{c_1}\}$ *which leads the system in the region* $\omega = \{c_{10}, \ldots, c_{40}\}$ *from the initial configuration to a desired one* s_d *defined by:* $s_d(c_i) = 1 \forall \quad 10 \leq i \leq 40$ *at time* $T = 64$, *see Figure (.6.6).*

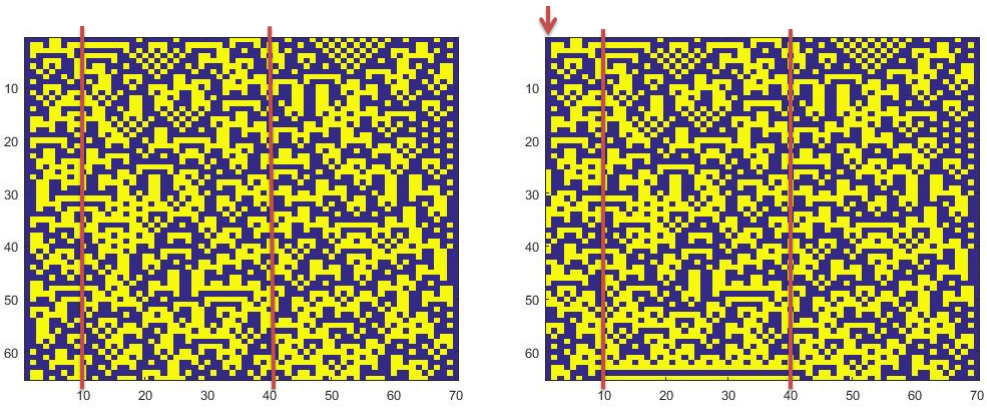


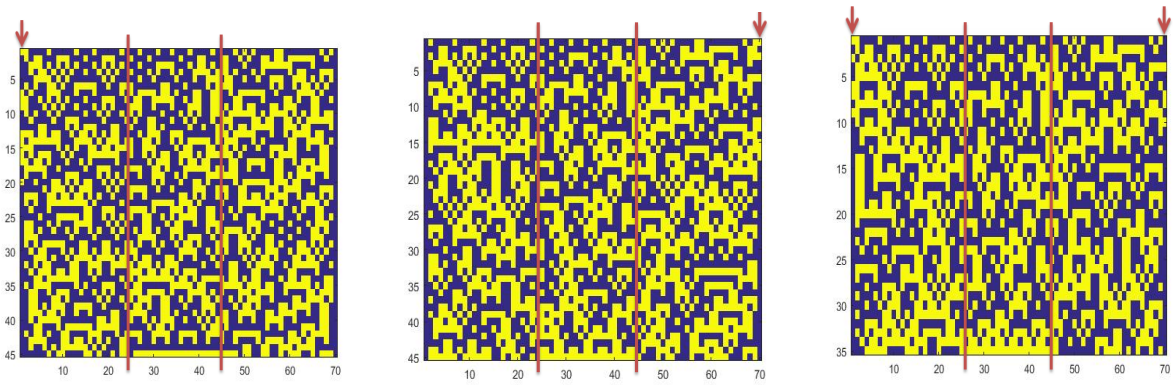
Figure .6.6 : Evolution of rule 105 in both cases: with and without control, the controlled region is $\omega = \{c_{10}, \ldots, c_{40}\}$

We are able to control the complement of double peripheral rules from two sides, the following example show that:

Example 6.3.12 *Let a CA rule 105 consists of* $N = 70$ *cells,* $\omega = \{c_{26}, \ldots, c_{45}\}$ *by applying the control from two sides we get the desired state in minimum time.*

6.3.7 Boundary Regional controllability of linear CA case 2:

We study the problem of boundary controllability via regional actions i.e, find the sequence of control $\{u^0_{c_{posmin}}, \ldots, u^0_{c_{posmax}}\}$, in order to achieve a desired state on one



Left c_1 T=44 **Right c_n T=44** **Two Sides T=34**

Figure .6.7 : Control of rule 105, the controlled region is $\omega = \{c_{26}, \ldots, c_{45}\}$

boundary of the lattice c_1 or c_n for a given time interval $[t_1, t_2]$, such as:

$$s_t(c_1) = s_d(c_1) \quad t = t_1, \ldots, t_2$$

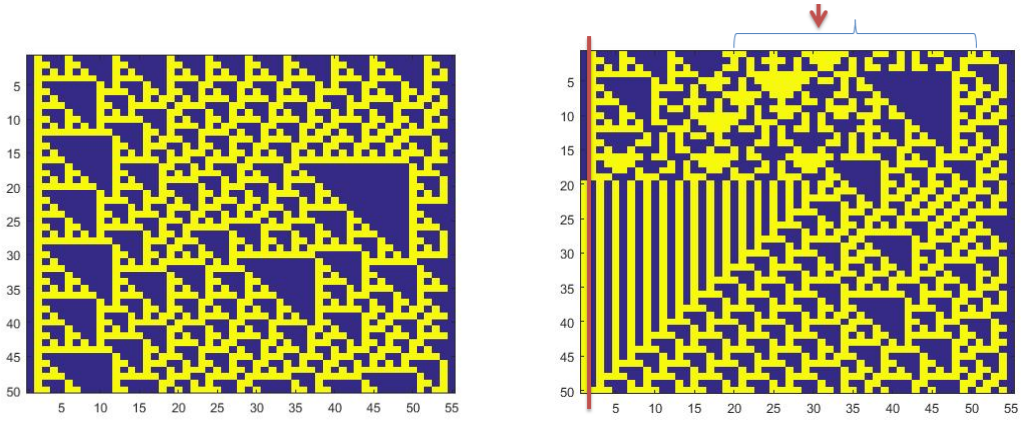
Results are illustrated in the following example:

Simulation Example

Example 6.3.13 *We consider 1D linear CA which is consisting of $N = 55$ cells noted c_i, $i = 1, \ldots, N$. Each cell can take a value from the set of states $S = \{0, 1\}$, the transition of the state of cell is done according to the neighbourhood $\mathcal{N}(c_i) = \{c_{i-1}, c_i, c_{i+1}\}$ where the rule is given by:*

$$s_{t+1}(c_i) = s_t(c_{i-1}) \oplus s_t(c_i)$$

One needs to achieve a desired state $s_d(c_1) = 1$ in the region $\Gamma = \{c_1\}$ at time $T = 19$ and wants to keep this value until a given time $T = 49$.
To achieve the desired state in the boundary we need to apply the control on the cells $\{c_{20}, \ldots, c_{50}\}$. The evolution of the rule is given before and after applying the control in the following figure:



Without applying the control **By applying the control**

Figure .6.8 : Boundary Regional Control of rule 60, $\omega = \{c_1\}$

By applying the control from two sides we get the desired state in minimum time.

6.4 Regional Controllability of Nonlinear CA: Generating the preimages

Counting the preimages of one dimensional CA has been already studied in several works by many authors as Jen in 1989 [66], Voorhees in 1993 [92] and others in [67, 81, 82] where they have used the Brujuin diagram.
In this section, we study the problem of regional controllability via boundary actions of nonlinear CA, by proposing a new algorithm to generate all preimages in order to numerically test them for controllability. One should check in principle the possibility of reaching the initial configuration starting from the desired configuration by searching the required control sequence $(u_l^0, u_l^1, \ldots, u_l^{T-1})$ and $(u_r^0, u_r^1, \ldots, u_r^{T-1})$ that generate the target configuration in the controlled region. Before going on, let us give some definitions:

Definition 36 *[26] The Hamming distance between two sequences of bits having the same length is the number of positions where the bit differ.*

Example 6.4.1 *The rules* 150 *and* 22 *their representations by Wolfram are:* 01101001 *and* 01101000*, in this case, the Hamming distance is equal to* 1*, this distance represents the degree of non-linearity.*

The classification of the rules of cellular automata according to the deviation from linearity is given in the following table [26]:

Classification of three-variable Boolean functions on the basis of deviation from linearity

Name of the class	Rules in the class	Number of rules in the class
CLASS 0	0, 60, 90, 102, 150, 170, 204, 240	8
CLASS 1	1, 2, 4, 8, 16, 22, 26, 28, 32, 38, 42, 44, 52, 56, 61, 62, 64, 70, 74, 76, 82, 88, 91, 94, 98, 100, 103, 110, 112, 118, 122, 124, 128, 134, 138, 140, 146, 148, 151, 158, 162, 168, 171, 174, 176, 182, 186, 188, 196, 200, 205, 206, 208, 214, 218, 220, 224, 230, 234, 236, 241, 242, 244, 248	64
CLASS 2	3, 5, 6, 9, 10, 12, 17, 18, 20, 23, 24, 27, 29, 30, 33, 34, 36, 39, 40, 43, 45, 46, 48, 53, 54, 57, 58, 63, 65, 66, 68, 71, 72, 75, 77, 78, 80, 83, 86, 89, 92, 95, 96, 99, 101, 106, 108, 111, 113, 114, 116, 119, 120, 123, 125, 126, 129, 130, 132, 135, 136, 139, 141, 142, 144, 147, 149, 154, 156, 159, 160, 163, 166, 169, 172, 175, 177, 178, 180, 183, 184, 187, 189, 190, 192, 197, 198, 201, 202, 207, 209, 210, 212, 215, 216, 219, 221, 222, 225, 226, 228, 231, 232, 235, 237, 238, 243, 245, 246, 249, 250, 252	112
CLASS 3	7, 11, 13, 14, 19, 21, 25, 31, 35, 37, 41, 47, 49, 50, 55, 59, 67, 69, 73, 79, 81, 84, 87, 93, 97, 104, 107, 109, 115, 117, 121, 127, 131, 133, 137, 143, 145, 152, 155, 157, 161, 164, 167, 173, 179, 181, 185, 191, 193, 194, 199, 203, 211, 213, 217, 223, 227, 229, 233, 239, 247, 251, 253, 254	64
CLASS 4	15, 51, 85, 105, 153, 165, 195, 255	8

Principle:

The idea consists of generating all preimages of the controlled region see Figure (.6.9).

