



Universidad  
Rey Juan Carlos

# DOCTORAL THESIS

*Game Theory, Complexity and Control*

Author:

*Gaspar Alfaro García*

Supervisors:

*Miguel Ángel Fernández Sanjuán*

*Rubén Capeáns Rivas*

Doctoral Program in Sciences  
International Doctoral School

2025



*A mis padres, que me han soportado  
siempre. En los dos sentidos de la  
palabra.*



# **Agradecimientos**

Por la guía que me han prestado estoy muy agradecido a Miguel Ángel Fernández Sanjuán y Rubén Capeáns Rivas, mis directores de tesis sin los cuales no hubiera podido haber hecho los progresos que han permitido la conclusión de esta tesis. Miguel Ángel siempre me ha mostrado su apoyo y cordialidad, y es un gran referente para mí en el mundo de la investigación. Trabajar con Rubén ha sido un placer por la facilidad de comunicación y los buenos consejos que me ha otorgado.

Agradezco también a mis compañeros del Grupo de Dinámica No Lineal, Teoría del Caos y Sistemas Complejos de la URJC. Todos ellos me han mostrado calidez y hacen que trabajar en la universidad sea un lujo.

No podría dejar de mencionar a mi familia. Doy las gracias a estas personas maravillosas sin los cuales yo no sería el mismo. A mis padres, quienes me han apoyado en todas las decisiones que he tomado, les gustaran más o menos. A mi hermano, mi mayor referente en la vida y a Priscila, por retarme a mejorar cada día. Y por último a Marta, quien es mi faro en los momentos más oscuros y me llena de sonrisas en los más claros.

Quisiera también agradecer a la Universidad Rey Juan Carlos por la concesión del contrato C1PREDOC2021 de investigadores predoctorales en formación. Por último destaco que los trabajos de investigación realizados durante esta tesis doctoral han sido financiados por los siguientes proyectos de investigación: PID2019-105554GB-I00, y PID2023-148160NB-I00 financiados por la Agencia Estatal de Investigación (AEI) y por el Fondo Europeo de Desarrollo Regional (FEDER).



# Preface

*"I checked it very thoroughly,' said the computer, 'and that quite definitely is the answer. I think the problem, to be quite honest with you, is that you've never actually known what the question is.."*

-Douglas Adams, The Hitchhiker's Guide to the Galaxy

This doctoral thesis is the result of four years of work within the Research Group on Nonlinear Dynamics, Chaos and Complex Systems of the Rey Juan Carlos University.

Beginning with an introductory chapter presenting the basic ideas, the results are organized in five chapters which correspond to different scientific publications. In Chapter 3, the study of the public goods game with a square lattice and imitation as the evolutionary drive is included, where players chose three strategies: cooperate, defect, or punish the defectors. After an analysis of time-dependent effects on the game, we have questioned ourselves several issues concerning the complexity of the local dynamics of the game. In Chapter 4, we have answered this issue and in Chapter 5, we have followed up the question to Elementary Cellular Automata.

Another fundamental aspect of this doctoral thesis is the concept of control. First, we employ a method known as partial control to prevent orbits from escaping a transient chaotic region in nonlinear dynamics. This investigation is detailed in Chapter 6. Later, we extend this idea to a novel problem in game theory, establishing a connection between control theory and strategic decision-making. The results of this research are presented in Chapter 7, where partial control principles are applied to construct and analyze a competitive game between two players. Finally, we outline the structure of the thesis and provide a summary of each chapter.

## **Chapter 1. Introduction**

To introduce the topics of this thesis, we first give a few basics ideas of game theory, complex systems, and partial control method of controlling chaos along a review of the investigations done in the doctoral thesis.

## **Chapter 2. Methodology**

Next, to familiarize the reader with the tools and methods used in this research, we provide an explanation of their application. The most important among them are numerical simulations of agent-based models and cellular automata. Additionally, we describe the methods used to compute Lyapunov exponents and construct bifurcation diagrams.

## **Chapter 3. Time-dependent effects on the public goods game**

Here, we examine two distinct time-dependent effects on the public goods game with punishment. First, we analyze the impact of perturbations in the key parameter that governs the players' payoffs. Additionally, we investigate the effect of introducing a delay in the time it takes for punishment to influence defectors. Our main finding is that both parameter oscillations and delayed punishment hinder cooperation.

## **Chapter 4. Measuring complexity in evolutionary games with the Hamming distance**

In this chapter, we analyze the complexity of two significant social games. To this end, we measure the Hamming distance between two configurations differing by a single agent. We begin with the prisoner's dilemma, where our analysis confirms the presence of spatiotemporal chaos in certain parameter regimes, consistent with previous findings by May and Nowak. We then examine the public goods game, but the results remain inconclusive due to the inherent randomness of the evolutionary model used.

## **Chapter 5. Classification of Cellular Automata based on Hamming distance**

Building on the complexity analysis of the previous games, we now examine the complexity of Elementary Cellular Automata. We measure the Hamming distance between two initial configurations differing by a single cell and analyze its temporal evolution. Based on whether the distance exhibits periodic or chaotic behavior, we

classify each rule and find that our classification aligns with Wolfram's established framework.

## **Chapter 6. Escaping from transient chaos with partial control**

Here, we turn our focus to partial control. We develop a method for guiding a trajectory within a chaotic transient to escape directly to one of its attractors. Specifically, we propose two approaches: one that ensures the fastest possible escape and another that allows for escape after an exact predetermined number of iterations.

## **Chapter 7. Two-player Yorke's game of survival in chaotic transients.**

Using techniques from partial control, we design a game in which two players compete within a region of transient chaos. By defining and computing the winning sets, we determine the initial conditions that guarantee victory for each player. Furthermore, we analyze how the winning conditions change based on the information available to each player regarding their opponent's actions.

## **Chapter 8. Results and Discussion**

The main results of the investigations are presented and discussed.

## **Chapter 9. Conclusions**

Finally, the main conclusions are summarized.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Evolutionary Games and Cellular Automata . . . . .	1
1.1.1	Time Dependent Effects on the Public Goods Game .	1
1.1.2	Complexity in Spatial Evolutionary Games . . . . .	2
1.1.3	Complexity in Cellular Automata . . . . .	2
1.2	Partial Control . . . . .	3
1.2.1	Escape from Transient Chaotic Region . . . . .	3
1.2.2	Two-Players survival game of control . . . . .	3
	Bibliography . . . . .	4
<b>2</b>	<b>Methodology</b>	<b>7</b>
2.1	Monte Carlo agent-based simulations . . . . .	7
2.2	Numerical analysis tools . . . . .	7
2.2.1	Bifurcation diagrams . . . . .	8
2.2.2	Lyapunov exponents . . . . .	8
2.2.3	Hamming distance divergence measure . . . . .	8
	Bibliography . . . . .	9
<b>3</b>	<b>Time-dependent effects on the public goods game</b>	<b>11</b>
3.1	Introduction . . . . .	12
3.2	Model of the simulation . . . . .	13
3.3	The punishment methods: pool and peer punishment . . . . .	15
3.4	Varying the enhancement factor as oscillations on time . . . . .	16
3.5	Having a delay in punishment . . . . .	19
3.6	Discussion and conclusions . . . . .	22
	Bibliography . . . . .	22
<b>4</b>	<b>Measuring complexity in evolutionary games with Hamming distance</b>	<b>25</b>
4.1	Introduction . . . . .	25
4.2	Prisoner's Dilemma . . . . .	26
4.2.1	Model . . . . .	26
4.2.2	Hamming distance measure . . . . .	28

4.3	Public Goods Game . . . . .	33
4.3.1	Model . . . . .	33
4.3.2	Hamming distance measure . . . . .	35
4.4	Discussion and conclusions . . . . .	37
	Bibliography . . . . .	38
<b>5</b>	<b>Classification of Cellular Automata based on the Hamming distance</b>	<b>39</b>
5.1	Introduction . . . . .	40
5.1.1	Elementary Cellular Automata . . . . .	40
5.2	Hamming distance classification . . . . .	46
5.3	Discussion and conclusions . . . . .	50
	Bibliography . . . . .	51
<b>6</b>	<b>Escaping from transient chaos with partial control</b>	<b>53</b>
6.1	Introduction . . . . .	53
6.2	Escape from a region with partial control . . . . .	54
6.3	Application of the method . . . . .	55
6.3.1	Escaping as quick as possible . . . . .	55
6.3.2	Escaping at an exact number of iterations . . . . .	57
6.3.3	Shifting chaotic transients . . . . .	60
6.4	Discussion and conclusions . . . . .	63
	Bibliography . . . . .	65
<b>7</b>	<b>Two-player Yorke's game of survival in chaotic transients</b>	<b>67</b>
7.1	Introduction . . . . .	68
7.2	The game . . . . .	69
7.3	Solving the game: The winning sets . . . . .	70
7.4	Game of survival in the logistic map . . . . .	73
7.4.1	Exploring possible games in the $(u_0^B, u_0^A)$ space . . . . .	75
7.4.2	Boundary games . . . . .	75
7.4.3	Is it worth being informed? . . . . .	77
7.5	Discussion and conclusions . . . . .	79
	Bibliography . . . . .	79
<b>8</b>	<b>Results and Discussion</b>	<b>81</b>
<b>9</b>	<b>Conclusions</b>	<b>83</b>
	<b>Resumen de la tesis en castellano</b>	<b>85</b>
9.1	Introducción . . . . .	85
9.2	Metodología . . . . .	87
9.3	Resultados y Discusión . . . . .	88
9.3.1	Conclusiones . . . . .	89

# Chapter 1

# Introduction

*"Wizards don't like philosophy very much. As far as they are concerned, one hand clapping makes a noise like 'cl'."*

-Terry Pratchett, Sourcery

This thesis explores three interconnected branches of game theory and complex systems: evolutionary game theory, complexity analysis, and strategic control.

Game theory studies how individuals make strategic decisions. This helps understand better behaviors present in diverse fields, from social sciences to economics and physics [1, 2, 3]. A powerful way to study these interactions is through evolutionary game theory, which utilizes dynamical evolving populations to model the behavior of real-world communities.

## 1.1 Evolutionary Games and Cellular Automata

Through the inspection of evolutionary games, we focus on understanding how co-operation emerges and evolves in complex social systems. This remains one of the most fascinating challenges across multiple disciplines [4]. We begin by examining how cooperation can emerge in a social game when faced with temporal variations.

### 1.1.1 Time Dependent Effects on the Public Goods Game

While previous research has typically assumed static conditions, we investigate how the time dependence of the enhancement factors, representing the productivity of the returns of a economical activity, and delayed punishment affect the evolution of cooperative behavior. This approach provides novel insights into more realistic scenarios where the benefits of cooperation and the consequences of actions fluctuate over time collected in Chapter 3. In particular, we obtain that a loss of stability in

the productivity of returns diminishes cooperation between individuals, as so does delays in the punishment to defectors.

### 1.1.2 Complexity in Spatial Evolutionary Games

This evolutionary models are thoroughly considered by physicists because of their complex dynamics [3]. Complex systems theory analyzes nonlinear and emergent processes. When studying big populations, the principal and more interesting aspect is that the increasing population manifests in processes that can not be explained by examining individuals alone. Much like one can not explain the sound of clapping hands when only one hand is clapping.

Research in [5] suggests the complex formation of patterns in spatial games like the ones we have considered. Traditionally, one can quantify the complexity of a dynamical system by calculating the Lyapunov exponents. This could help us determine the complexity of the dynamic fluctuations of the strategy frequencies. This analysis would yield negative, or null values for the Lyapunov exponents since the system is at Nash equilibrium. But what we want to analyze is the complexity of this spatial patterns through the local interactions between agents in the evolutionary dynamical system, and this cannot be done through the analysis of the Lyapunov exponents or other similar complexity measures.

However, an analysis of complexity similar to the one we intend was done in [6, 7]. This study analyses the Hamming distance of configurations that differ initially by a small number of agents in a complex biological system and in rock-paper-scissors models. By applying the Hamming distance metric to both the prisoner's dilemma and public goods games, we can quantify and characterize the complexity of a system. This analysis reveals how minimal variations, through local interactions can lead to enormous global changes and helps identify parameter regions where complex dynamics emerge. Collecting the investigation in Chapter 4, we found out that the pattern formation discovered in [5] is indeed chaotic.

### 1.1.3 Complexity in Cellular Automata

Then, we shift our attention to cellular automata, as a more simple case of an evolutionary game. Included in Chapter 5, our research provides another perspective on pattern formation. Through a systematic analysis of elementary cellular automata, we explore how simple local rules can generate complex global behaviors. Furthermore, we establish a new classification of the elementary cellular automata that helps better understand the fundamental classification of Wolfram for cellular automata [8].

The most important breakthrough in the research was the observation of transient chaotic dynamics present in Wolfram class 4, underlying the phenomenon of the edge of chaos [9].

## 1.2 Partial Control

Finally, we extend the concept of partial control [10, 11, 12, 13] to competitive scenarios, developing a novel framework for analyzing strategic interactions in chaotic systems.