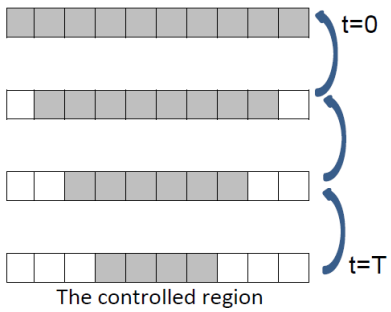


Figure .6.9 : Preimages

First, the procedure begins by generating the neighbourhood $s^{i-1}(c_{posmax-1}, c_{posmax}, c_{posmax+1})$ given the value $s^i(c_{posmax})$, looking in function f associated to the nonlinear rule (class 1 or 2), see Figure (.6.10):

$$\begin{array}{rccc} f\colon & 000 & \mapsto & b_0 \\ & 001 & \mapsto & b_1 \\ & 010 & \mapsto & b_2 \\ & 011 & \mapsto & b_3 \\ & 100 & \mapsto & b_4 \\ & 101 & \mapsto & b_5 \\ & 110 & \mapsto & b_6 \\ & 111 & \mapsto & b_7 \end{array}$$

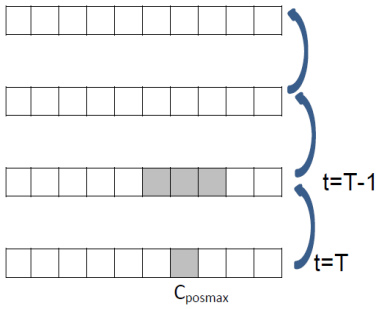


Figure .6.10 : Preimages of the neighbourhood of the boundary cell of the controlled region (right)

One continues generating the preimages that remain given the values of the target sites as in Figure (.6.11) $s^i(c_{posmax-1}), \ldots, s^i(c_{posmin})$ by using the following formula:

$$s^{i-1}(c_{j-1}) = mod(s^i(c_j) + a_2 \times s^{i-1}(c_j) + a_3 \times s^{i-1}(c_{j+1}), 2) \quad j = posmax-1, \ldots, posmin$$

Where a_2, a_3 are the coefficients linked to the linear rule (class 0). When the task is finished we do a test by using the function f:

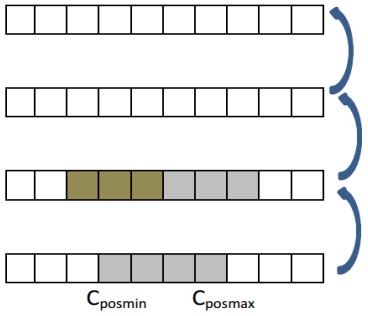


Figure .6.11 : Preimages of the neighbourhood of the boundary cell of the controlled region (left)

$F(s^{i-1}(c_{posmin-1},\ldots,c_{posmax})) = s^i(c_{posmin},\ldots,c_{posmax-1})$, if yes we go up and we continue the task until instant 0. If the previous condition is true, the CA is controllable, otherwise: if $F(s^{i-1}(c_{posmin-1},\ldots,c_{posmax})) \neq (c^i_{posmin},\ldots,c^i_{posmax-1})$, we choose another neighbourhood $s^{i-1}(c_{posmax-1}, c_{posmax}, c_{posmax+1})$ given the value $s^i(c_{posmax})$, looking the function f such as the chosen neighbourhood is different from the last one (for that we use matrix tabou to save all visited lines associated to the taken configurations).

Notation 5 *We introduce the following notations:*

1. *posmin: is the minimum position of the cell in the region* ω^i.
2. *posmax: is the maximum position of the cell in the region* ω^i.
3. *the vector result is defined as follows and is related to the CA rule:*

$$result = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{pmatrix}$$

4. *$mat(i,j) = s^i(c_j)$ is the state of cell c_j at time i.*

5. *allpossible: is a vector $(1 \times (time + 1))$. We save the number of appearance of state of cell c^i_{posmax}, for instance the rule 22:*

$$f_{22}: \begin{array}{ccc} 111 & \mapsto & 0 \\ 110 & \mapsto & 0 \\ 101 & \mapsto & 0 \\ 100 & \mapsto & 1 \\ 011 & \mapsto & 0 \\ 010 & \mapsto & 1 \\ 001 & \mapsto & 1 \\ 000 & \mapsto & 0 \end{array}$$

if $s^i(c_{posmax}) = 1$ then $allpossible(i-1) = 3$.

6. *We denote by ω^i the region at time $i = 0, \ldots, T$ (the blue region) as it is shown in the following Figure (.6.12):*

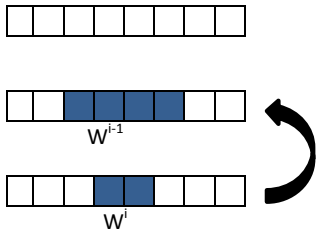


Figure .6.12 : Preimages of a controlled region

Remark 10 *The algorithm works also relevant to the class 3 and 4 which are the complement of class 1 and 0 according to the table, but it only works just for small sized controlled region.*

Algorithm: *Initialization*

Enter CA size and time, then the position min and the position max of the controlled region.

Define two vectors posmin and posmax of size (time+1).

posmax(time+1) ← positionmax; posmin(time+1) ← positionmin;
for $j \leftarrow time$ **to** 2 **do**
  if $posmax(j+1) < n$ **then**
    posmax(j) ← posmax(j+1)+1;
  else
    posmax(j) ← posmax(j+1);
  end if
  if $posmin(j+1) > 1$ **then**
    posmin(j) ← posmmin(j+1)-1;
  else
    posmin(j) ← posmin(j+1);
  end if
end for
for $j \leftarrow 2$ **to** $time$ **do**
  if $posmax(j) = n$ **then**
    posmax(j) ← posmax(j)-1;
  end if
  if $posmin(j) = 1$ **then**
    posmin(j) ← posmmin(j)+1;
  end if
end for
Define a vector of dimension (time-1) called arbre.
arbre(1) ← 1;
for $i \leftarrow 2$ **to** $(time-1)$ **do**
  arbre(i) ← $arbre(i-1) \times 8$;
end for
sumarbre ← 0;
for $i \leftarrow 1$ **to** $(time-1)$ **do**
  sumarbre ← sumarbre+arbre(i);
end for
$size \leftarrow sumarbre \times 8$;
Enter the Wolfram rule of classe 2 or 3.

Enter the coefficients linked to the linear rule a_1, a_2, a_3.
Enter the values of the initial configuration except for the boundaries c_1, c_n and the desired state.
Define the matrix tabou of size $(time + 1) \times size$.
Initialize two vectors of size (time+1) called caseprise and allpossible by zeros.

Algorithm: *Main Loop*

indice $\leftarrow 1$; i $\leftarrow time + 1$;
address 1:
while $i > 1$ **do**
  if $i \neq 2$ **then**
    for $j \leftarrow posmax(i)$ **to** $posmin(i)$ **do**
      if $(j = posmax(i))$ **then**
        allpossible(i-1) $\leftarrow 0$;
        for $k \leftarrow 1$ **to** 8 **do**
          if $mat(i, j) = result(k, 1)$ **then**
            allpossible(i-1) $\leftarrow$ allpossible(i-1)+1;
          end if
        end for
        $su \leftarrow 0$;
        for $r \leftarrow 1$ **to** 8 **do**
          if $su = 0$ **then**
            if $mat(i, j) = result(r, 1)$ **then**
              $lol \leftarrow 0$;
              for $l \leftarrow 1$ **to** $indice$ **do**
                if $tabou(i - 1, l) = r$ **then**
                  $lol \leftarrow lol + 1$;
                end if
              end for
              if $lol = 0$ **then**
                fill the states of cells at time $i - 1$ such as:
                $(s^{i-1}(c_{posmax-1}), s^{i-1}(c_{posmax}), s^{i-1}(c_{posmax+1})) = s^i(c_{posmax})$

$su \leftarrow 1$;
$tabou(i-1, indice) \leftarrow r$;
$caseprise(i-1) \leftarrow caseprise(i-1) + 1$;
$indice \leftarrow indice + 1$;
end if
end if
end if
end for
end if
if $j \neq posmax(i)$ **then**
calculate the states of cells which remain by using the following formula: $mat(i-1, j-1) \leftarrow mod(mat(i,j) + a_2 \times mat(i-1,j) + a_3 \times mat(i-1, j+1), 2)$
end if
end for
if $F(s(\omega_{i-1})) = s(\omega_i)$ **then**
$i \leftarrow i-1$; goto address 1;
else
if all the lines in vector result are visited **then**
if $(i = time + 1)$ **then**
go out from the main loop, it's not regional controllable
else
if there is an equality between each box of the vectors allpossible and caseprise from (i-1) to time **then**
go out from the main loop, it is not controllable
else
for l←1 **to** indice **do**
tabou(i-1,l)←0;
end for
caseprise(i-1) ← 0; $i \leftarrow i+1$;
goto address 1;
end if

end if
else
goto address 1;
end if
end if
end if
if $(i = 2)$ **then**
fill the states of the boundaries of CA in $T = 0$ i.e, $s^0(c_1), s^0(c_N)$ using the table of the transition rule f.
if $F(s(\omega_{i-1}) = \omega_i)$ **then**
write('it is boundary regional controllable ');
go out from the main loop
else
if we have the equality between each box for the two vectors caseprise and allpossible **then**
write('it is not boundary regional controllable ');
go out from the main loop
else
did $\leftarrow 0$;
for m $\leftarrow 2$ **to** (time) **do**
if did =0 **then**
if $caseprise(m) \neq allpossible(m)$ **then**
$i \leftarrow m$;
$did \leftarrow m$;
end if
end if
end for
for $lili \leftarrow 1$ **to** (did-1) **do**
caseprise(lili) $\leftarrow 0$;
for $mimi \leftarrow 1$ **to** indice **do**
tabou(lili,mimi) $\leftarrow 0$;
end for

```
            end for
            goto address 1;
          end if
        end if
      end if
    end while
```

6.5 Conclusions

Controlling complex systems that naturally appear in numerous fields, including fluid mechanics, biology, population dynamics, medicine, constitutes a very challenging problem that has drawn an increasing interest during the last decade. Cellular Automata are spatially extended systems that are particularly suitable for modelling and simulating various complex and highly non-linear systems.