Partial control is a method of chaos control. Instead of forcing a single controlled trajectory, the method tries to avoid some unwanted regions in the dynamics by strategically controlling the trajectory to a range of *safe* points, traditionally called *safe sets*.

### 1.2.1 Escape from Transient Chaotic Region

In Chapter 6, we extend the consideration of partial control to a case in which the controller aims to expel the trajectory as quickly as possible, or instead, in an orderly manner, from a chaotic region.

We obtain the value of control needed to control trajectories starting from all possible initial conditions so the trajectory escapes as we intend. These values are obtained from the quick escape function when we want the trajectory to escape swiftly in no more iterations than the controller sets; or from the exact escape functions when the trajectories must escape at an exact number of iterations.

These functions depend on the system's noise bound, and when setting the control to be lesser than this noise bound, we find that there are initial conditions that, surprisingly, can be controlled to achieve the goal. This is a key feature of the partial control method.

### 1.2.2 Two-Players survival game of control

The investigation culminates in Chapter 7 with a two-player game where participants compete for control over a chaotic trajectory. The game, which may seem dependent of the player's choices is found to be solvable with the help of partial control. With these tools we identify the initial conditions that assure the victory for each player.

We illustrate our game with the logistic map, which provides us with a rich example of asymmetry in player's goals. The system's dynamics favor the player who intends on leaving the chaotic region, while the player who wants to stay there. Nonetheless, this undermined player can still win at some cases, even with lesser control bound than their opponent!

Furthermore, the study analyses the importance of player's information, which can alter the solution to the game.

Our findings contribute to a deeper understanding of how game theory, complexity, and control interplay in social and dynamical systems, with potential applications ranging from social physics to game theory and chaos control.

## Bibliography

- [1] P. Kollock, Social dilemmas: the anatomy of cooperation, *Annu. Rev. Sociol.* **24**, 183–214 (1998). <https://doi.org/10.1146/annurev.soc.24.1.183>
- [2] Y. Xiao, Y. Peng, Q. Lu, and X. Wua, Chaotic dynamics in nonlinear duopoly Stackelberg game with heterogeneous players, *Physica A* **492**, 1980–1987 (2018). <https://doi.org/10.1016/j.physa.2017.11.112>
- [3] W. Hu, G. Zang, H. Tian, and Z. Wang, Chaotic dynamics in asymmetric rock-paper-scissors games, *IEEE Access* **7**, 175614–175621 (2019). <https://doi.org/10.1109/ACCESS.2019.2956816>
- [4] M. Jusup, P. Holme, K. Kanazawa, M. Takayasu, B. Podobnik, L. Wang, W. Luo, T. Klanjšček, J. Fan, S. Boccaletti, and M. Perc, Social physics. *Phys. Rep.* **948**, 1–148 (2022).  
<https://doi.org/10.1016/j.physrep.2021.10.005>
- [5] M. A. Nowak and R. M. May, Evolutionary games and spatial chaos. *Nature* **359**, 826–829 (1992). <https://doi.org/10.1038/359826a0>
- [6] D. Bazeia, M. B. P. N. Pereira, A. V. Brito, B. F. Oliveira, and J. G. G. S. Ramos, A novel procedure for the identification of chaos in complex biological systems. *Sci. Rep.* **7**, 44900 (2017). <https://doi.org/10.1038/srep44900>
- [7] D. Bazeia, J. Menezes, B. F. De Oliveira, and J. G. G. S. Ramos, Hamming distance and mobility behavior in generalized rock-paper-scissors models. *EPL* **119**, 58003 (2017). <https://doi.org/10.1209/0295-5075/119/58003>
- [8] S. Wolfram, Universality and complexity in cellular automata. *Physica D* **10**, 1–35 (1984). [https://doi.org/10.1016/0167-2789\(84\)90245-8](https://doi.org/10.1016/0167-2789(84)90245-8)
- [9] N. H. Packard, Adaptation toward the edge of chaos, Dynamic patterns in complex systems **212**, 293–301 (1988).  
<https://doi.org/10.1142/9789814542043>
- [10] J. Aguirre, F. d’Ovidio, and M. A. F. Sanjuán, Controlling chaotic transients: Yorke’s game of survival, *Phys. Rev. E* **69**, 016203 (2004).  
<https://doi.org/10.1103/PhysRevE.69.016203>
- [11] Juan Sabuco, Miguel A. F. Sanjuán, and James A. Yorke, Dynamics of partial control, *Chaos* **22**, 047507 (2012). <https://doi.org/10.1063/1.4754874>
- [12] R. Capeáns, J. Sabuco, and M. A. F. Sanjuán, Beyond partial control: controlling chaotic transients with the safety function. *Nonlinear Dyn.* 107:2903–2910 (2022). <https://doi.org/10.1007/s11071-021-07071-1>

- [13] R. Capeáns, J. Sabuco, and M. A. F. Sanjuán, A new approach of the partial control method in chaotic systems. *Nonlinear Dyn.* 98:873–887 (2019).  
<https://doi.org/10.1007/s11071-019-05215-y>



# **Chapter 2**

# **Methodology**

*“The purpose of computing is insight, not numbers.”*

-Richard Hamming

For this doctoral thesis, we employ a theoretical and computational approach, employing a combination of analytical methods and numerical simulations to corroborate theoretical predictions with observable phenomena. These simulations provided us with insights and a better understanding of the problems at hand. Following, we describe the methods applied.

## **2.1 Monte Carlo agent-based simulations**

For Chapters 3 and 4 we make agent-based simulations of individuals playing social games. Each agent has a strategy to play the game. Games are played between neighbors in a square grid and gaining different payoff, which represent the fitness of each player. After each game has been played and we have obtained the payoffs, through the Monte Carlo method [1], random individuals are selected to update their strategy to that of a random neighbor. The likelihood of adopting the new strategy will depend on the difference of both agents’ payoff. The games are iterated many times and we can see the dynamical changes of strategies by plotting each strategy in different colors. These simulations were produced using custom made code in the *Julia* programming language.

## **2.2 Numerical analysis tools**

This collects methods to analyze systems and data to characterize nonlinear systems. This methods include elaboration of bifurcation diagrams, calculations of Lyapunov

exponents, and the algorithm used in Chapters 4 and 5 to asses complexity using the Hamming distance.

### 2.2.1 Bifurcation diagrams

A bifurcation diagram [2] is a graphic that represents the different dynamical behaviors in the systems dynamics according to the variation of a certain parameter. The graphic plots all the values of the steady state through time after relaxation versus each value of the parameter. The bifurcation diagram shows a single line when the system is in a fixed state. As the parameter changes, the line may bifurcate in two or more lines as the system's fixed state becomes a periodic state. For system that present nonlinear behavior, as the parameter reaches a critical value, the period increases by doubling, until the critical value is reached and a continuous set of points is represented in the bifurcation diagram for each parameter value. This indicates that the system has a chaotic attractor and the dynamics is chaotic.

To obtain the bifurcation diagram one must first let the dynamics get to a steady state where the transient dynamics is discarded. In the case of multistable systems, one must calculate the steady state for more than one initial condition, which produces different steady states because they reach different attractors. By plotting the steady state of each initial condition separately or in a different color, we can analyze the differences.

A bifurcation diagram for a multistable system was plotted in Chapter 6. It was computed using *MATLAB*.

### 2.2.2 Lyapunov exponents

To know when the dynamics of a system is chaotic, one can measure the local divergence of trajectories. One method to do this is to calculate the Lyapunov exponents [3]. There are as many Lyapunov exponents as the dimensionality of phase space. In chaotic systems at least one of the Lyapunov exponents is positive, so one could simply calculate the Maximum Lyapunov Exponent MLE to determine the complexity of the systems. For discrete time systems, the MLE can be obtained with the following formula:

$$\lambda(x_0) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \log \left| \frac{df(x_i)}{dt} \right| \quad (2.1)$$

This formula was used in Chapter 6 implemented using *MATLAB*.

### 2.2.3 Hamming distance divergence measure

The Hamming distance measures the number of different elements in a group, vector or matrix. To measure the divergence of similar initial conditions in Chapters 4

and 5, we first let the system get to a steady state. Then, we made a copy of the system and changed only one agent state. After this, we let the systems evolve and measure the Hamming distance between the two configurations. If the Hamming distance grows, this is a sign that the system may be chaotic (or random). Since the size of the system is finite, the Hamming distance saturates at a certain point which can be calculated. It depends on the proportions of the fixed-state frequencies of each strategy.

In Chapter 5 we found that the different cellular automata Wolfram classes can be obtained by looking at the Hamming distance divergence, whether it is zero, or grows, and depending on the behavior when it saturates.

## Bibliography

- [1] D. Reiter. The Monte Carlo method, an introduction. In *Computational Many-Particle Physics* Lecture Notes in Physics **739**, 63—78 (Springer, Berlin, Heidelberg, 2008). [https://doi.org/10.1007/978-3-540-74686-7\\_3](https://doi.org/10.1007/978-3-540-74686-7_3)
- [2] H. E. Nusse, J. A. Yorke, E. J. Kostelich, Bifurcation Diagrams. In *Dynamics: Numerical Explorations*, Applied Mathematical Sciences **101** **101** (Springer, New York, NY 1994). [https://doi.org/10.1007/978-1-4684-0231-5\\_6](https://doi.org/10.1007/978-1-4684-0231-5_6)
- [3] A. Pikovsky and A. Politi, *Lyapunov Exponents: A Tool to Explore Complex Dynamics*, (Cambridge University Press, 2016).



## **Chapter 3**

# **Time-dependent effects on the public goods game**

*“Time is the longest distance between two places.”*

-Tennessee Williams

Understanding how individuals cooperate and form groups is fundamental to many fields, from biology to social sciences and economics [1, 2, 3]. When we study these interactions, we find that when one adds more agents to a population, the problem does not scale linearly. On the contrary it creates entirely new patterns of behavior that we could not predict by looking at individuals alone. These emerging patterns make the study of populations a fascinating example of complex systems dynamics, where simple rules can lead to intricate and surprising outcomes.

Social physics research has developed various tools to understand these patterns [4], but one framework stands out as particularly powerful: Evolutionary Game Theory (EGT). This approach combines the mathematical precision of game theory with the dynamic nature of evolution, helping us understand how behaviors spread and change in populations over time.

One of the most intriguing questions EGT helps us explore is how altruism emerges. Why would individuals choose to help others at a cost to themselves? This seemingly counterintuitive behavior has been studied through various games, including the well-known rock-paper-scissors [5], the prisoner’s dilemma [6], and the public goods game [7].

---

G. Alfaro and M. A. F. Sanjuán, Time-dependent effects hinder cooperation on the public goods game, *Chaos, Solitons Fractals* **160**, 112206 (2022).  
<https://doi.org/10.48550/arXiv.2501.12188>

### 3.1 Introduction

In the public goods game (PGG) a community where each person can either contribute to shared resources, cooperate, or benefit without contributing, defect. Cooperators ( $C$ ) contribute a unit amount to each of their groups, while defectors ( $D$ ) contribute nothing. The collective contributions are then multiplied by an enhancement factor  $r$  and then the sum is shared equally among all group members, regardless of their contributions. The enhancement factor  $r$  accounts for the profit of a certain investment, including the benefits of working as a group with greater productivity than if individuals were working alone.

This setup creates an interesting dilemma: while everyone benefits when many people cooperate, individuals can gain more by defecting and letting others do the work. Cooperation only flourishes when the enhancement factor  $r$  approaches the group size [8]. Otherwise, defectors eventually dominate the population, leading to the “tragedy of the commons”, a situation where pursuing individual interests leads to collective failure.

To encourage cooperation, societies often employ various mechanisms. These include allowing people to move between groups [9], maintaining reputations [10], rewarding cooperators [11], or punishing those who do not contribute. Punishment can take different forms, from social exclusion [12] to monetary fines [13]. Our work focuses on monetary punishment by introducing a third type of individual: punishers ( $P$ ). These individuals not only contribute to the public good but also pay extra to penalize defectors. While punishers earn less than normal cooperators, since they bear the cost of punishment, their presence can make defection less attractive and thereby increase overall cooperation. However, maintaining cooperation requires a critical mass of punishers, as shown in [12].

Previous research have assumed that the rules of the game remain constant over time. However, real-world situations rarely work this way. While some researchers have explored how allowing people to choose their group members affects outcomes [14], few have investigated what happens when the fundamental parameters of the system change over time. The enhancement factor  $r$  is particularly likely to fluctuate in real situations as of changes in the productivity of the investments. To understand these effects, we consider what happens when  $r$  oscillates sinusoidally over time. We have chosen this dependence as a simple pattern that can help us understand more complex variations.

Moreover, real-world actions rarely have immediate consequences, yet most studies assume instant punishment. We explore this by introducing a time delay  $\tau$  as the intercept between the instant when someone defects and when they receive punishment. The effect on delay was previously studied with human subjects [15], but with a sample much smaller than in our simulations.

Our research reveals two key findings. First, large amplitudes of oscillation in the enhancement factor make it harder for cooperation to persist. Second, delays in punishment also hinder cooperation. We also noticed that rapid fluctuations,

similar to random noise, do not significantly affect outcomes.

## 3.2 Model of the simulation

We have implemented the public goods game as a spatial model where individuals interact with their neighbors in a square grid. Each individual participates in five different groups, with each group containing exactly five members ( $G = 5$ ). When an individual participates in a game, they receive a payoff,  $\Pi^g$ , based on their strategy and the strategies of others in that particular group. Their accumulated payoff,  $\Pi$  comes from adding up their earnings from all five games they participated in,  $\Pi = \sum_g^G \Pi^g$ . This payoff is a measure of the fitness of an individual in the system.

Our simulation grid has  $L$  cells on each side. We typically use  $L = 300$  throughout all the simulations in this chapter unless specified otherwise. This means we are modeling a total population of  $N = L^2$  individuals, distributed among cooperators ( $N_C$ ), defectors ( $N_D$ ), and punishers ( $N_P$ ), so that  $N = N_C + N_D + N_P$ .

In nature and society, successful behaviors tend to spread, people often imitate strategies that seem to be working well for others. We model this through an imitation rule implemented in a Monte Carlo simulation. In each step, we randomly select an individual,  $x$ , and one of their neighbors  $y$ . After they have both played their games and received their payoffs,  $x$  might adopt  $y$ 's strategy according to a probability that depends on the payoff of both. The probability follows the Fermi distribution:

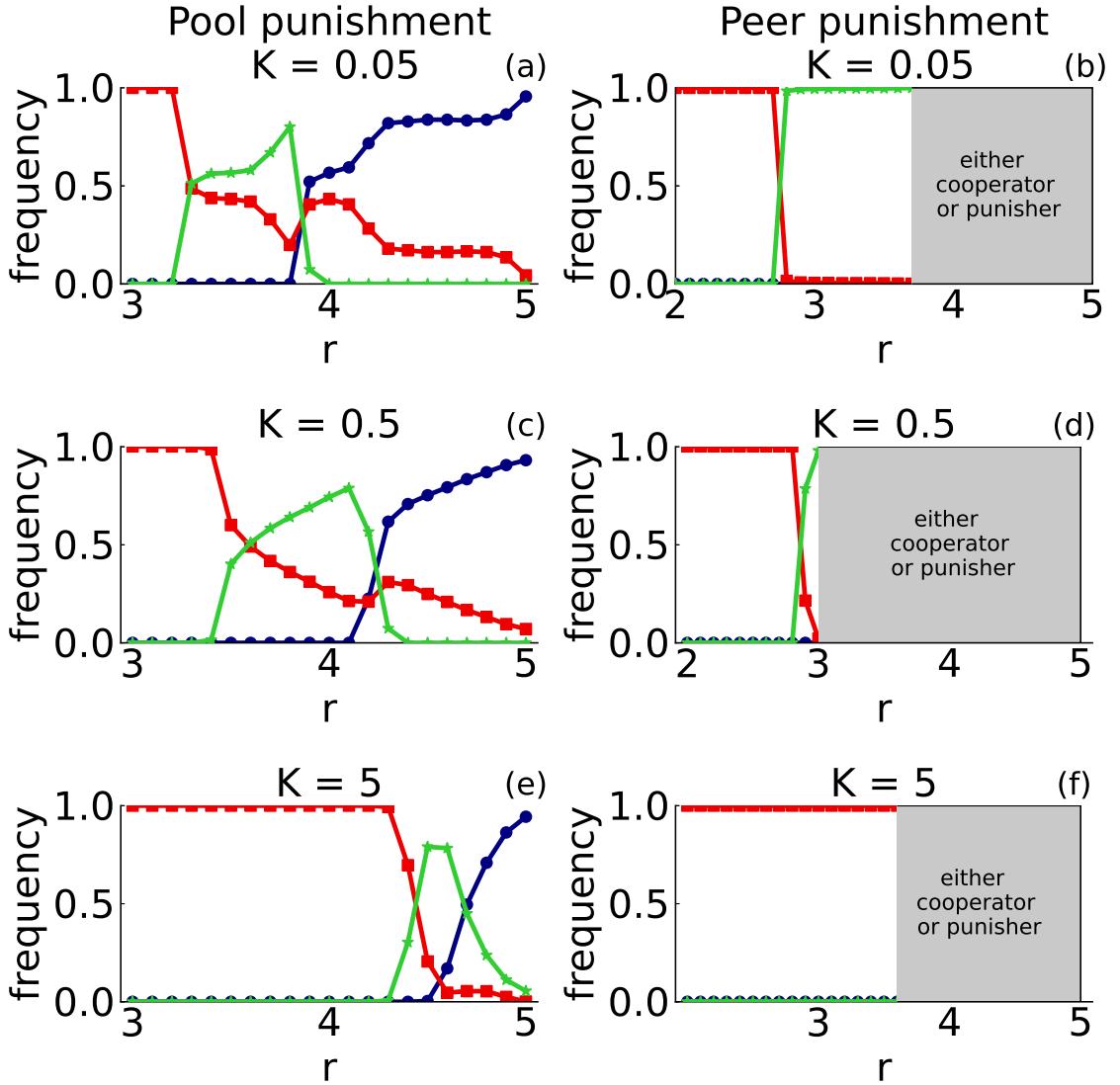
$$W(s_x \rightarrow s_y) = \frac{1}{1 + \exp[(\Pi_x - \Pi_y)/K]}, \quad (3.1)$$

If  $y$  is doing much better than  $x$  ( $\Pi_y \gg \Pi_x$ ), the probability of  $x$  adopting  $y$ 's strategy becomes very high. On the other hand, if  $x$  is doing much better than  $y$  ( $\Pi_x \gg \Pi_y$ ),  $x$  will likely keep their current strategy. When both are doing similarly well, the change becomes more random.

$K$  serves as a way to regulate this randomness that appears when adopting a strategy. Humans commit errors and are not purely driven by rationality. Therefore they do not always act to the best response. As  $K \rightarrow 0$  the individuals always change the strategy if their payoff is smaller than the neighbor's. As  $K \rightarrow \infty$  the probability of change is  $1/2$ , regardless of the payoff. As [16] suggests, we set  $K = 0.5$  as a fully representative value.

We define one Monte Carlo Step (MCS) as  $N$  iterations, meaning that on average, each individual has had one opportunity to update their strategy. This represents one generation in our evolutionary timeline.

The starting configuration of individual's strategies can change the outcome of the simulation. All the results presented in this paper come from randomly assigning strategies to individuals across the grid at the beginning.



**Figure 3.1.** Frequency of each strategy after 5000 MCS of a PGG simulation as a function of  $r$  for different values of the stochastic noise regulator  $K$ . Cooperators in blue, defectors in red and punishers in green. For greater noise values, cooperation is less profitable. (b) (d) (f) For peer punishment, at some  $r$  values defectors become extinct, so punishers have the same payoff as cooperators. With sufficiently large relaxation times, either punishers or cooperators will go extinct as of neutral drift. (f) For  $K = 5$  the phase shift between whole domination from defectors and its extinction is very sudden (the step between consecutive  $r$  values is 0.1). The lines are used to guide the eye, and its width is larger than the corresponding error. We have used the following parameter values:  $\beta = 0.125$ ,  $\gamma = 0.0125$ .

### 3.3 The punishment methods: pool and peer punishment

Societies have long used monetary fines to discourage undesirable behavior and promote cooperation. Think of parking tickets, environmental penalties, or business regulations. All these represent ways that communities punish those who do not cooperate with social rules. In our model, we explore two distinct approaches to implementing such punishment: pool punishment and peer punishment. In pool punishment, punishers contribute to a common fund. Each time punishers participate in the public goods game, they pay a small amount ( $\gamma = 0.0125$ ) into this communal punishment fund. When they encounter a defector, the defector must leave a fine ( $\beta = 0.125$ ). Importantly, this fine is applied only once per defector, regardless of how many punishers are in the group.

Peer punishment works differently. Instead of contributing to a central fund, punishers directly confront defectors. Each punisher pays a personal cost ( $\gamma = 0.0125$ ) to punish each defector they encounter, and each defector receives a separate fine ( $\beta = 0.125$ ) from each punisher. This can result in more severe punishment when multiple punishers are present.

Both approaches reflect different real-world systems for enforcing cooperation. Pool punishment reflects individuals reporting defectors to a gubernamental enforcement institution, paying taxes to maintaining it. Alternatively in peer punishment, the punishers take a more personal approach conflicting directly with the defectors.

These approaches lead to different mathematical expressions for how individuals fare in the game.

For pool punishment:

$$\begin{aligned} \text{Cooperators earn: } \Pi_C^g &= \frac{r}{G}(N_C^g + N_P^g) - 1 \\ \text{Punishers earn: } \Pi_P^g &= \frac{r}{G}(N_C^g + N_P^g) - 1 - \gamma \\ \text{Defectors earn: } \Pi_D^g &= \left\{ \begin{array}{ll} \frac{r}{G}(N_C^g + N_P^g) & \text{if } N_P^g = 0 \\ \frac{r}{G}(N_C^g + N_P^g) - \beta & \text{if } N_P^g \neq 0. \end{array} \right\} \end{aligned} \quad (3.2)$$

For peer punishment, the equations change to reflect the multiple punishments possible:

$$\begin{aligned} \text{Cooperators still earn: } \Pi_C^g &= \frac{r}{G}(N_C^g + N_P^g) - 1 \\ \text{Punishers now earn: } \Pi_P^g &= \frac{r}{G}(N_C^g + N_P^g) - 1 - \gamma N_D^g \\ \text{Defectors now earn: } \Pi_D^g &= \frac{r}{G}(N_C^g + N_P^g) - \beta N_P^g. \end{aligned} \quad (3.3)$$

When we simulate these systems over 5000 generations (Monte Carlo Steps), we observe how different strategies succeed under varying conditions in Fig. 3.1. At low enhancement factors  $r$ , defection dominates because the benefits of cooperation are not high enough to offset its costs. As  $r$  increases, punishers begin to thrive because the greater benefits of cooperation make their enforcement role more valuable.

At even higher  $r$  values, normal cooperators can sometimes outcompete punishers because the high returns make punishment less necessary.

The stochastic noise parameter  $K$  influences these patterns significantly. Higher noise levels make cooperation harder to maintain, shifting the transition points toward higher  $r$  values. This reflects how uncertainty or imperfect information can make maintaining cooperation more challenging.

Peer punishment shows some particularly interesting dynamics. Under certain conditions, defectors can be completely eliminated. When this happens, punishers and cooperators become equally successful since there's no one left to punish. Then, random chance eventually leads one strategy to dominate through what is called neutral drift. The outcome depends on the relative numbers of punishers and cooperators when defectors disappear, and possibly on their spatial distribution and the system's size.

Comparing the two punishment systems under the same fine  $\beta$  and cost  $\gamma$  values, peer punishment proves more effective at promoting cooperation. This makes intuitive sense since defectors face multiple fines under peer punishment, making the consequences of non-cooperation more severe.

### 3.4 Varying the enhancement factor as oscillations on time

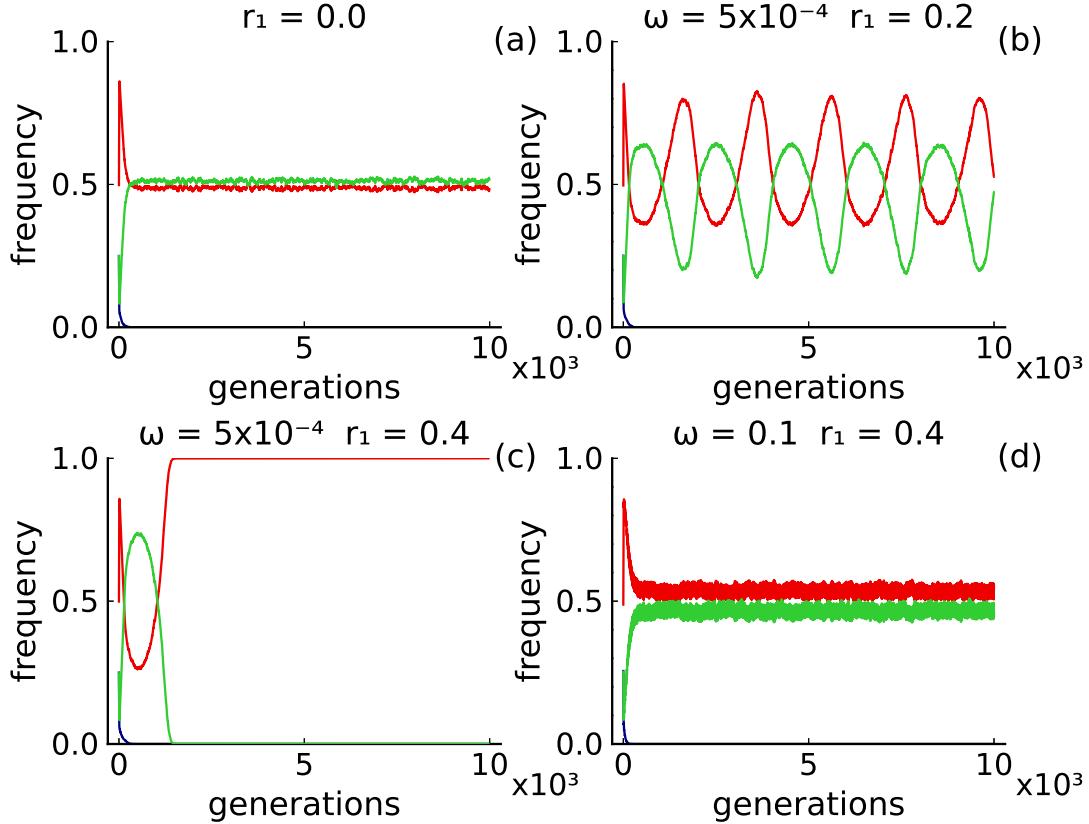
In real societies, the benefits of cooperation rarely remain constant. Consider investing in a business venture: some days the market soars, other days it plunges. Another example of cooperative activity is farming, which experiences cycles of abundance and scarcity. The public goods game, as typically studied, simplifies this reality by assuming constant returns on cooperation. To better understand how these real-world fluctuations affect cooperation, we need to introduce time-varying returns into our model.