In this chapter, we studied one of the most prominent problems related to control theory using CA: the problem of boundary regional controllability. We showed how can we reach a desired state on a region by acting on the boundary of one-dimensional linear CA rules and proved the controllability via boundary actions of this type of rules by using the well known Kalman condition.
We investigated also the case of the logical complement of these rules by using the same algorithm. The results presented in this chapter are related to linear 1-D deterministic CA based on a proposed algorithm and for nonlinear ECA among the 256 rules, by exploring more suitable methods. A first backtracking algorithm based on a method for generating all pre-images of a target configuration on the sub-region was successfully tested only for small regions and limited time span.
Many phenomena can be described using probabilistic cellular automata. It may worth to deal with studying the regional controllability problem of PCA. In the following chapter, we study the regional controlability of probabilistic CA using Markov Chains approach.

Optimal and Suboptimal Regional Control of Probabilistic Cellular Automata

CHAPTER 7

REGIONAL CONTROL OF PROBABILISTIC CELLULAR AUTOMATA

Résumé:

Dans ce chapitre [1], On a abordé le problème de la contrôlabilité régionale des automates cellulaires probabilistes qui sont des systèmes stochastiques étendus, largement utilisés pour la modélisation des phénomènes dans de nombreuses disciplines. La possibilité de contrôler leur comportement est donc un sujet important. Nous utilisons la même approche celle des chaînes de Markov basée sur la matrice de transition. En particulier, nous nous intéressons au contrôle optimal, qui est assez exigeant en termes de calcul, et nous présentons donc aussi une méthode moins exigeante et sous-optimale. Enfin, nous présentons un exemple de contrôle optimal avec contraintes.

[1] Cette partie a fait l'objet des travaux suivants:
F. Bagnoli, S. Dridi, S. El Yacoubi, R. Rechtman (2018)
Regional control of probabilistic cellular automata, International Conference on Cellular Automata, Springer cham.
F. Bagnoli, S. Dridi, S. El Yacoubi, R. Rechtman (2019)
Optimal and Sub-optimal Regional Control of Probabilistic Cellular Automata. Journal of Natural computing.

7.1 Introduction

Cellular Automata (CA) are widely used tool for modelling physical systems [30, 33, 54, 72]. They come in two major "flavours": deterministic CA (DCA) [70, 90, 95] and probabilistic CA (PCA) [34, 79].

Probabilistic cellular automata can be thought as an extension of DCA where the transition function gives the probability that the target node goes in a certain state. If all these probabilities are either zero or one, that the PCA reduces to a DCA. In both cases, the state of the CA is the collection of states at the nodes of the graph and this state changes in time according to functions defined in every node of the graph.

In this chapter we study the problem of regional controllability of probabilistic cellular automata using the Markov chains approach.

7.2 Probabilistic Cellular Automata

Probabilistic CA constitute an extension of DCA. Let us introduce the transition probability $\tau(1|\boldsymbol{\mathcal{N}})$ that, given a certain configuration $\boldsymbol{\mathcal{N}} = \boldsymbol{\mathcal{N}}_i$ of the neighbourhood of site i, gives the probability of observing $s'_i = 1$ at next time step. Clearly $\tau(0|\boldsymbol{\mathcal{N}}) = 1-\tau(1|\boldsymbol{\mathcal{N}})$. DCA are such that $\tau(1|\boldsymbol{\mathcal{N}})$ is either 0 or 1, while for PCA it can take any value in the middle. For a PCA with k inputs, there are 2^k independent transition probabilities, and for totalistic PCA there are $k+1$ independent probabilities. If one associate each transition probability to a different axis, the space of all possible PCA is an unit hypercube, with corners corresponding to DCA.

PCA can be also *partially* deterministic, i.e., the transition probability $\tau(1|\boldsymbol{\mathcal{N}})$ can be zero or one for certain $\boldsymbol{\mathcal{N}}$. This opens the possibility for the automata to have one or more absorbing state, i.e., configurations that always originate the same configuration (or give origin to a cyclic behaviour). The BBR model illustrated below has two absorbing states.

The evolution of all possible configurations $\boldsymbol{s}$ of a PCA can be written as a Markov chain. Let us define the probability $P(\boldsymbol{s}, t)$, i.e., the probability of observing

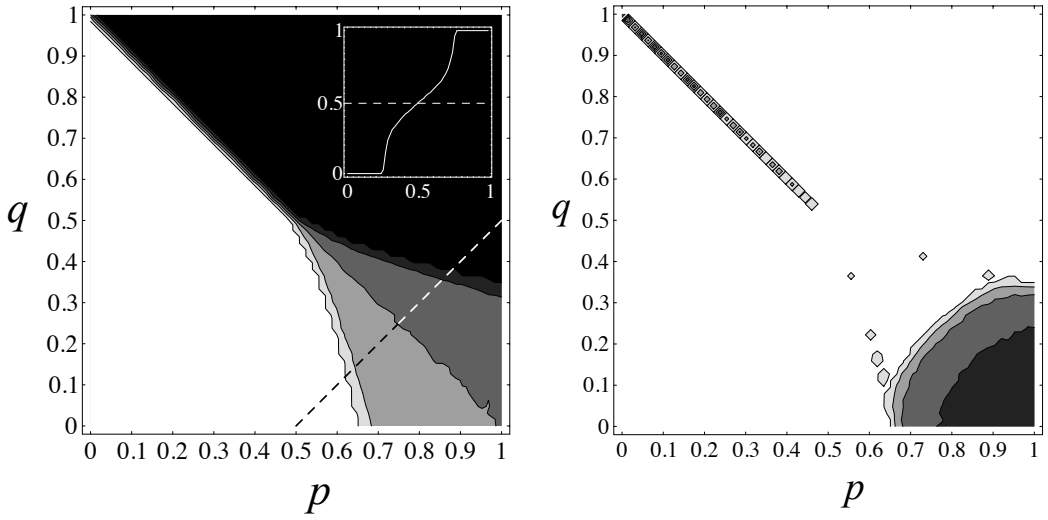


Figure .7.1 : Phase diagram of the BBR model. Left: Density phase diagram. Right: Damage phase diagram.

the configuration $\boldsymbol{s}$ at time t. Its evolution is given by

$$P(\boldsymbol{s}, t+1) = \sum_{\boldsymbol{y}} M(\boldsymbol{s}|\boldsymbol{y}) P(\boldsymbol{y}, t), \tag{7.1}$$

where the matrix M is such that

$$M(\boldsymbol{s}|\boldsymbol{y}) = \prod_{i=1}^{N} \tau\left(s_i|\boldsymbol{\mathcal{N}}_i(\boldsymbol{y})\right). \tag{7.2}$$

For a CA on a 1D lattice and $k = 3$ we have

$$M(\boldsymbol{s}|\boldsymbol{y}) = \prod_{i=1}^{N} \tau(s_i|y_{i-1}, y_i, y_{i+1}). \tag{7.3}$$

Phase transitions for PCA can be described as degeneration of eigenvalues in the limit $N \to \infty$ and (subsequently) $T \to \infty$ [56].

Notice that since DCA are limit cases of PCA, they also can be seen as particular Markov chains.

A Markov chain such that, for some t, $(M^t)_{ij} > 0$ for all i, j is said to be regular, and this implies that any configuration can be reached by any configuration in time t.

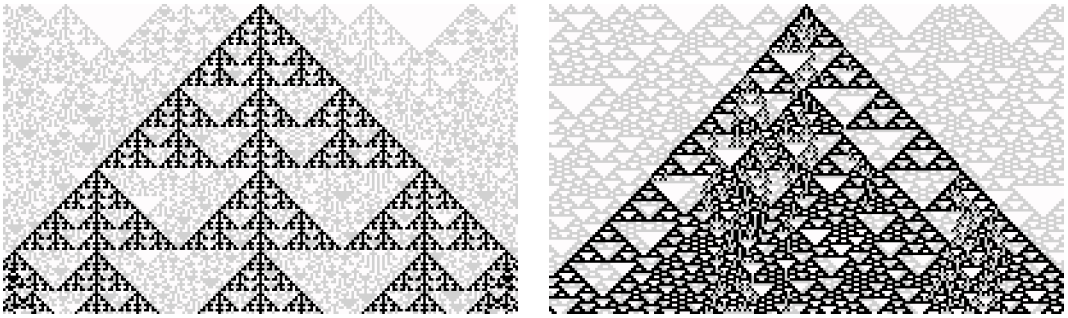


Figure .7.2 : Damage spreading; time runs downwards. Left: CA Rule 150. Right: CA rule 126.

A weaker condition (ergodicity) says that t may depend on the pair i, j (for instance, one may have an oscillating behaviour such that certain pairs can be connected only for even or odd values of t). Also for ergodic systems all configurations are connected.