We model these changing investment returns by making the enhancement factor  $r$  oscillate over time. While real-world fluctuations can be complex, we start with a simple sinusoidal pattern that captures the essential behavior of periodic changes, a sinusoidal oscillation:  $r = r_0 + r_1 \sin(\frac{2\pi\omega}{L^2}t)$ .

The frequency  $\omega$ , measured in inverse Monte Carlo Steps ( $MCS^{-1}$ ), lets us explore different types of real-world fluctuations. Low frequencies might represent long-term economic cycles, while high frequencies could model rapid fluctuations like daily market volatility or random noise in the system.

When we examine in Fig. 3.2 how populations evolve under these oscillating conditions, we discover that larger oscillations make it harder for cooperation to survive. When we increase the amplitude  $r$ , defectors gain an advantage, sometimes even taking over the entire population, as shown in Fig. 3.2(c). This suggests that stability, in the sense of predictable returns on cooperation – plays a crucial role in maintaining cooperative behavior.

The frequency of oscillations also matters. When oscillations happen very quickly (Fig. 3.2(d)), the population behaves almost as if there were no oscillations at all.

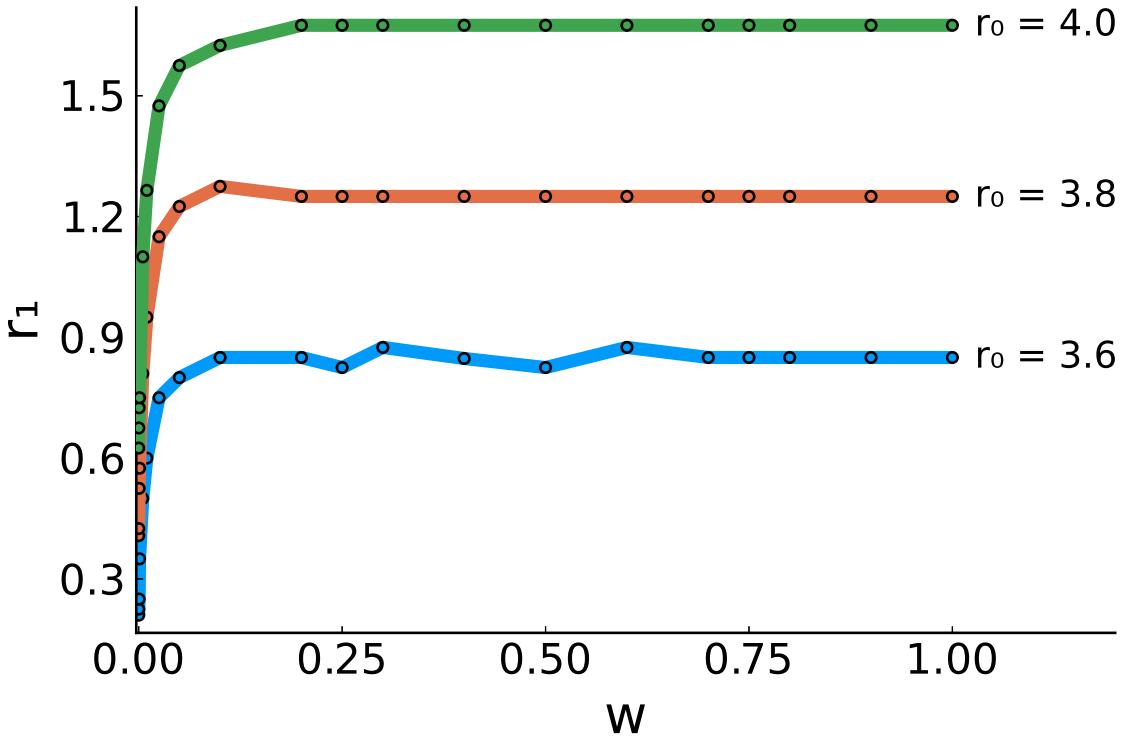


**Figure 3.2.** Frequency of each strategy as time passes with an oscillating enhancement factor of the form  $r = r_0 + r_1 \sin(\frac{2\pi\omega}{L^2}t + \delta)$ , where  $r_0 = 3.6$  is the mean value of  $r$ ,  $\omega$  is the oscillation frequency in units of  $\text{MCS}^{-1}$  and  $\delta = 0$ . Results are made in the pool punishment. Cooperators in blue, defectors in red and punishers in green. (a) At the value of  $r = 3.6$  the punishers and defectors are almost on equal terms rapidly oscillating due to noise. (b) For small oscillation frequencies, the defectors and punishers periodically dominate one another, being the punishers the ones with a greater mean frequency. (c) The amplitude of the oscillation is so big that defectors completely dominate after one cycle. (d) As the oscillation frequency increases, the dynamics is more similar as compared to the case without any  $r$  oscillation; while very quick  $r$  oscillations, possibly another type of noise, are insignificant to final frequency. We have used the parameter values  $\beta = 0.125$ ,  $\gamma = 0.0125$ .

This makes intuitive sense: if conditions change faster than people can adapt their strategies, they respond to the average conditions instead.

These findings hold true regardless of whether we use pool or peer punishment, and they persist across different levels of decision-making noise,  $K$ . The fundamental relationship between instability and reduced cooperation appears to be a robust feature of the system.

To understand exactly when defection becomes dominant, we created Fig. 3.3.

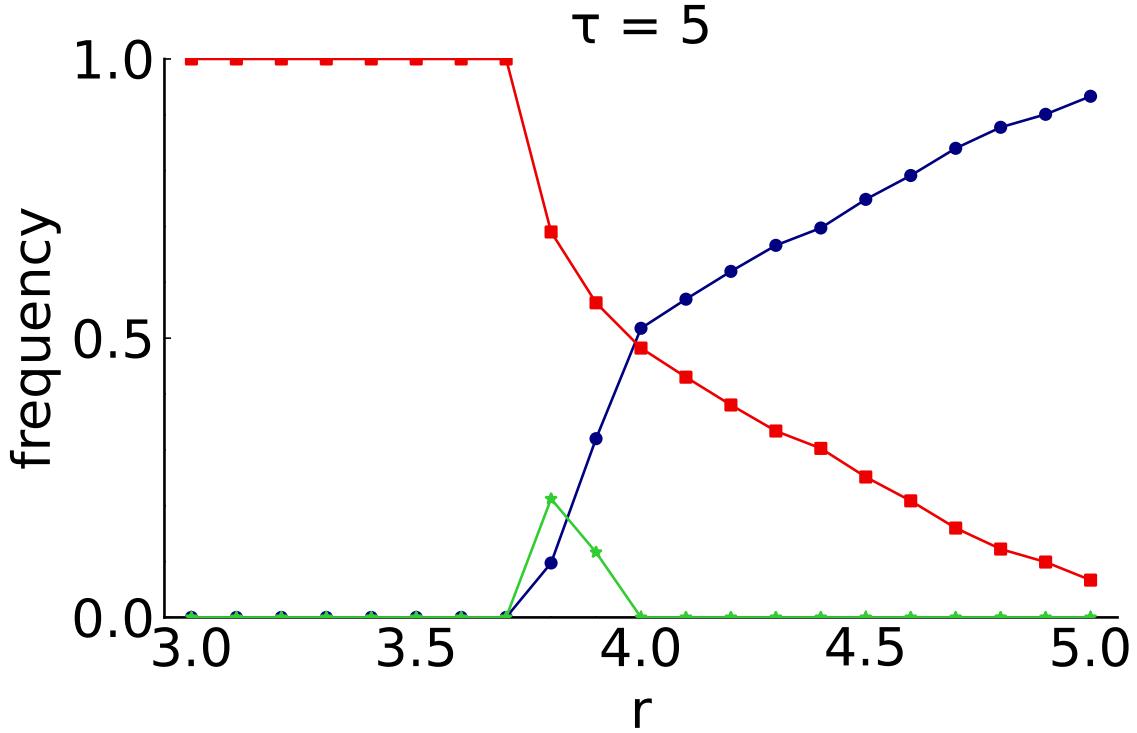


**Figure 3.3.** The figure represents the amplitude  $r_1$  of the oscillation  $r = r_0 + r_1 \sin(\frac{2\pi\omega}{L^2}t + \delta)$  for different values of  $r_0$ , that limits the phase of only defectors (greater amplitudes) and defectors plus punishers (lower amplitudes) in the strict pool punishment regime with a punisher threshold  $T = 0$ . The limiting amplitude  $r_1$  grows as  $r_0$  raises and, at low oscillation frequencies, when  $\omega$  raises. The lines are used to guide the eye, and its width is comparable to the corresponding error. We have used the parameters:  $\delta = 0$ ,  $\beta = 0.125$ ,  $\gamma = 0.0125$ .

This figure shows the critical amplitude  $r_1$  that separates regions where punishers can survive from regions where defectors completely take over, plotted against the oscillation frequency  $\omega$  for different values  $r_0$ . The resulting curve reveals several key insights:

First, the critical amplitude increases with frequency until reaching a plateau. This means that faster oscillations need larger amplitudes to disrupt cooperation completely. At low frequencies, even small oscillations can lead to defector dominance because there are long periods when  $r$  dips low enough for defectors to multiply significantly. As frequency increases, the system has less time to respond to each low- $r$  period, requiring larger amplitudes to achieve the same effect.

Second, higher baseline values of  $r_0$  push the critical amplitude higher. This makes sense since when cooperation is more profitable on average, it takes larger disruptions to make defection the winning strategy.



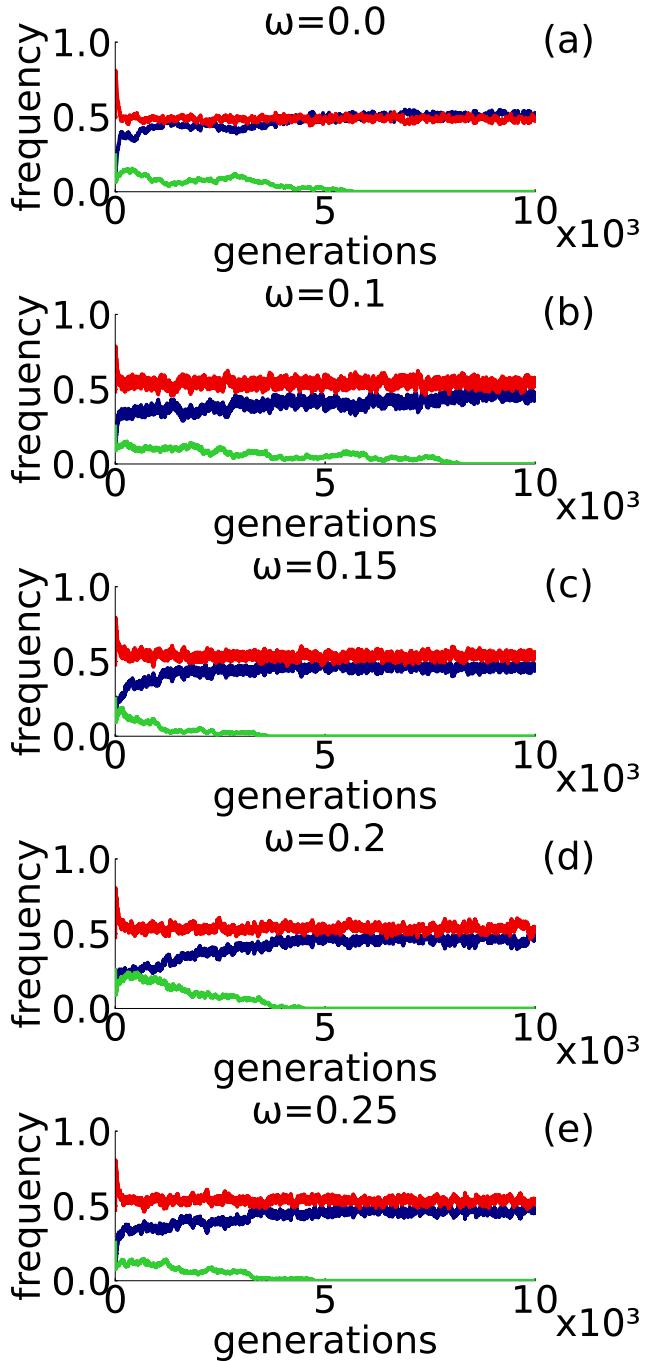
**Figure 3.4.** Frequency of each strategy averaged on the last iterations of a PGG simulation with a delay  $\tau = 5$  MCS when the defectors are charged with the fine as a function of  $r$  in the pool punishment regime. Cooperators in blue, defectors in red and punishers in green. Punishers almost do not appear and defectors are more abundant than on Fig. 3.1(c). The lines are used to guide the eye, and its width is larger than the corresponding error. We have used the parameters:  $\beta = 0.125$ ,  $\gamma = 0.0125$ ,  $L = 100$ .

We observed these patterns starting from  $\omega = 0.00001$ , our lowest tested frequency. Interestingly, at even lower frequencies, punishers dominate the system. This occurs because we start our simulations with  $\delta = 0$ , meaning  $r$  initially increases, giving cooperation an early advantage that becomes decisive over very long oscillation periods.

This analysis reveals that cooperation is most vulnerable to slow, large-amplitude changes in the benefits of cooperation. Fast changes, even of large amplitude, have less impact because populations adapt to average conditions. These findings suggest that societies are more resilient to rapid fluctuations in cooperative benefits than to long-term cycles of boom and bust.

### 3.5 Having a delay in punishment

In real life, actions and their consequences rarely happen simultaneously. When you speed on the highway, the ticket might arrive weeks later. This delay between action and consequence, or punishment, can significantly affect behavior. Understanding



**Figure 3.5.** Frequency of each strategy as time passes of a PGG simulation with a delay  $\tau = 5$  MCS when the defectors are charged with the fine for different oscillation frequencies  $\omega$  in the pool punishment regime. (a) No oscillation on the factor  $r$ . The oscillation of the frequency is due to the delay. With oscillation in the factor  $r$ , the (b-e) graphics look similar. There is no apparent difference in (d) although  $w = 0.2 = 1/\tau$  is the frequency we expected for a resonant behaviour. Cooperators in blue, defectors in red and punishers in green. We have used the parameters:  $r_0 = 4.0$ ,  $r_1 = 0.5$ ,  $\beta = 0.125$ ,  $\gamma = 0.0125$ .

these effects is crucial for designing effective enforcement systems.

To explore how such delays influence cooperation, we modified our model to include a time lag so defectors stay some iterations of the game without receiving the punishment of their infringement. Punishers detect defection and moments later, defectors actually pay their fines. This delay gives violators a temporary advantage as they can continue benefiting from their behavior before facing consequences.

The added complexity of tracking delayed punishments significantly increased computational demands, so we reduced our population size to  $L=100$  to keep simulation times manageable. Even with this smaller population, the results proved illuminating.

Looking at Fig. 3.4, which shows the long-term frequency of each strategy with a delay of  $\tau = 5$  MCS, we see a dramatic shift from our previous results. Comparing this to Fig. 3.1(c) (the no-delay case), we observe two significant changes: (1) Punishers have almost completely disappeared across most values of the enhancement factor  $r$ . (2) While normal cooperators appear at lower  $r$  values than before, defectors maintain a much stronger presence.

This weakening of cooperation makes intuitive sense when we consider how people learn and adapt their behavior. Just as children need to connect punishment directly with their actions to understand consequences, our simulated individuals become less likely to change their behavior when punishment is delayed. The longer the gap between action and consequence, the weaker the learning effect becomes. A principle that holds true whether we use peer punishment or vary the noise in decision-making.

To further understand these delay effects, we investigated whether they might interact with the oscillating returns we studied earlier. Fig. 3.5 shows how populations evolve over time with both delay ( $\tau = 5$ ) and oscillating enhancement factors ( $r_0 = 3.8$ ,  $r_1 = 0.5$ ) at different frequencies. We were particularly interested in whether we might see resonance, a strengthening of effects, when the oscillation frequency matches the natural frequency created by the delay ( $\omega = 1/\tau = 0.2$ ).

Nonetheless we found no evidence of such resonance. The population dynamics looked similar across different frequencies, even near the theoretical resonance point as seen in Fig. 3.5(d). Even more unexpectedly, we didn't observe the periodic oscillations we might have expected given our sinusoidal variation in returns.

What explains this lack of periodicity? Our first thought was that random noise in decision-making might be masking any periodic patterns. However, when we ran simulations in the noiseless regime ( $K \rightarrow 0$ ), we still saw no periodic behavior. Then, another account for stochasticity is the randomness of our Monte Carlo simulation. Specifically, in how we randomly choose which individuals get to update their strategies. This inherent randomness in the update process appears to overwhelm any periodic forcing from our oscillating enhancement factor. Another fact that restrains the observation of oscillation and resonance is the large value of  $\omega$ , as high-frequency oscillations do not provide sufficient time for population to adapt.

This finding highlights an important principle: while both delayed punishment and oscillating returns can independently weaken cooperation, their combined effects do not produce simple, predictable patterns, at least for the parameters studied. A further study with increased range of parameters for larger  $\tau$  and smaller  $\omega$  values is necessary.

### 3.6 Discussion and conclusions

In this chapter, we examined time-dependent effects in the public goods game by adding an additional strategy, the punisher, among cooperators and defectors. Two forms of punishment, peer and pool punishment, were analyzed. While both punishment strategies yielded similar outcomes, peer punishment promoted cooperation more effectively. This greater efficacy comes from the fact that defectors are penalized multiple times for the same infraction, leading to a cumulative punishment effect.

To explore the dynamical nature of the game, we introduced a time-dependent enhancement factor  $r$  and a time delay  $\tau$  in the application of punishment to defectors. Both mechanisms were found to have a negative effect in cooperation. Large oscillation amplitudes led to an increase in the population of defectors, as oscillations in productivity disrupted the stability of cooperative behavior. For high oscillation frequencies, the system's behavior resembled that of a non-oscillatory scenario, suggesting that rapid oscillations, similar to noise, have minimal impact.

When a time delay in punishment was introduced, the effectiveness of punishment diminished, resulting in higher levels of defection. Thus, delayed punishment also impairs cooperation.

## Bibliography

- [1] L. A. Dugatkin, Cooperation in animals: An evolutionary overview Biol. and Philos. **17**, 459–476, (2002). <https://doi.org/10.1023/A:1020573415343>
- [2] P. Kollack, Social dilemmas: the anatomy of cooperation, Annu. Rev. Sociol. **24**, 183-214 (1998). <https://doi.org/10.1146/annurev.soc.24.1.183>
- [3] R. Schaap and Andries Richter, Overcapitalization and social norms of cooperation in a small-scale fishery, Ecol. Econ. **166**, 10643 (2019). <https://doi.org/10.1016/j.ecolecon.2019.106438>
- [4] M. Jusup, P. Holme, K. Kanazawa, M. Takayasu, B. Podobnik, L. Wang, W. Luo, T. Klanjšček, J. Fan, S. Boccaletti, and M. Perc, Social physics. Phys. Rep. **948**, 1–148 (2022).  
<https://doi.org/10.1016/j.physrep.2021.10.005>

- [5] Z. Bi and H. Zhou, Optimal Cooperation-Trap Strategies for the Iterated Rock-Paper-Scissors Game, PLoS One **9**, e111278 (2014).  
<https://doi.org/10.1371/journal.pone.0111278>
- [6] G. Szabó and C. Töke, Evolutionary Prisoner's Dilemma Game on a Square Lattice, Phys. Rev. E **58**, 69 (1998).  
<https://doi.org/10.1103/PhysRevE.58.69>
- [7] M. Archetti and I. Scheuring, Game Theory of Public Goods in One-Shot Social Dilemmas Without Assortment, J. Theor. Biol. **299**, 9 (2012).  
<https://doi.org/10.1016/j.jtbi.2011.06.018>
- [8] T. Wu, F. Fu, and L. Wang, Partner Selections in Public Goods Games with Constant Group Size, Phys. Rev. E **80**, 026121 (2009).  
<https://doi.org/10.1103/PhysRevE.80.026121>
- [9] D. Helbing and W. Yu, The Outbreak of Cooperation among Success-Driven Individuals under Noisy Conditions, Proc. Natl. Acad. Sci. U.S.A. **106**, 3680 (2009). <https://doi.org/10.1073/pnas.0811503106>
- [10] Z. Wang, L. Wang, Z. Yin, and C. Xia, Inferring Reputation Promotes the Evolution of Cooperation in Spatial Social Dilemma Games, PLoS One **7**, e40218 (2012). <https://doi.org/10.1371/journal.pone.0040218>
- [11] Y. Li, S. Chen, and B. Niu, Reward Depending on Public Funds Stimulates Cooperation in Spatial Prisoner's Dilemma Games, Chaos Solitons Fractals **114**, 38 (2018). <https://doi.org/10.1016/j.chaos.2018.07.002>
- [12] T. Sasaki and S. Uchida, The Evolution of Cooperation by Social Exclusion, Proc. R. Soc. B **280**, 20122498 (2013).  
<https://doi.org/10.1098/rspb.2012.2498>
- [13] A. Szolnoki, G. Szabó, and M. Perc, Phase Diagrams for the Spatial Public Goods Game with Pool Punishment, Phys. Rev. E **83**, 036101 (2011).  
<https://doi.org/10.1103/PhysRevE.83.036101>
- [14] X. Sun, Y. Li, H. Kang, Y. Shen, J. Peng, H. Wang, and Q. Chen, Co-Evolution of Complex Network Public Goods Game under the Edges Rules, MDPI Entropy **22**, 199 (2020). <https://doi.org/10.3390/e22020199>
- [15] I. Waichman and L. Stenzel, When Punishment Strikes Late: The Effect of a Delay in Punishment and Punishment Feedback on Cooperation and Efficiency, J. Neurosci. Psychol. Econ. **12**, 1 (2019).  
<https://doi.org/10.1037/npe0000099>

- [16] A. Szolnoki, M. Perc, and G. Szabó, Topology-Independent Impact of Noise on Cooperation in Spatial Public Goods Games, *Phys. Rev. E* **80**, 056109 (2009). <https://doi.org/10.1103/PhysRevE.80.056109>

## Chapter 4

# Measuring complexity in evolutionary games with the Hamming distance

*"We should not judge people by their peak of excellence; but by the distance they have traveled from the point where they started."*

-Henry Ward Beecher

Physicists have studied evolutionary game dynamics for their complex dynamics. From simple rules, complex behavior can be observed emerging from population size. Here, we are interested in the complex pattern formation first discovered by May and Nowak in 1992 [1]. They investigated the prisoner's dilemma where players were organized in a square grid and each round played games with their immediate neighbors. Each player was set to be either a cooperator or a defector, which determined the payoff they got each game. Then, the payoff was summed through every game they played that round. After each round, the players adapted the strategy of the individual with better payoff. They observed complex patterns when plotting in different colors cooperators and defectors when considering a specific parameter region.

### 4.1 Introduction

One can easily see the difference between two configurations of cooperators and defectors by measuring the Hamming Distance. This measures the number of different elements between two groups of elements of same size. It was introduced by Richard Hamming in 1950 in [2]. Since then it has been very helpful for computer science

---

G. Alfaro and M. A. F. Sanjuán, Hamming distance as a measure of spatial chaos in evolutionary games, Phys. Rev. E **109**, 014203 (2024)  
<https://doi.org/10.1103/PhysRevE.109.014203>

and cryptography, but its uses are broader. The Hamming distance has also been used in the context of social games [3, 4].

Moreover in [5, 6], the Hamming distance is used to asses complexity on social games. The authors measure the difference between two close configurations of rock-paper-scissors models. The Hamming distance converged to a certain value. But the distance oscillated increasing amplitude towards zero or the population size when shifting a parameter that represents the mobility beyond a critical point. For this parameter regime one strategy will end up being the only one present, and by varying just one individual's strategy at the beginning, the outcome can change completely, changing which strategy ends up winning. This shows that the susceptibility to initial perturbations of the system.

Through this novel research, we measured the complexity of the patterns observed by May and Nowak in [1] through this same method, by analyzing the Hamming distance between two initially close configurations of players. We found that for the parameter region where May and Nowak found the complex patterns, the Hamming distance grew in time as a sigmoid until it reached a convergence value; and for the rest of values, the distance quickly decreased to zero. This shows that the divergence of the Hamming distance could be an indicator of complex behavior.