7.3 The model

We shall use as a testbed model the one presented in [9], which is an extension of the Domany-Kinzel CA [34]. We shall refer to it as the BBR model from the name of its authors. It is a totalistic PCA defined on a one-dimensional lattice, with connectivity $k = 3$. The transition probabilities of the model are

$$\tau(1|0) = 0; \qquad \tau(1|1) = p; \qquad \tau(1|2) = q; \qquad \tau(1|3) = 1. \tag{7.4}$$

This model has two absorbing states, corresponding to configurations $\mathbf{0} = (0, 0, 0, \dots)$ and $\mathbf{1} = (1, 1, 1, \dots)$. Notice that for $p = 1$ and $q = 0$ we have DCA rule 150.

The implementation of a stochastic model makes use of one or more random numbers. For instance, the BBR model can be implemented using the function

$$\begin{aligned} s'_i = f(s_{i-1}, s_i, s_{i+1}; r_i) &= [r_i < p](s_{i-1} \oplus s_i \oplus s_{i+1} \oplus s_{i-1}s_is_{i+1}) \\ &\oplus [r_i < q](s_{i-1}s_i \oplus s_{i-1}s_{i+1} \oplus s_is_{i+1} \oplus s_{i-1}s_is_{i+1}) \\ &\oplus s_{i-1}s_is_{i+1}, \end{aligned} \tag{7.5}$$

where $[\cdot]$ is the truth function which takes value one if $\cdot$ is true and zero otherwise, and $\oplus$ is the sum modulo two. The $r_i = r_i(t)$ random numbers have to be extracted

for each site and for each time. One can think of extracting them once and for all at the beginning of the simulation, i.e., running the simulation on a space-time lattice on which a random field $r_i(t)$, $i = 1, \ldots, N$; $t = 0, \ldots$ is defined. Notice that in this way one has a deterministic CA over a quenched random field, i.e., the automata is now deterministic, since the random numbers r_i are now a property of the lattice.

This is similar to the implementation of the Kauffman model [32], in which one has different deterministic functions in different sites, chosen at random at beginning. Indeed, this transformation of a probabilistic problem into a deterministic one in the presence of a quenched random field has allowed the porting of concepts related to deterministic CA, like the idea of damage spreading (the evolution of an initially small difference between two replicas) to PCA [80].

The phase diagram of the BBR model is reported in Figure (7.1)-left. One can see three regions. The one marked in white, for $p < 0.65$, is where the only asymptotically stable configuration is the absorbing state formed by all zeros, i.e., the asymptotic probability distribution of configurations $P(\boldsymbol{s})$ is a delta on zero. The symmetric region marked in black, for $q > 0.35$ is where the only stable configuration is formed by all ones. Actually, in a region near the diagonal $q = 1 - p$, for $p < 0.5$ the two absorbing states are both stable, the transition line is fixed by the initial configuration, which in the figure is drawn at random with the same probability of extracting a zero and a one. These regions are denoted with the term "quiescent". The region marked in shades of grey, for $p > 0.65$ and $q < 0.35$ is a region where the two absorbing states are unstable, and the asymptotic probability distribution is distributed over many configurations, with average number of ones proportional to the shades of grey. In the inset it is reported the asymptotic average number of ones (the "density") computed along the dashed lines. This region is denoted with the term "active".

7.3.1 Damage spreading

One possibility for controlling the evolution of a system with little efforts is offered by the sensitive dependence on initial conditions, i.e., when a small variation in the initial state propagates to the whole system. Indeed, this is also the main ingredient of chaos, which in general prevents a careful control. But in discrete systems the

situation is somehow different. These systems are not affected by infinitesimal perturbations in the variables (assuming that they can be extended in the continuous sense), only to finite ones. The study of the propagation of a finite perturbation in CA goes under the name of "damage spreading", indicating how an initial disturbance (a "defect" or "damage") can spread in the system. A CA where a damage typically spreads is said to be chaotic.

Mathematically, one has two copies of the same system, say s and y, evolving with the same rule but starting from different initial conditions. We shall indicate with $z_i = s_i \oplus y_i$ the local difference at site i. Typical patterns of the spreading of a damage (i.e., the evolution of z) are reported in Figure (.7.2).

For PCA, the concept of damage spreading is meant "given the random field". The phase diagram of the damage z for the BBR model is shown in Figure (.7.1)-right.

7.4 Reachability problem

We shall mainly deal here with the problem of regional control via boundary actions, i.e., boundary reachability as illustrated in Figure (.7.3), however the techniques of analysis can be extended to other cases.

Let us now consider the problem of computing the probability $C_{\boldsymbol{sy}}(a,b) = C(\boldsymbol{s}|\boldsymbol{y};a,b)$ which is the probability of getting configuration s at time $t+1$ given the configuration y at time t, and boundaries a and b (for simplicity we refer here only to one-dimensional cases). The Markov control matrix $C(a,b)$ is given by

$$C_{\boldsymbol{sy}}(a,b) = \tau(s_1|a,y_1,y_2)\tau(s_2|y_1,y_2,y_3)\ldots\tau(s_n|y_{n-1},y_n,b),$$

where n indicates the size of the target region.

For a given control sequence $\boldsymbol{a} = a_1,\ldots,a_T$ and $\boldsymbol{b} = b_1,\ldots,b_T$, the resulting Markov control matrix for time T is

$$C(\boldsymbol{a},\boldsymbol{b}) = \prod_{t=1}^{T} C(a_t,b_t).$$

We can define several control problems. A first one is about ergodicity: which is the best control sequence $\boldsymbol{a}$ and $\boldsymbol{b}$ so that $C_{\boldsymbol{sy}}(\boldsymbol{a},\boldsymbol{b}) > 0$ for all pairs $\boldsymbol{s},\boldsymbol{y}$ and

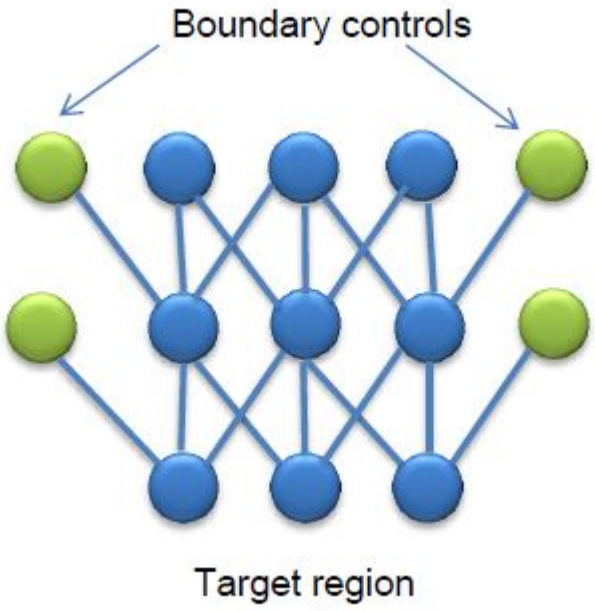


Figure .7.3 : Boundary value problem

minimum time T? Another is: given a certain time T and a pair $\boldsymbol{s}, \boldsymbol{y}$, which is the best control sequence $\boldsymbol{a}$ and $\boldsymbol{b}$ that maximises $M_{\boldsymbol{sy}}(\boldsymbol{a}, \boldsymbol{b}) > 0$?

Clearly, one can also be interested in avoiding certain configurations, for instance, if $s_i = 1$ represents the presence of some animal or plant in position i at time t, one could be interested in devising a control that prevents the extinction of animals, i.e., avoid the state $\boldsymbol{s} = 0$.

As we shall show in the following, so far we have not found algorithms for finding the best control but exhaustive search.

Beyond finding the actual sequence that maximises the observable, one could be rather interested in determining the *existence* of such a sequence, for a certain time interval T, or to find the minimum time T for which an optimal sequence exists.

In particular this latter problem can be faced with less computer efforts than

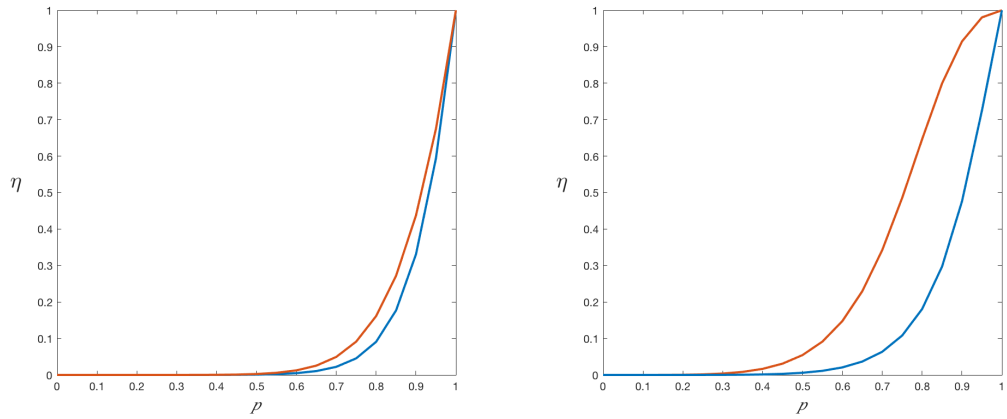


Figure .7.4 : The ratio $\eta = \min(Q)/\max(Q)$ for the BBR model with $n = 5$ for $T = 3$ (lower, blue curve) and $T = 5$ (upper, red curve). Left: $q = 0$, Right: $q = 1-p$.

finding the actual sequence for the best control. If one considers the matrix

$$Q = \frac{1}{4}\sum_{a,b} C(a,b) = \frac{1}{4}\big(C(0,0) + C(0,1) + C(1,0) + C(1,1)\big),$$

and then computes its power Q^T, all possible control sequences of length T are contained in such a power. Therefore, the problem of the existence of a control sequence for a given time T reduces to checking if $(Q^T)_{\boldsymbol{sy}} > 0$. One can also quantify the effective of the control by computing the ratio η between the minimum and maximum values of C. If this ratio is zero, it means that there are certain pairs of configurations that cannot be connected by any control sequence, while $\eta = 1$ means that all pairs of configurations can be connected with equal easiness.