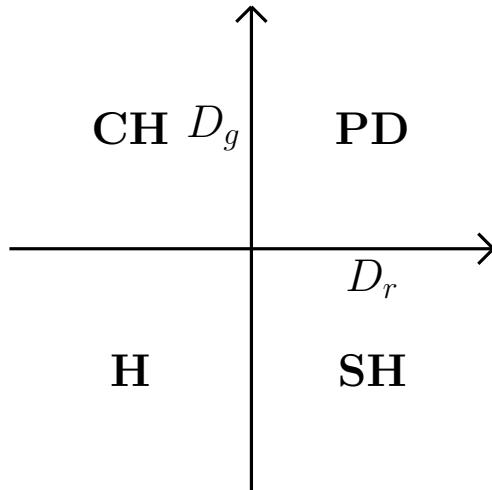
We also analyzed the divergence of the Hamming distance in a different game, the public goods game. Now, the Hamming distance grew until saturation for all cases. This may not indicate that all cases were complex, but instead be related to the fact that the evolutionary drive had random factors. The growth of the Hamming distance may be accounting for that randomness, even though we made sure to make all random decisions the same for both configurations. This means that our algorithm does not distinguish random processes from chaotic ones. However the growth was more rapid for some values which may indicate a higher complexity for those cases.

## 4.2 Prisoner's Dilemma

We developed a tool to asses the complexity of dynamic social games. The first thing to do is to check its validity with a game we know, presents complex behavior. This is the case for the game studied by May and Nowak in [1], the prisoner's dilemma, PD.

### 4.2.1 Model

The first game studied was the prisoner's dilemma. This is a game between two players that can be reproduced with the payoff matrix of Table 4.2.1 under  $T > R > P > S$ .



**Figure 4.1.** Diagram representing the four types of pairwise social games according to dilemma strength. CH stands for the Chicken game, PD for the Prisoner's Dilemma, SH stands for the Stag-hunt game and H stands for Harmony game.

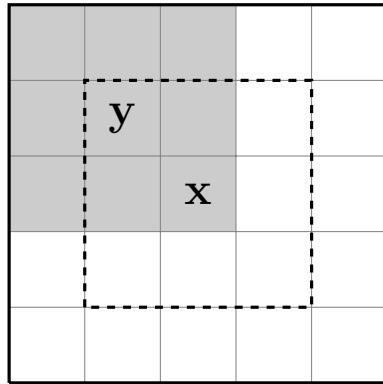
	C	D
C	R	S
D	T	P

**Table 4.1.** Payoff matrix for pairwise social games. C and D stands for the options of the players, either cooperate or defect. Players gain the reward payoff R if both cooperate, the punish reward P if both defect, and the temptation T or the sucker S payoff if one defects while the other defects.

We have set  $R = 1$ , the payoff if both players cooperate;  $P = 0$ , the payoff if both defect;  $T > 1$ , the payoff for a defector playing against a cooperator, which gains  $S = 0$ . Because the risk-averting dilemma strength results in  $D_r = P - S = 0$  this setup is a boundary game, not fully representing the PD. We have chosen this setup to reproduce the exact conditions observed by May and Nowak in [1].

The risk-adverting dilemma strength, together with the gamble-intending dilemma strength  $D_g = T - R$  composes the map of types of games pairwise social games represented in Fig. 4.1. These games are the harmony game, where  $D_r, D_g < 0$ , no dilemma is found and the logic option is to cooperate; the stag hunt game with,  $D_r > 0, D_g < 0$  where the logic option is to do the same as your opponent; the chicken game with  $D_r < 0, D_g > 0$ , where the logic option is to do the opposite than your opponent; and finally the prisoner's dilemma with  $D_r, D_g > 0$  where the logic option is to defect. However if both cooperated they would have better than both defecting but the temptation  $T$  is higher than the reward  $R$ , thus promoting the tragedy of the commons and receiving their punishment  $P$  [7].

Each player is put in a square grid with periodic boundaries, and plays eight games with their Moore neighbors, Fig. 4.2. Then they sum the payoff gained in



**Figure 4.2.** Agent  $x$ 's strategy is affected by the payoff of all agents in his Moore neighborhood (inside the dashed line); for example agent  $y$ , whose payoff depends of his own Moore neighborhood (shaded region). Therefore, the strategy of the agent  $x$  depends on all 25 agents in his second Moore neighborhood (inside the bold line).

all those games and compare the result with their neighbors and copy the strategy of the one with greater total payoff in that round. This is done for all players simultaneously and repeated through time.

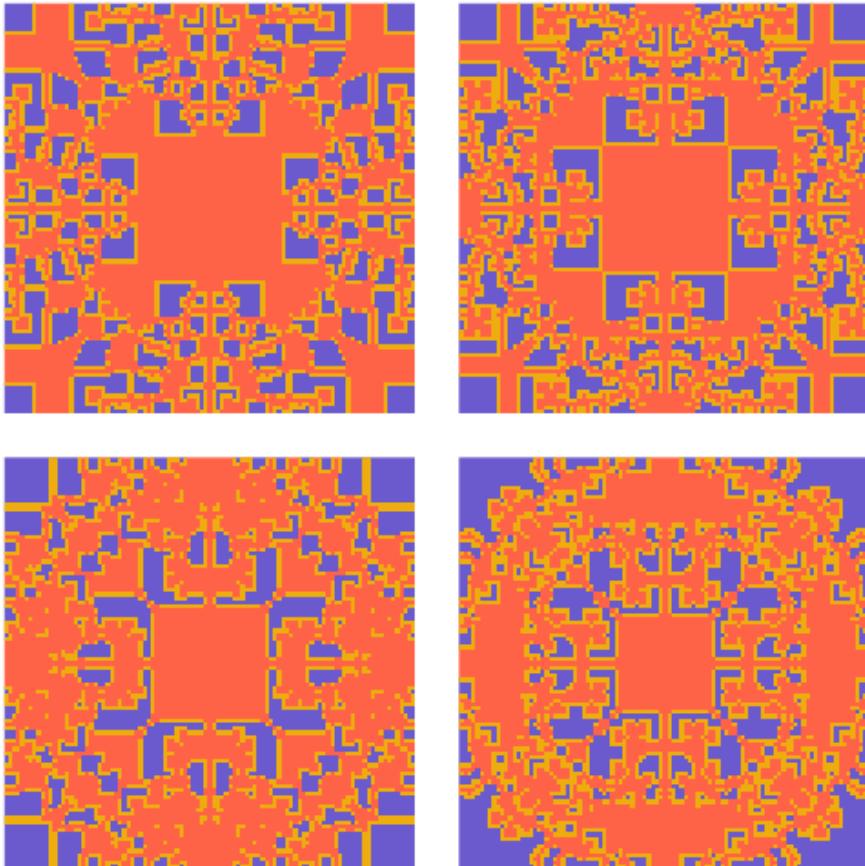
We plot cooperators in blue and defectors in red in Fig. 4.3. These plots show spatio-temporal chaotic patterns that evolve quickly. In [1] the authors claimed them to be fractal-like structures. However, as we can see on Fig. 4.4, when we augment the population size  $N$ , the patterns do not scale, and instead the clusters all size similarly.

The authors of [1] made an observation that the patterns seemed chaotic for these values of the temptation reward,  $1.8 < T < 2$ . We want a tool that quantifies the complexity of the pattern evolution that presents spatio-temporal chaos. We think that the algorithm we have developed, described in the following subsection, serves as an indicator for complex behavior and provides a value that quantifies how quickly the patterns evolve, which may be related to the Lyapunov time.

### 4.2.2 Hamming distance measure

The Hamming distance measures the difference between two objects as the number of different elements they contain. Think of two matrices or vectors of the same size, the Hamming distance between is the number of elements that differ from each position. Now we can represent the configuration of cooperators and defectors at each time as a matrix of 0's and 1's. Then we can calculate the Hamming Distance of two configurations by subtracting them and taking absolute values since they are binary matrices.

When we do so we observe that the distance quickly goes to zero for simulations in all parameter regimes except for values of the temptation reward in  $1.8 < T < 2$ , where it growths until it saturates at a given value that augments with population



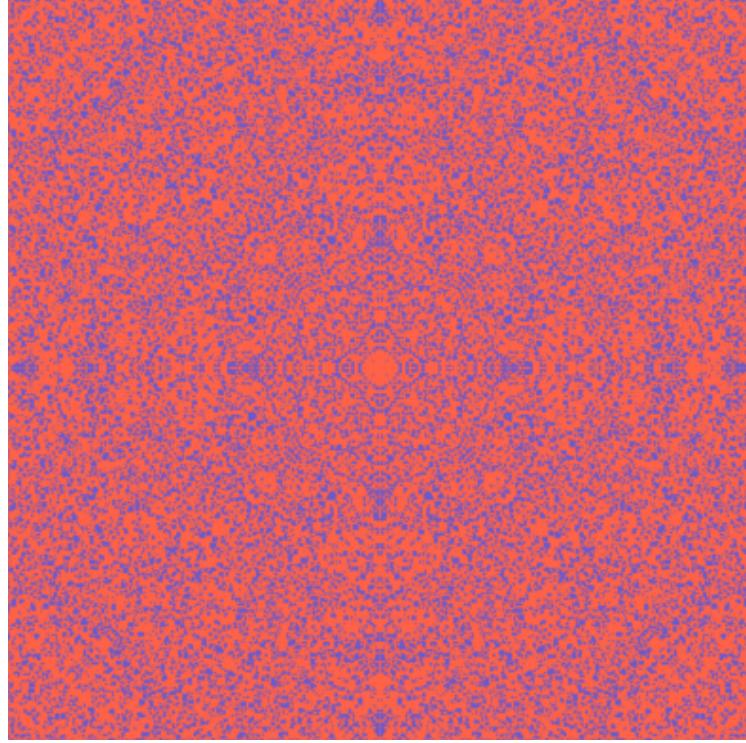
**Figure 4.3.** Snapshots of the public goods game at 4 different times of cooperators in blue, defectors in red, and agents that have changed from the previous configuration in yellow. The snapshots are taken at instants  $t_1 = 1000$ ,  $t_2 = 1002$ ,  $t_3 = 1004$  and  $t_4 = 1006$  after one defector is introduced at the center in a sea of cooperators in a grid of  $101 \times 101$  agents with periodic boundaries.

size as we can see in Fig. 4.5. We can analytically calculate this value, since it is the statistical Hamming distance, that two random configurations of 0's and 1's with certain probabilities would have. This value is proportional to the population size, and is calculated as:

$$H_{stat}(t) = N_C(t)p_D(t) + N_D(t)p_C(t), \quad (4.1)$$

where  $N_C$  and  $N_D$  indicate the number of cooperators and defectors; and  $p_C$  and  $p_D$ , the proportion of them. Since the proportion of cooperators and defectors is the number of them divided by the total population, we find that the proportionality constant for the saturation values of the Hamming distance is

$$h_{stat}(t) = p_C(t)p_D(t) + p_D(t)p_C(t), \quad (4.2)$$



**Figure 4.4.** Cooperators and defectors in blue and red, respectively.

which, in the case of binary strategies, it can be simplified as:

$$h_{stat}(t) = 2p_C(t)(1 - p_C(t)). \quad (4.3)$$

This value could be time dependent, but since the system is stable, the proportions of cooperators and defectors follow the Nash equilibrium and depend only on the values of  $T$ .

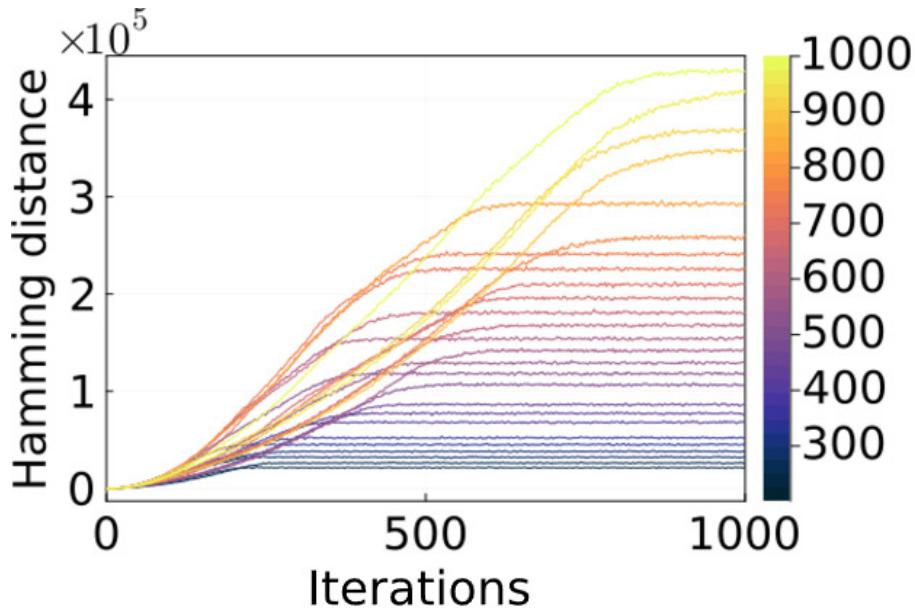
Then, if we normalize the Hamming distance from Fig. 4.5, dividing it by  $H_{stat} = h_{stat} * N$ , we get the normalized Hamming distance in Fig. 4.6. This figure shows sigmoids with different growth rate. Larger population sizes take longer to reach the saturation value.

We have fitted this curves to the Weibull "stretched exponential" function [8]

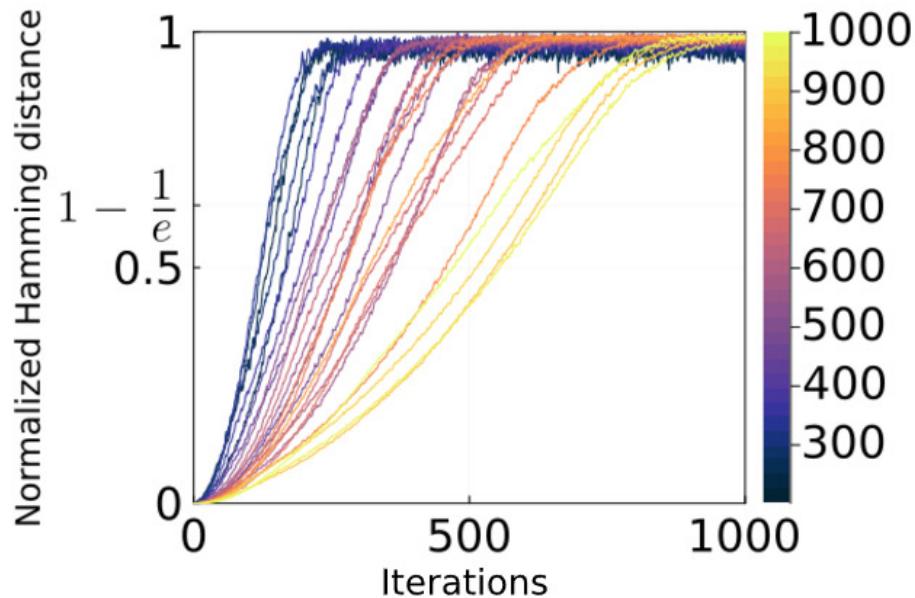
$$F(t; k, a) = 1 - e^{-(t/a)^k}, \quad (4.4)$$

where  $a$  is the timescale of growth and  $k$  tells how abrupt the growth is. We chose this function because it is the most simple sigmoid function that values zero at the origin and converge to one, having only two relevant parameters.

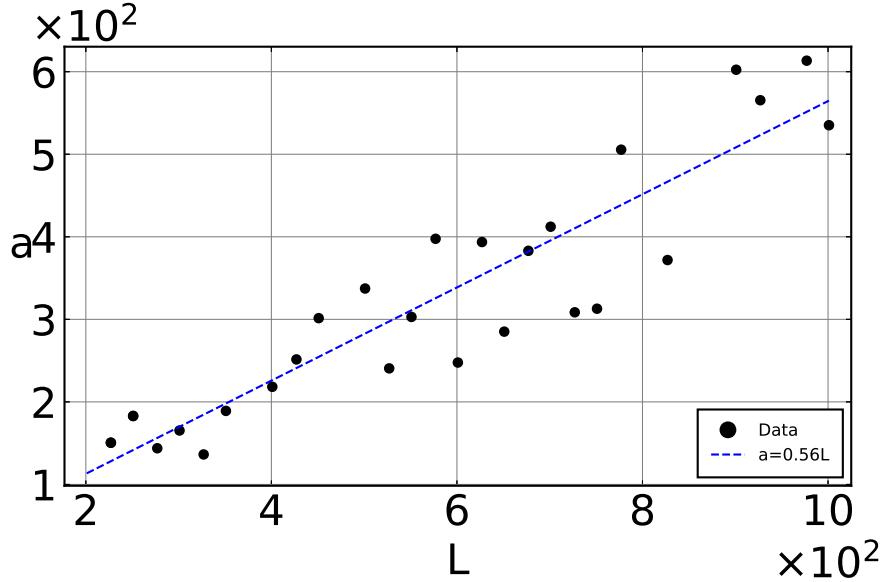
The value of  $a$  can be seen as something similar to the Lyapunov time, and as seen on Fig. 4.7, it is proportional to the grid size  $L$ , with a slope of  $0.56 \pm 0.05$  and



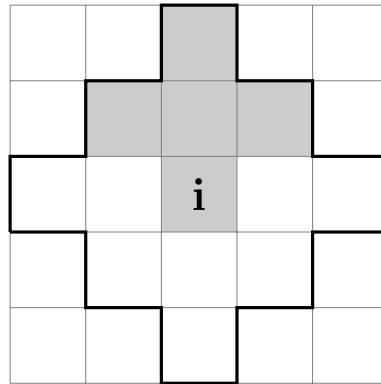
**Figure 4.5.** Hamming distance of the two solutions of the prisoner's dilemma versus time. Different colors represent different values of the grid size  $L$ . The larger the grid size, the longer it takes for the Hamming distance to reach the saturation value  $H_{stat}$ .



**Figure 4.6.** Normalized Hamming distance of the two solutions of the prisoner's dilemma versus time. Multiple curves are shown with different colors, representing the different grid size  $L$  values. The curves grow in a sigmoid-like curve towards one. They are normalized to the statistical Hamming distance which depends on  $L$ . The larger  $L$  is, the longer it takes for the normalized Hamming distance to reach 1.



**Figure 4.7.** Parameter  $a$  from the Weibull “stretched exponential” function  $F(t; k, a) = 1 - e^{-(t/a)^k}$  fitted to the normalized Hamming distance of the solutions for the prisoner’s dilemma versus grid size  $L$ . It shows a linear regression (dashed blue line) where  $a$  grows proportionally to  $L$ .



**Figure 4.8.** Von Neumann neighborhood at a distance 2 of agent  $i$ . The individual  $i$  plays 5 games with agents in cross-like patterns like the one shaded.

a negligible intercept. Note that the population size is  $N = L^2$ . The maximum slope is 0.5, since at maximum velocity any change in the agent’s strategy propagates to two agents away as seen on Fig. 4.2. The measured value of the slope is near the maximum which indicates a great propagation of the mismatches, and that may be related to complex behavior. As for the values of  $k$ , every curve is around  $k = 2.5 \pm 0.4$ .

## 4.3 Public Goods Game

Now that we have checked that our algorithm can asses the complexity in the PD, we want to measure weather the game we studied in the previous chapter was chaotic. The public goods game, PGG.

### 4.3.1 Model

This time, to make the model simpler we only allow the cooperate and defect strategy, and leave out the punish strategy. The rest of the model we leave the same. That is, agents play 5 games along 4 other agents among their second Von Neumann neighbors, Fig. 4.8 and their payoff depends whether they are cooperators,  $\Pi_C$ , or defectors,  $\Pi_D$  and the number of cooperators  $N_C^g$  that there are in their group.

$$\begin{aligned}\Pi_C^g &= \frac{r}{G} \cdot N_C^g - 1 \\ \Pi_D^g &= \frac{r}{G} \cdot N_C^g\end{aligned}\tag{4.5}$$

Here  $r$  is a parameter that rewards cooperation and  $G = 5$  is the size of the group.

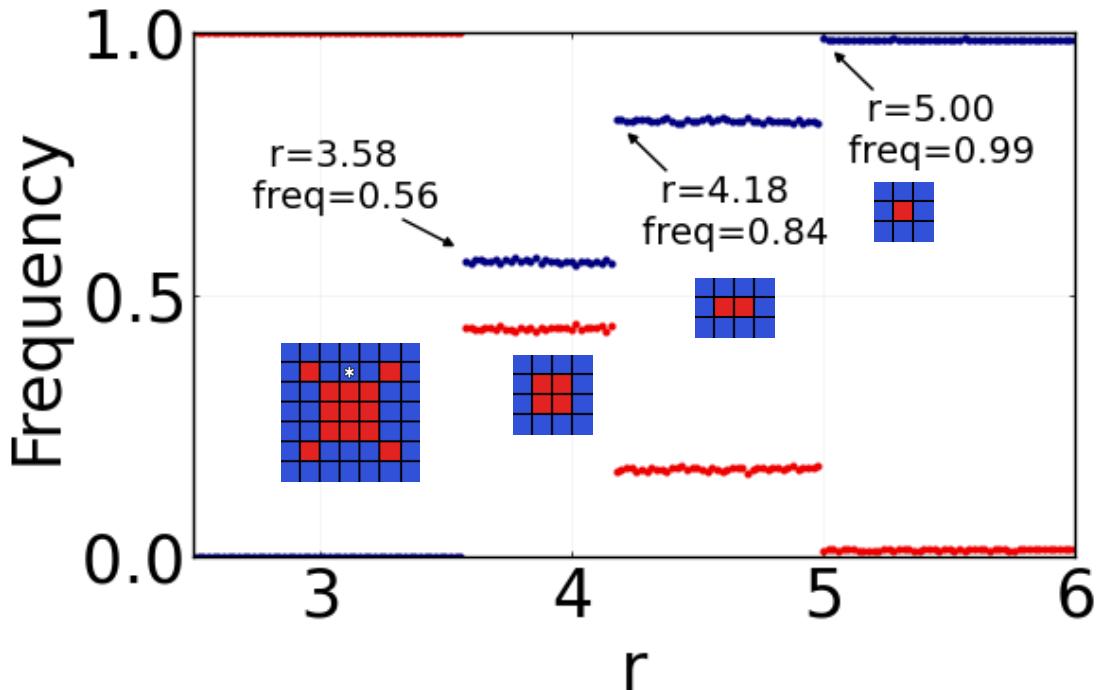
Then, the accumulated payoff  $\Pi = \sum_g^G \Pi^g$  determines the fitness of each agent in an evolutionary model where a random agent chooses whether to adopt or not the strategy of a random neighbor. This time we wanted to reduce the stochasticity. Therefore, this time, if the accumulated payoff of the agent's neighbor is greater, it adopts its strategy. Whereas if it is lesser, it does nothing. Nonetheless we still maintained the random election of the agent that will change its strategy to maintain the same game as previous chapter, where the parameter of noise  $K$  is in the limit of  $K \rightarrow 0$ .

Under these rules, we observe that after a transitory period, the average population stabilizes with cooperator and defector proportions given by Fig. 4.9. It shows that the proportion only changes at some values of  $r$  and then stays constant. We are able to predict the values of  $r$  where these shifts appear analytically. We just have to compare the payoff of the defectors and the cooperators that surround them in configurations that are clusters of defectors of different sizes that are shown in the figure.

As an example here we have the payoff of the white marked cooperator in the first configuration and the defector below it.

$$\begin{aligned}\Pi_C &= \frac{1}{5}(5r + 4r + 2 \times 3r + r) - 5 = \frac{16}{5}r - 5 \\ \Pi_D &= \frac{1}{5}(4r + 2 \times 2r + r + 0) = \frac{9}{5}r\end{aligned}\tag{4.6}$$

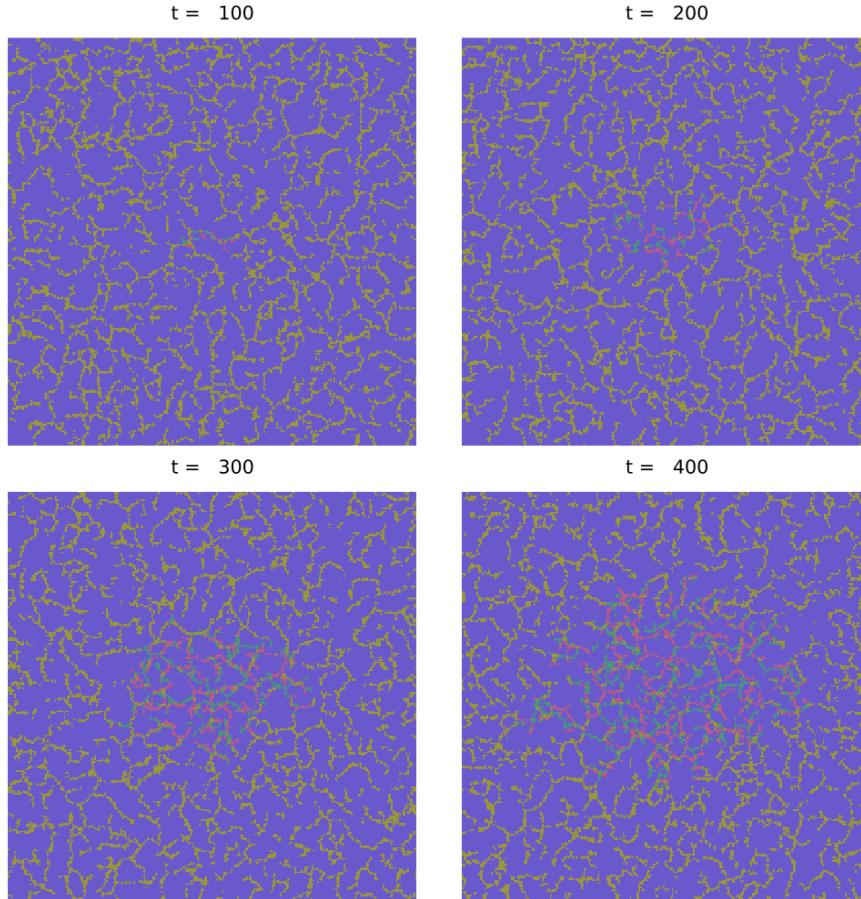
When we equal these two payoffs we get that the value of  $r$  where we expect a shift is  $r = 25/7 \approx 3.57$ , which is indeed where we find the shift. The next shifts



**Figure 4.9.** Proportion of cooperators (in blue) and defectors (in red) after running a simulation of the public goods game starting with a random strategy in each of the  $N = 90000$  agents with a relaxation time of 2000 generations. We can see that the proportion changes only at some values of  $r$ . These values have been calculated analytically. Each shift corresponds to a change in whether the surrounding cooperators have greater or lesser payoff than the surrounded defectors in the configurations we show. For the first configuration this holds strictly for the white-marked cooperator and the defector below it at the first shift at  $r = 25/7 \approx 3.57$ .

can be calculated similarly with the rest of the configurations and give values of  $25/6 = 4.1\hat{6}$ ,  $25/5 = 5$  and  $25/3 = 8.\hat{3}$ .