Let us illustrate some of these concepts for the BBR model, for $p = q$ and for $q = 0$. In Figure (.7.4) we show the easiness parameter η in function of p for $q = 0$ and $q = 1 - p$, for $n = 5$ and different values of T. One can see that in the "quiescent" phase $p < 0.5$ the control is almost impossible, and that on the line $q = 1 - p$, for $p > 0.5$, the easiness of the control increases with T faster that on the line $q = 0$. Indeed, referring to Figure (.7.1), one can see that this portion of the diagram corresponds to the "active" phase, where the BBR model is ergodic. One can also notice that the easiness of the control is not related to the damage spreading

phase: considering for instance the line $q = p$, from Figure (7.1)-right one sees that the damage spreading phase starts for $p > 0.75$, while from Figure (7.4)-right one sees that the control is possible well before this threshold. The control properties are probably associated to the "chaoticity" of the associated deterministic CA over the random quenched field, a problem which will be faced in the future (for "chaotic" CA and the associated Boolean derivatives, see [8, 90].

7.4.1 Optimal control

Let us now turn to the problem of finding the best control. For compactness, let us consider the case $n = 3$, for which the minimum control time is $T = 2$. The highest probability for each pair of configurations $\boldsymbol{s}$ (row index in base two) and $\boldsymbol{y}$ (column index in base two) for $q = 1 - p$ and $p = 0.7$ is

$$C_{\max}(0.7, 0.3) = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 1.000 & 0.262 & 0.213 & 0.396 & 0.262 & 0.396 & 0.396 & 0.240 \\ 0.700 & 0.278 & 0.208 & 0.293 & 0.208 & 0.293 & 0.293 & 0.343 \\ 0.343 & 0.221 & 0.221 & 0.253 & 0.221 & 0.195 & 0.253 & 0.490 \\ 0.343 & 0.293 & 0.293 & 0.278 & 0.293 & 0.208 & 0.208 & 0.700 \\ 0.700 & 0.208 & 0.208 & 0.293 & 0.278 & 0.293 & 0.293 & 0.343 \\ 0.490 & 0.253 & 0.195 & 0.221 & 0.253 & 0.221 & 0.221 & 0.343 \\ 0.343 & 0.293 & 0.293 & 0.208 & 0.293 & 0.208 & 0.278 & 0.700 \\ 0.240 & 0.396 & 0.396 & 0.262 & 0.396 & 0.213 & 0.262 & 1.000 \end{pmatrix} \end{array},$$

corresponding to controls $\boldsymbol{a}$ and $\boldsymbol{b}$ (again in base two)

$$\boldsymbol{a} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & 3 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 & 3 & 0 & 0 & 3 \\ 0 & 3 & 3 & 0 & 2 & 1 & 1 & 2 \\ 2 & 2 & 1 & 2 & 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 3 & 3 & 2 & 2 & 3 \end{pmatrix} \end{array}, \qquad \boldsymbol{b} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 2 \\ 0 & 2 & 3 & 1 & 3 & 1 & 0 & 2 \\ 3 & 1 & 0 & 1 & 0 & 2 & 3 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \\ 2 & 0 & 2 & 3 & 2 & 3 & 2 & 3 \\ 1 & 3 & 2 & 0 & 2 & 0 & 1 & 3 \\ 2 & 3 & 2 & 2 & 2 & 2 & 3 & 3 \end{pmatrix} \end{array}.$$

These results should be read in this way. Let us consider for instance the initial configuration $\boldsymbol{y} = 3 = 110|_2$ (numbers are coded in reverse order) and final configuration $\boldsymbol{s} = 4 = 001|_2$. The best control is given by a sequence $\boldsymbol{a} = 0 = 00|_2$ and $\boldsymbol{b} = 3 = 11|_2$, which is reasonable since one is trying to force zeros on the left side of the configurations and ones on the right side.

Notice however that the entries for $\boldsymbol{a}$ and $\boldsymbol{b}$ are not always either 0 or 3, meaning that the best control is not a uniform one for all pairs. For instance, for going from

$\boldsymbol{y} = 3 = 110|_2$ to $\boldsymbol{s} = 1 = 100|_2$ one has to apply $\boldsymbol{a} = 2 = 01|_2$ and $\boldsymbol{b} = 1 = 10|_2$, exploiting the fact that $q = \tau(1|2) = 1 - p = 0.3$ and therefore for forcing a zero in the presence of a neighbourhood already containing a one, it is better to insert another one than a zero.

Another example is the following, again for $L = 3$ and $T = 2$ but for $p = 0.8$ and $q = 0$,

$$C_{\max}(0.8, 0) = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 1.000 & 0.656 & 0.656 & 1.000 & 0.656 & 1.000 & 1.000 & 1.000 \\ 0.800 & 0.646 & 0.646 & 0.800 & 0.646 & 0.800 & 0.800 & 1.000 \\ 0.640 & 0.640 & 0.640 & 0.800 & 0.640 & 0.640 & 0.800 & 1.000 \\ 0.512 & 0.516 & 0.516 & 0.640 & 0.516 & 0.512 & 0.640 & 1.000 \\ 0.800 & 0.646 & 0.646 & 0.800 & 0.646 & 0.800 & 0.800 & 1.000 \\ 0.640 & 0.512 & 0.512 & 0.640 & 0.512 & 0.640 & 0.640 & 0.800 \\ 0.512 & 0.516 & 0.516 & 0.640 & 0.516 & 0.512 & 0.640 & 1.000 \\ 0.410 & 0.512 & 0.512 & 0.512 & 0.512 & 0.410 & 0.512 & 1.000 \end{pmatrix} \end{array},$$

corresponding to controls

$$\boldsymbol{a} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 3 & 3 & 3 & 2 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 & 3 & 0 & 3 & 3 \\ 0 & 3 & 3 & 0 & 2 & 1 & 1 & 2 \\ 2 & 1 & 1 & 2 & 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 3 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 & 3 & 0 & 3 & 3 \end{pmatrix} \end{array}, \qquad \boldsymbol{b} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 2 & 3 & 1 & 3 & 1 & 0 & 2 \\ 0 & 2 & 3 & 1 & 3 & 1 & 0 & 2 \\ 3 & 2 & 3 & 1 & 3 & 2 & 3 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 3 & 1 \\ 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \\ 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \\ 1 & 3 & 2 & 3 & 2 & 0 & 1 & 3 \\ 2 & 3 & 2 & 3 & 2 & 3 & 1 & 3 \end{pmatrix} \end{array}.$$

7.4.2 Faster suboptimal control

The problem is that these optimal controls have been obtained by means of the scan of all possible sequences. For a one-dimensional CA with two free boundaries one has 4 possible control, and the exhaustive search implies the check of 4^T combinations, where T is the estimated control time, proportional to L.

An approximation to optimal control can be obtained by means of the Q matrix, which corresponds to the superposition of all possible controls. The idea is that of computing the quantity $Q^{T-1}(\boldsymbol{s}|\boldsymbol{y})$ which gives the possibility of getting $\boldsymbol{s}$ starting from configuration $\boldsymbol{y}$ in T time steps, for some control, and then search for the best control at the final time step$T$, i.e., maximise

$$C(a_T, b_T)Q^{T-1}$$

for each pair $(\boldsymbol{s}, \boldsymbol{y})$ over a_T and b_T, obtaining $\tilde{a}_T$ and $\tilde{b}_T$.

The procedure is then repeated for

$$C(\tilde{a}_T, \tilde{b}_T)C(a_{T-1}, b_{T-1})Q^{T-2}$$

and so forth. This suboptimal algorithm implies $4T$ steps. The results corresponding to the previous examples are, for $p = 0.7$ and $q = 1 - p$

$$\tilde{C}_{\max}(0.7, 0.3) = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 1.000 & 0.262 & 0.213 & 0.396 & 0.262 & 0.396 & 0.396 & 0.240 \\ 0.700 & 0.278 & 0.208 & 0.293 & 0.197 & 0.293 & 0.293 & 0.343 \\ 0.343 & 0.221 & 0.221 & 0.253 & 0.221 & 0.195 & 0.253 & 0.490 \\ 0.343 & 0.293 & 0.293 & 0.278 & 0.293 & 0.208 & 0.197 & 0.700 \\ 0.700 & 0.197 & 0.208 & 0.293 & 0.278 & 0.293 & 0.293 & 0.343 \\ 0.490 & 0.253 & 0.195 & 0.221 & 0.253 & 0.221 & 0.221 & 0.343 \\ 0.343 & 0.293 & 0.293 & 0.197 & 0.293 & 0.208 & 0.278 & 0.700 \\ 0.240 & 0.396 & 0.396 & 0.262 & 0.396 & 0.213 & 0.262 & 1.000 \end{pmatrix} \end{array},$$

with controls

$$\tilde{\boldsymbol{a}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 \\ 0 & 0 & 0 & 3 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 1 & 3 & 0 & 2 & 3 \\ 0 & 1 & 3 & 0 & 2 & 1 & 1 & 2 \\ 2 & 2 & 1 & 2 & 0 & 3 & 3 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 3 & 3 & 2 & 2 & 3 \end{pmatrix} \end{array}, \quad \tilde{\boldsymbol{b}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 2 \\ 0 & 2 & 3 & 1 & 1 & 1 & 0 & 2 \\ 3 & 1 & 0 & 1 & 0 & 2 & 3 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 2 & 2 & 2 & 3 & 2 & 3 & 2 & 3 \\ 2 & 0 & 2 & 3 & 2 & 3 & 2 & 3 \\ 1 & 3 & 2 & 2 & 2 & 0 & 1 & 3 \\ 2 & 3 & 2 & 2 & 2 & 2 & 3 & 3 \end{pmatrix} \end{array},$$

and, for $p = 0.8$ and $q = 0$

$$\tilde{C}_{\max}(0.8, 0) = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 1.000 & 0.656 & 0.656 & 1.000 & 0.656 & 1.000 & 1.000 & 1.000 \\ 0.800 & 0.646 & 0.646 & 0.800 & 0.646 & 0.800 & 0.800 & 1.000 \\ 0.640 & 0.640 & 0.640 & 0.800 & 0.640 & 0.640 & 0.800 & 1.000 \\ 0.512 & 0.512 & 0.512 & 0.640 & 0.512 & 0.512 & 0.640 & 1.000 \\ 0.800 & 0.646 & 0.646 & 0.800 & 0.646 & 0.800 & 0.800 & 1.000 \\ 0.640 & 0.512 & 0.512 & 0.640 & 0.512 & 0.640 & 0.640 & 0.800 \\ 0.512 & 0.512 & 0.512 & 0.640 & 0.512 & 0.512 & 0.640 & 1.000 \\ 0.410 & 0.410 & 0.410 & 0.512 & 0.410 & 0.410 & 0.512 & 1.000 \end{pmatrix} \end{array}$$

with controls

$$\tilde{\boldsymbol{a}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 3 & 3 & 3 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 3 & 3 \\ 0 & 3 & 3 & 0 & 2 & 1 & 1 & 2 \\ 2 & 1 & 1 & 2 & 0 & 3 & 3 & 0 \\ 0 & 3 & 3 & 3 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 3 & 3 \end{pmatrix} \end{array}, \quad \tilde{\boldsymbol{b}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 2 & 3 & 1 & 3 & 1 & 0 & 2 \\ 0 & 2 & 3 & 1 & 3 & 1 & 0 & 2 \\ 3 & 2 & 3 & 1 & 3 & 2 & 3 & 1 \\ 0 & 2 & 3 & 1 & 3 & 1 & 3 & 1 \\ 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \\ 2 & 3 & 2 & 3 & 2 & 3 & 2 & 3 \\ 1 & 0 & 1 & 3 & 1 & 0 & 1 & 3 \\ 2 & 0 & 1 & 3 & 1 & 3 & 1 & 3 \end{pmatrix} \end{array}.$$

As marked by boxes, some non-optimal entries are present.

7.4.3 Avoiding configurations

As anticipated, one can be also interested in driving a system towards a target but trying to avoid some states. For instance, in case of a population of organisms, one could be interested in avoiding their extinction, i.e., state $\mathbf{0}$. In this case the quantity to be maximised is

$$C(\boldsymbol{a}, \boldsymbol{b}) = \prod_{t=1}^{T} [C^*(a_t, b_t) \odot \overline{Z}(a_t, b_t)]$$

where C^* is equal to C but in column zero ($C^*_{\boldsymbol{s},\mathbf{0}} = 0$, i.e., the configuration $\mathbf{0}$ has been made absorbing, regardless of the controls), and $\overline{Z}_{\boldsymbol{s},\boldsymbol{y}} = 1 - C_{\mathbf{0},\boldsymbol{y}}$, i.e., is the probability of not jumping from $\boldsymbol{y}$ to $\mathbf{0}$. The $\odot$ symbol denoted the Hadamard (component-by-component) product.

In this way, we are maximising the probability of going from a configuration $\boldsymbol{s}$ to a configuration $\boldsymbol{y}$ in T steps, not passing from configuration $\mathbf{0}$.

Let us consider as an example the case $p = 0.7$ and $q = 1 - p$,

$$C^*{}_{\max}(0.7, 0.3) = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.142 & 0.234 & 0.178 & 0.165 & 0.178 & 0.112 & 0.143 & 0.343 \\ 0.190 & 0.210 & 0.210 & 0.201 & 0.210 & 0.136 & 0.201 & 0.490 \\ 0.218 & 0.278 & 0.278 & 0.255 & 0.278 & 0.163 & 0.192 & 0.700 \\ 0.142 & 0.178 & 0.178 & 0.143 & 0.234 & 0.112 & 0.165 & 0.343 \\ 0.143 & 0.230 & 0.179 & 0.169 & 0.230 & 0.122 & 0.169 & 0.343 \\ 0.218 & 0.278 & 0.278 & 0.192 & 0.278 & 0.163 & 0.255 & 0.700 \\ 0.164 & 0.382 & 0.382 & 0.256 & 0.382 & 0.193 & 0.256 & 1.000 \end{pmatrix} \end{array},$$

with controls

$$\boldsymbol{a}^* = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 3 & 0 & 0 & 3 \\ 0 & 0 & 0 & 3 & 1 & 0 & 1 & 1 \\ 1 & 2 & 2 & 1 & 3 & 0 & 2 & 3 \\ 3 & 3 & 3 & 3 & 2 & 2 & 2 & 2 \\ 1 & 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 3 & 3 & 2 & 2 & 3 \end{pmatrix} \end{array}, \quad \boldsymbol{b}^* = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 2 & 3 & 2 & 3 & 2 & 3 & 2 \\ 3 & 1 & 0 & 1 & 0 & 0 & 3 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 3 & 2 & 0 & 2 & 0 & 1 & 3 \\ 1 & 0 & 2 & 0 & 2 & 0 & 1 & 3 \\ 1 & 3 & 2 & 2 & 2 & 0 & 1 & 3 \\ 2 & 3 & 2 & 2 & 2 & 2 & 3 & 3 \end{pmatrix} \end{array}.$$

As one can see, by comparing these matrices with that in Section 7.4.1, column 7 is not affected (except entry 0) because configuration $\mathbf{1}$ (with controls $a = b = 1$) is also absorbing, but all other entries are lowered, especially those with a small Hamming distance from 0, like configurations 1, 2 and 4, while configurations 3, 5

and 6 are less affected. This can be put into evidence by looking at the percentage difference with the unconstrained problem of Section 7.4.1

$$\frac{C_{\max} - {C^*}_{\max}}{C_{\max}} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{pmatrix} 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 & 1.000 \\ 0.797 & 0.159 & 0.145 & 0.437 & 0.145 & 0.619 & 0.511 & 0.000 \\ 0.447 & 0.051 & 0.051 & 0.207 & 0.051 & 0.300 & 0.207 & 0.000 \\ 0.363 & 0.050 & 0.050 & 0.084 & 0.050 & 0.217 & 0.076 & 0.000 \\ 0.797 & 0.145 & 0.145 & 0.511 & 0.159 & 0.619 & 0.437 & 0.000 \\ 0.708 & 0.093 & 0.079 & 0.236 & 0.093 & 0.449 & 0.236 & 0.000 \\ 0.363 & 0.050 & 0.050 & 0.076 & 0.050 & 0.217 & 0.084 & 0.000 \\ 0.319 & 0.035 & 0.035 & 0.025 & 0.035 & 0.090 & 0.025 & 0.000 \end{pmatrix} \end{array}.$$

In practice, the problem of avoiding a configuration reflects into a lowering of the probability of appearance of neighbouring ones. This suggests a possible implementation of this technique, i.e., that of adding, to the transition probabilities, the effects of a "repulsion" field that keeps the stochastic trajectories far from the "forbidden" ones.

7.5 Conclusions

We have introduced the problem of controlling probabilistic cellular automata by an action performed on the boundary of a target region. We have formulated the problem and presented the first results.

We have presented a method for obtaining the optimal control, which is rather demanding in computational terms, and we introduced also a less demanding sub-optimal method.

Finally, we presented an example of optimal control in the case in which one wants to avoid some configurations, for instance because they represent the extinction of a species.

CHAPTER 8

CONCLUSION AND FUTURE PERSPECTIVES

This work launched one of the most important problems of control theory that is of regional controllability of Cellular Automata which are considered as a good alternative of Distributed systems parameters modelled by partial differential equations. We proved the regional controllability of CA through an original approaches that is of Markov Chains and graph theory.

Firstly, by using the fact that a controlled CA can be seen as a walk of a Markov chain. We have constructed a matrix in the same way as a transition matrix of a Markov chain, then by using it we have proved the regional controllability through the definition of ergodic and regular Markov chains.

The most important point in this approach that is a give an information not only about the controllability but also the required time T to achieve it by using the definition of regular MC which states that if it exists a power such that $\mathcal{C}^T > 0$ in our case, it means that it exists a time T for which all configurations are reachable. It allows to find the required controls to reach a given desired configuration starting from an initial configuration. We proved the non-regional controllability by using the definition of an absorbing state.

Looking to the relationship between Markov Chains and Graph theory. We have used the graph theory notions to prove the regional controllability by using this approach we have given some necessary and sufficient conditions. We showed that the existence

of a Hamiltonian circuit guarantees the regional controllability. As the problem of finding a Hamiltonian circuit is considered as an NP complete problem. We have proposed another method by using the definition of strongly connected component which allows to know whether the controlled CA is regional controllable.

An efficient algorithm for generating the preimages has been proposed in order to prove controllability. On the other hand to find the required controls on the boundaries of the controlled region in the time horizon. This approach allows to find the required time for achieving controllability by finding the biggest cost rute between vertices.

In the chapter which follows, we have studied a particular case of regional controllability that is of boundary regional controllability of one dimensional CA. By using the Kalman condition we have proved the regional controllability of one dimensional CA. An efficient algorithm has been proposed to prove the regional controllability of one dimensional linear CA. This algorithm works for a big sizes of a controlled region. We investigated the case of the compliment of linear rules by using this algorithm. A backtracking algorithm based on a method for generating all preaimges of a target configuration on the region was successfully tested.

In the last chapter, we have studied the regional controlability problem of probabilistic cellular automata by using the Markov chains approach which has been used in the fourth chapter. We have presented a method for obtaining the optimal control and introduced a method to find the faster suboptimal control

Many tracks can be explored in order to continue our work. We could be interested in the following to the problem of regional controllability in optimal time. We can solve the problem of regional controlability by formulating it to a problem of optimisation with constraints. Another perspective that could eventually done in the following is studying the problem of regional observability of CA and creating the preimages of two dimensional CA. Last, as we have briefly spoken about the difficulties of controlling DPS written by PDE, it may worth examining some cases.

BIBLIOGRAPHY

[1] MS Windows NT kernel description. http://web.archive.org/web/20080207010024/http://www.808multimedia.com/winnt/kernel.htm. Accessed: 2010-09-30.

[2] A. I. Adamatzky. *Identification of cellular automata*. CRC Press, 2014.

[3] L. Affifi. Analyse régionale: Systèmes discrets sources et détection. *Thèse d'état, Université Kénitra, Maroc*, 1994.

[4] L. Afifi, A. El Jai, and E. Zerrik. Regional analysis of linear distributed parameter systems, 2005.

[5] L. Afifi, A. El Jai, and E. H. Zerrik. *Systems Theory: Regional Analysis of Infinite Dimensional Linear Systems*. Presses universitaires de Perpignan, 2012.

[6] N. Ammor. Contrôlabilité régionale d'une classe de systèmes distribués gouvernés par des modèles d'automates cellulaires. *Thèse de Doctorat d'Etat, Université de Rabat, Maroc*, 367, 2004.

[7] M. Amouroux, A. El Jai, and E. Zerrik. Regional observability of distributed systems. *International Journal of Systems Science*, 25(2):301–313, 1994.

[8] F. Bagnoli. Boolean derivatives and computation of cellular automata. *International Journal of Modern Physics C*, 3(02):307–320, 1992.

[9] F. Bagnoli, N. Boccara, and R. Rechtman. Nature of phase transitions in a probabilistic cellular automaton with two absorbing states. *Physical Review E*, 63(4):046116, 2001.

[10] F. Bagnoli, S. Dridi, S. El Yacoubi, and R. Rechtman. Regional control of probabilistic cellular automata. In *International Conference on Cellular Automata*, pages 243–254. Springer, 2018.

[11] F. Bagnoli, S. Dridi, S. El Yacoubi, and R. Rechtman. Optimal and suboptimal regional control of probabilistic cellular automata. *Natural Computing*, 18(4):845–853, 2019.

[12] F. Bagnoli, S. El Yacoubi, and R. Rechtman. Toward a boundary regional control problem for boolean cellular automata. *Natural Computing*, 17(3):479–486, 2018.

[13] A. T. Balaban. Applications of graph theory in chemistry. *Journal of chemical information and computer sciences*, 25(3):334–343, 1985.

[14] A. Belfekih. Une nouvelle approche pour la contrôlabilité et l'observabilité des systèmes distribués. *Thèse de Doctorat d'Etat, Université de Kénitra, Maroc*, 1990.

[15] A. Belfekih and A. El Jai. Regional analysis of a class of cellular automata models. In *Cellular Automata, 7th International Conference on Cellular Automata, for Research and Industry, ACRI 2006, Perpignan, France, September 20-23, 2006, Proceedings*, pages 48–57. Springer, 2006.

[16] R. Bellman and R. E. Kalaba. *Dynamic programming and modern control theory*, volume 81. Citeseer, 1965.

[17] L. Berrahmoune. Actionneurs et capteurs dans la contrôlabilité et la stabilité des systèmes distribués flexibles. *Thèse d'état, Université de Rabat, Maroc*, 1990.

[18] L. Berrahmoune. Localisation d'actionneurs pour la contrôlabilité de systèmes paraboliques et hyperboliques. application par dualité à la localisation de capteurs. *Thèse de Doctorat d'Etat, Université Mohammed V, Rabat.*, Avril 1984.

[19] A. Butkovsky, A. Egorov, and K. Lurie. Optimal control of distributed systems (a survey of soviet publications). *SIAM Journal on Control*, 6(3):437–476, 1968.

[20] J. Byl. Self-reproduction in small cellular automata. *Physica D: Nonlinear Phenomena*, 34(1-2):295–299, 1989.

[21] P. Callahan. What is the game of life. *Math. com-World of Math Online, http://www. math. com/students/wonders/life/life. html*, 2000.

[22] S.-Y. Chao. Graph theory and analysis of biological data in computational biology. In *Advanced Technologies*. IntechOpen, 2009.

[23] B. Chopard. A cellular automata model of large-scale moving objects. *Journal of Physics A: Mathematical and General*, 23(10):1671, 1990.

[24] B. Chopard. Cellular automata modeling of physical systems. In *Meyers (eds) Computational Complexity*, pages 407–433. Springer, New York, NY, 2012.

[25] B. Chopard, P. O. Luthi, and P.-A. Queloz. Cellular automata model of car traffic in a two-dimensional street network. *Journal of Physics A: Mathematical and General*, 29(10):2325, 1996.

[26] P. Choudhry, S. Sahoo, M. Chakraborty, S. Bhandari, and A. Pal. Investigation of the global dynamics of cellular automata using boolean derivatives. *Computers and mathematics with applications*, 197(9–12):1056–1079, 2008.

[27] P. P. Choudhury, B. K. Nayak, S. Sahoo, and S. P. Rath. Theory and applications of two-dimensional, null-boundary, nine-neighborhood, cellular automata linear rules. *arXiv preprint arXiv:0804.2346*, 2008.

[28] S. N. Cole. Real-time computation by n-dimensional iterative arrays of finite-state machines. In *7th Annual Symposium on Switching and Automata Theory (swat 1966)*, pages 53–77. IEEE, 1966.

[29] J. Conway. On numbers and games, acad. *Press, London, New York, San Francisco*, 1976.

[30] C. Damiani, R. Serra, M. Villani, S. Kauffman, and A. Colacci. Cell–cell interaction and diversity of emergent behaviours. *IET systems biology*, 5(2):137–144, 2011.

[31] M. Dascalu, S. Goschin, and E. Franti. Using cellular automata in traffic modeling. In *ICEIS*, pages 467–472, 2002.

[32] B. Derrida and D. Stauffer. Phase transitions in two-dimensional kauffman cellular automata. *EPL (Europhysics Letters)*, 2(10):739, 1986.

[33] A. Deutsch, S. Dormann, et al. *Cellular automaton modeling of biological pattern formation*. Springer, 2005.

[34] E. Domany and W. Kinzel. Equivalence of cellular automata to ising models and directed percolation. *Physical review letters*, 53(4):311, 1984.

[35] S. Dridi, F. Bagnoli, and S. El Yacoubi. Markov chains approach for regional controllability of deterministic cellular automata, via boundary actions. *JOURNAL OF CELLULAR AUTOMATA*, 14(5-6):479–498, 2019.

[36] S. Dridi, S. El Yacoubi, and F. Bagnoli. Boundary regional controllability of linear boolean cellular automata using markov chain. In *Recent Advances in Modeling, Analysis and Systems Control: Theoretical Aspects and Applications*, pages 37–48. Springer, 2019.

[37] S. Dridi, S. El Yacoubi, F. Bagnoli, and A. Fontaine. A graph theory approach for regional controllability of boolean cellular automata. *International Journal of Parallel, Emergent and Distributed Systems*, pages 1–15, 2019.

[38] A. El jai. Controlability and observability via actuators and sensors for a class of hyperbolic systems. *Proc, IFAC symp on cont dyst param syst, LOS Angelos USA*, 1986.

[39] A. El Jai. Distributed systems analysis via sensors and actuators. *Sensors and Actuators A: Physical*, 29(1):1–11, 1991.

[40] A. El Jai. Analyse régionale des systèmes distribués. *ESAIM: Control, Optimisation and Calculus of Variations*, 8:663–692, 2002.

[41] A. El Jai and S. El Yacoubi. On the relations between actuator structures and final-constraint minimum-energy problem. *Sensors and Actuators A: Physical*, 33(3):175–182, 1992.

[42] A. El Jai and A. Pritchard. Sensors and actuators in distributed systems. *International Journal of Control*, 46(4):1139–1153, 1987.

[43] A. El Jai, M. Simon, E. Zerrik, and A. Pritchard. Regional controllability of distributed parameter systems. *International Journal of Control*, 62(6):1351–1365, 1995.

[44] A. El Jai, E. Zerrik, M. Simon, and M. Amouroux. Regional observability of a thermal process. *IEEE Transactions on Automatic Control*, 40(3):518–521, 1995.

[45] S. El Yacoubi. Théorie des systèmes distribués : des edp vers les automates cellulaires. *Habilitation à Diriger des Recherches. Université de Perpignan*, 1999.

[46] S. El Yacoubi. A mathematical method for control problems on cellular automata models. *International Journal of Systems Science*, 39(5):529–538, 2008.

[47] S. El Yacoubi and A. El Jai. Cellular automata modelling and spreadability. *Mathematical and Computer Modelling*, 36(9-10):1059–1074, 2002.

[48] S. El Yacoubi and A. El Jai. Notes on control and observation in cellular automata models. *WSEAS Transactions on Computers*, 2(4):1086–1092, 2003.

[49] S. El Yacoubi, A. El Jai, and N. Ammor. Regional controllability with cellular automata models. In *Cellular Automata, 5th International Conference on Cellular Automata for Research and Industry, ACRI 2002, Geneva, Switzerland, October 9-11, 2002, Proceedings*, pages 357–368. Springer, 2002.

[50] S. El Yacoubi, A. El Jai, P. Jacewicz, and J. G. Pausas. Lucas: an original tool for landscape modelling. *Environmental Modelling & Software*, 18(5):429–437, 2003.

[51] S. El Yacoubi and P. Jacewicz. Cellular automata and controllability problem. In *CD-Rom Proceeding of the 14th Int. Symp. on Mathematical Theory of Networks and Systems, june*, pages 19–23, 2000.

[52] S. El Yacoubi, P. Jacewicz, and N. Ammor. Analyse et contrôle par automates cellulaires. *Annals of the University of Craiova-Mathematics and Computer Science Series*, 30:210–221, 2003.

[53] S. El Yacoubi and A. B. Mingarelli. Controlling the dynamics of fca rule 90 in [0, 1]t. *Journal of Cellular Automata*, 6, 2011.

[54] G. B. Ermentrout and L. Edelstein-Keshet. Cellular automata approaches to biological modeling. *Journal of theoretical Biology*, 160(1):97–133, 1993.

[55] D. B. Fogel and E. Computation. Toward a new philosophy of machine intelligence. *IEEE Evolutionary Computation*, 1995.

[56] B. Franco. *Cellular Automata in Dynamical Modelling in Biotechnologies*. World Scientific, Singapore, 1998.

[57] E. Fredkin and T. Toffoli. Conservative logic. *International Journal of theoretical physics*, 21(3-4):219–253, 1982.

[58] W. Gao, H. Wu, M. K. Siddiqui, and A. Q. Baig. Study of biological networks using graph theory. *Saudi journal of biological sciences*, 25(6):1212–1219, 2018.

[59] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63. ACM, 1974.

[60] M. Garzon. *Models of massive parallelism: analysis of cellular automata and neural networks*. Springer Science & Business Media, 2012.

[61] R. J. Gaylord and K. Nishidate. *Modeling nature: Cellular automata simulations with Mathematica®*. Springer, 2013.

[62] D. G. Green. Simulated effects of fire, dispersal and spatial pattern on competition within forest mosaics. *Vegetatio*, 82(2):139–153, 1989.

[63] C. M. Grinstead and J. L. Snell. *Introduction to probability*. American Mathematical Soc., 2012.

[64] M. D. Intriligator. The applications of control theory to economics. In *Analysis and Optimization of Systems*, pages 605–626. Springer, 1980.

[65] A. Jai. Nouvelle approche pour la modélisation des systèmes en expansion spatiale: Dynamique de végétation, tendences nouvelles en modélisation pour l'environnement. *Elsevier*, 439, 1999.

[66] E. Jen. Enumeration of preimages in cellular automata. *Complex Systems*, 3(5):421–456, 1989.

[67] I. Jeras and A. Dobnikar. Cellular automata preimages: Count and list algorithm. In *International Conference on Computational Science*, pages 345–352. Springer, 2006.

[68] D. S. Johnson and M. R. Garey. *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman, 1979.

[69] R. E. Kalman. On the general theory of control systems,. *In Proceedings of the 1st World Congress of the International Federation of AutomaticControl*, 188(2):462–471, 1960.

[70] J. Kari. Theory of cellular automata: A survey. *Theoretical computer science*, 334(1-3):3–33, 2005.

[71] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[72] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

[73] K. Kaundal. Applications of graph theory in everyday life and technology. *Imperial Journal of Interdisciplinary Research*, 3(3):892–894, 2017.

[74] R. A. Khan, K. Dihidar, et al. Vlsi architecture of a cellular automata machine. 1997.

[75] E. Kreindler and P. Sarachik. On the concepts of controllability and observability of linear systems. *IEEE Transactions on Automatic Control*, 9(2):129–136, 1964.

[76] J. L. Lions. Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles. 1968.

[77] J.-L. Lions. Controlabilité exacte des systèmes distribués: remarques sur la théorie générale et les applications. In *Analysis and optimization of systems*, pages 3–14. Springer, 1986.

[78] W. Liu. *Elementary feedback stabilization of the linear reaction-convection-diffusion equation and the wave equation*, volume 66. Springer-Verlag, Berlin Heidelberg, 2009.

[79] P.-Y. Louis and F. R. Nardi. *Probabilistic Cellular Automata*. Springer, 2018.

[80] M. Martins, H. V. de Resende, C. Tsallis, and A. de Magalhes. Evidence for a new phase in the domany-kinzel cellular automaton. *Physical review letters*, 66(15):2045, 1991.

[81] H. McIntosh. Ancestors: Commentaries on the global dynamics of cellular automata by andrew wuensche and mike lesser, departamento de aplicación de microcomputadoras, universidad autónoma de puebla, apartado postal 461,(72000) puebla, puebla, méxico, 1993.

[82] H. V. McIntosh. *Linear cellular automata via de Bruijn diagrams*. Universidad Autónoma de Puebla, 1989.

[83] W. J. Palm. An application of control theory to population dynamics. *EO Roxin. PT Liu, and RL Sternberg-(editors), DifferentIal games and control theory*, pages 59–70, 1975.

[84] S. Rajeswaran and S. Rajasekaran. A study of vehicular traffic flow modeling based on modified cellular automata. *Journal of Mathematics*, 4(5):32–38, 2013.

[85] K. H. Rosen. *Handbook of discrete and combinatorial mathematics, Second edition*. Chapman and Hall/CRC, New Delhi, India, 2017.

[86] E. D. Sontag. *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer-Verlag, New York, NY, 1998.

[87] R. Tarjan. Depth first search and linear graph algorithms. *Siam Journal of Computing*, 1(2):146–160, 1972.

[88] T. Toffoli. Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica D: Nonlinear Phenomena*, 10(1-2):117–127, 1984.

[89] G. Vichniac. Boolean derivatives on cellular automata. *Physica D 4*, 54(3):411–424, 1990.

[90] G. Y. Vichniac. Simulating physics with cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):96–116, 1984.

[91] J. Von Neumann. The theory of self -reproducing automata. Lecture Notes in Computational Science and Engineering. Univ. of Illinois Press, Urbana and London, Berlin, 1996. ISBN 3-540-65081-4.

[92] B. Voorhees. Predecessors of cellular automata states ii. pre-images of finite sequences. *Physica D: Nonlinear Phenomena*, 73(1-2):136–151, 1994.

[93] S. Wolfram. Statistical mechanics of cellular automata. *Reviews of modern physics*, 55(3):601, 1983.

[94] S. Wolfram. Physica d 10, 1 (1984). *Physica D*, 10:1, 1984.

[95] S. Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):1–35, 1984.

[96] S. Wolfram. Cellular automata: collected papers, 1994.

[97] S. Wolfram. *Cellular automata and complexity: collected papers*. CRC Press, 2018.

[98] S. E. Yacoubi. A mathematical method for control problems on cellular automata models. *International Journal of Systems Science*, 39(5):529–538, 2008.

[99] E. Zerrik. Analyse régionale des systèmes à paramètres distribués. *Thèse de Doctorat d'Etat, Université de Rabat, Maroc*, 367, 1993.

[100] E. Zerrik, L. Badraoui, and A. El Jai. Sensors and regional boundary state reconstruction of parabolic systems. *Sensors and Actuators A: Physical*, 75(2):102–117, 1999.

[101] E. Zerrik, A. Boutoulout, and H. Bourray. Boundary strategic actuators. *Sensors and Actuators A: Physical*, 94(3):197–203, 2001.

[102] E. Zerrik, A. Boutoulout, and A. E. Jai. Actuators and regional boundary controllability of parabolic systems. *International Journal of Systems Science*, 31(1):73–82, 2000.

[103] E. Zerrik, M. Ouzahra, and K. Ztot. Regional stabilisation for infinite bilinear systems. *IEEE Proceedings-Control Theory and Applications*, 151(1):109–116, 2004.

[104] K. Ztot, E. Zerrik, and H. Bourray. Regional control problem for distributed bilinear systems: Approach and simulations. *International Journal of Applied Mathematics and Computer Science*, 21(3):499, 2011.

List of publications

1. F. Bagnoli, S. Dridi, S. El Yacoubi, Raul Rechtman,
Regional control of probabilistic cellular automata. In : International Conference on Cellular Automata. Springer, Cham, 2018. p. 243-254.
DOI $https://doi.org/10.1007/978-3-319-99813-822$

2. S. Dridi, S. El Yacoubi, F. Bagnoli, A. Fontaine,
A graph theory approach for regional controllability of Boolean cellular automata. International Journal of Parallel, Emergent and Distributed Systems, 2019, p. 1-15.
DOI $https://doi.org/10.1080/17445760.2019.1608442$

3. S. Dridi, S. El Yacoubi, F. Bagnoli,
Boundary regional controllability of linear boolean cellular automata using Markov chain. In : Recent Advances in Modeling, Analysis and Systems Control: Theoretical Aspects and Applications. Springer, Cham, 2019. p. 37-48.
DOI $https://doi.org/10.1007/978-3-030-26149-84$

4. S. Dridi, S. El Yacoubi, F. Bagnoli,
Markov Chains approach for Regional Controllability of Deterministic Cellular Automata, via Boundary Actions, journal of Cellular Automata, 14(5-6): 479-498, 2019.

5. F. Bagnoli, S. Dridi, S. El Yacoubi, Raul Rechtman,
Optimal and Sub optimal regional control of probabilistic of Cellular Automata, Journal of Natural computing, 2019.
DOI $https://doi.org/10.1007/s11047-019-09763-5$

6. S. Dridi, S. El Yacoubi, F. Bagnoli,
Kalman condition and algorithms for Regional Controllability of Deterministic Cellular Automata, Mathematical methods in the applied sciences journal, 2019 (to be submitted).