In the PGG cooperation is a Nash equilibrium for  $r \geq 5$ . At the Nash equilibrium the payoff cannot be increased when changing strategy given that the rest of agents don't change theirs. But we observe that for  $5 < r < 8.\hat{3}$  cooperation is not the only strategy present. This is because the payoff of each agent is affected by the strategies of its neighbors. Therefore, a defector surrounded by cooperators has greater payoff than them, even though it would have greater payoff if it were a cooperator. This means that being at a Nash equilibrium does not eliminate all defectors. We observe a last shift at  $r = 8.\hat{3}$ , beyond this last shift defection is not viable. We don't show this shift in Fig. 4.9 because it can hardly be appreciated at the current scale (from a cooperator proportion of 0.99 to 1). What's more, since the election of the agent



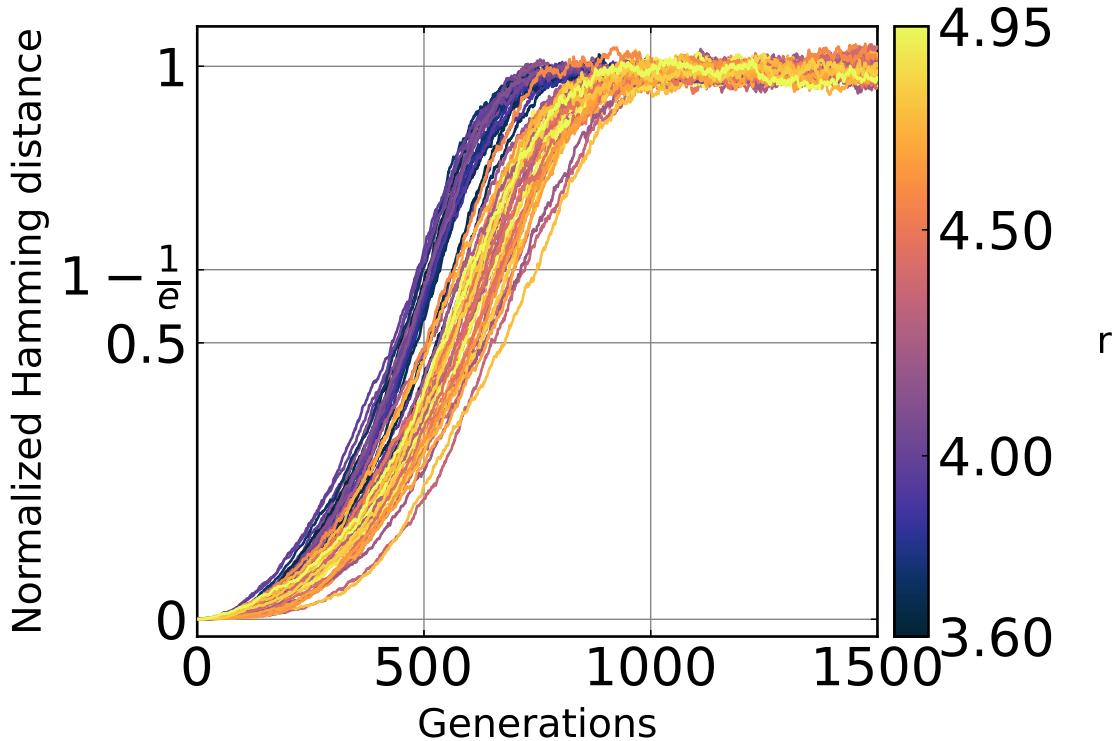
**Figure 4.10.** Snapshots at 4 different times of both configurations (original, and copy with altered agent at  $t = 0$  at the center) of the public goods game. Cooperators are plotted in blue, defectors in red for one configuration and in green for the other. Both configurations are plotted one on top of the other with a transparent effect. Thanks to this we can see that the mismatches propagate at average constant velocity and expand radially.

that changes its strategy is at random, the computational times for all defectors to adopt the cooperation strategy are large.

### 4.3.2 Hamming distance measure

To asses the complexity of our PGG we measured the Hamming distance of two initially close configurations differing of only one agent. Because the election of the agent that adopt a new strategy in the evolutionary model is chosen at random, we have to make sure every random interaction is chosen the same through both simulations.

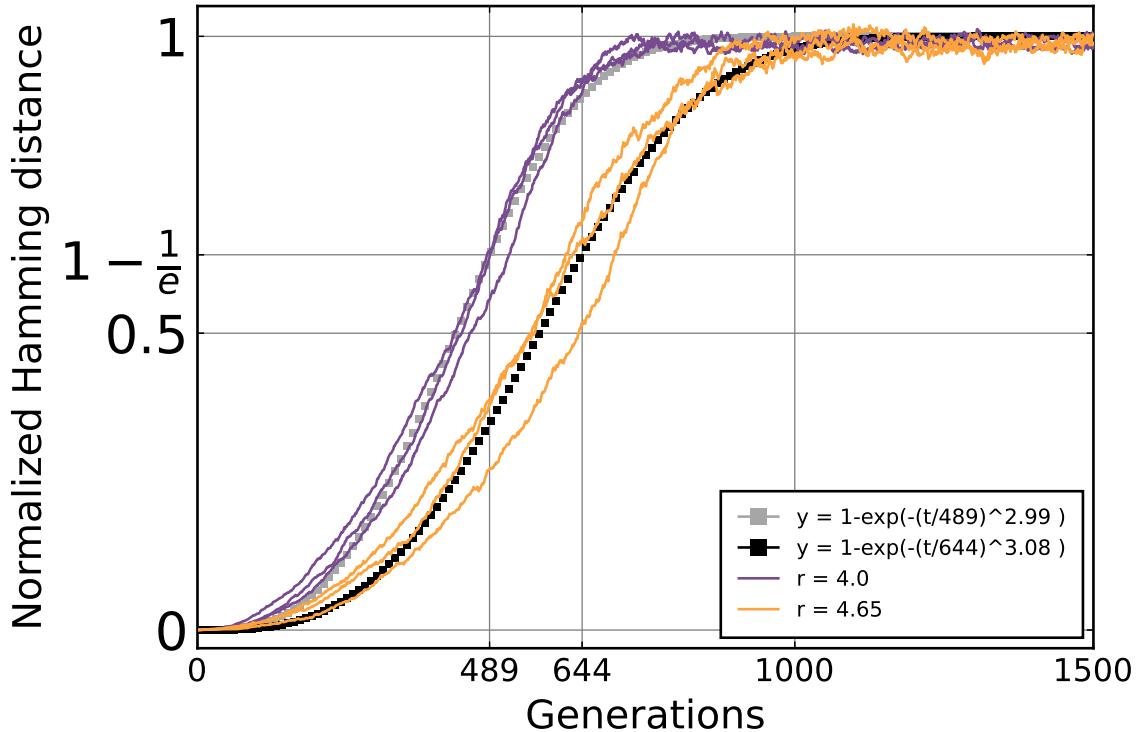
We can see how mismatches propagate in Fig. 4.10. For each snapshot both configuration, the original and the copy with a different agent at the center, are



**Figure 4.11.** Normalized Hamming distance between the solutions for the public goods game versus time. Multiple curves are shown with different colors, representing the different  $r$  values. The curves grow in a sigmoid-like curve towards one. They are normalized to the statistical Hamming distance, which depends on  $r$ , so curves ranging from  $25/7 \leq r < 25/6$  are normalized to a different value than those at  $25/6 \leq r < 5$ . There is a distinction between the two regimes, the curves of the first one (blue ones) reach its midpoint earlier than the ones in the second regime (orange ones).

plotted on the same space in two layers with one layer being slightly transparent. The defectors in one configuration are plotted red and in the other, green. This allows us to watch how the mismatches propagate. They do so radially and at constant velocity on average, but following the branches of defectors when watched closely.

The result was, as seen in Fig. 4.11 that the Normalized Hamming distance grew as a sigmoid like in the previous section for values of the parameter  $r$  between  $25/7 \approx 3.57$  and  $8.3$ . These are all the values where there are present cooperators and defectors . We cannot say that this is an indicator of chaos this time, because probably, what the algorithm is assessing is the randomness in the election of agent to change its strategy. Nonetheless, when we fit the curves to the Weibull “stretched exponential” function in Fig. 4.12, we get lower values of  $a$  for curves in the first regime,  $25/7 \leq r < 25/6$ . This means that the system is more susceptible to changes in the initial conditions. For values of  $5 > r > 8.3$ . we get a very large value



**Figure 4.12.** Normalized Hamming distance between the solutions for the public goods game versus time. Different colors represent two different  $r$  values. They are normalized to the statistical Hamming distance, which depends on  $r$ , so the two curves are normalized to a different value. The Weibull “stretched exponential” function  $F(t; k, a) = 1 - e^{-(t/a)^k}$  is fitted to the curves and we observe different values of  $a$ , which can be seen as similar to the Lyapùnوف time. The  $r = 4$  curves (blue and left ones), representing the regime  $25/7 \leq r < 25/6$  has a lower value of  $a$  than the curves with  $r = 4.65$  (orange and right ones). This indicates that, for  $25/7 \leq r < 25/6$ , the system is more sensitive to initial conditions.

of  $a = 3685$ .

## 4.4 Discussion and conclusions

We have developed an algorithm to measure the divergence of two binary configurations of cooperators and defectors in the prisoner’s dilemma and the public goods game using the Hamming distance. We observed that if the game has no random elements, the divergence can be linked to complex behavior. The Hamming distance grew as a sigmoid function in time for a system that was clear, had a complex behavior and was zero when not.

We also calculated a measure that could correlate to a Lyapunov time. The measure, which indicated approximately the midpoint of the sigmoids, was proportional

to the grid size and the constant of proportionality was near its maximum for the prisoner's dilemma with no random elements.

For the public goods game with some random elements, all games studied presented a divergence of the Hamming distance. Nonetheless, the divergence was slower, it took longer to reach the midpoint, for some games, which had a sparse population of defectors and a great number of cooperators.

Even though our studied was focused on two particular cases, we are sure the algorithm is adequate to measure divergence and stability in many systems and geometries due to the simplicity of its implementation. In fact, in the next chapter we explain how we used the same tool to make a classification of elementary cellular automata based on complexity.

## Bibliography

- [1] M. A. Nowak and R. M. May, Evolutionary games and spatial chaos. *Nature* **359**, 826–829 (1992). <https://doi.org/10.1038/359826a0>
- [2] R. Hamming, Error detecting and error correcting codes. *Bell Syst. Tech. J.* **29**, 147–160 (1950). <https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>
- [3] R. D'holst and G. J. Rodgers, The hamming distance in the minority game. *Physica A* **270**, 514–525 (1999).  
[https://doi.org/10.1016/S0378-4371\(99\)00211-3](https://doi.org/10.1016/S0378-4371(99)00211-3)
- [4] J.-W. Kim, A tag-based evolutionary prisoner's dilemma game on networks with different topologies. *JASSS*, **13**, 2 (2010).  
<https://doi.org/10.18564/jasss.1584>
- [5] D. Bazeia, M. B. P. N. Pereira, A. V. Brito, B. F. Oliveira, and J. G. G. S. Ramos, A novel procedure for the identification of chaos in complex biological systems. *Sci. Rep.* **7**, 44900 (2017). <https://doi.org/10.1038/srep44900>
- [6] D. Bazeia, J. Menezes, B. F. De Oliveira, and J. G. G. S. Ramos, Hamming distance and mobility behavior in generalized rock-paper-scissors models. *EPL* **119**, 58003 (2017). <https://doi.org/10.1209/0295-5075/119/58003>
- [7] G. Hardin, The tragedy of the commons: the population problem has no technical solution; it requires a fundamental extension in morality. *Science* **162**, 1243–1248 (1968). <https://doi.org/10.1126/science.162.3859.1243>
- [8] W. Weibull, A statistical distribution function of wide applicability. *J. Appl. Mech.* **18**, 293–297 (1951). <https://hal.science/hal-03112318v1>

## Chapter 5

# Classification of Cellular Automata based on the Hamming distance

*“Automation is good, so long as you know exactly where to put the machine.”*

-Eliyahu Goldratt

In the previous chapter we reviewed evolutionary social games and configured them for agents to play in a square lattice with their neighbors. We calculated the payoff of each individual according to some rules and then they adopted the strategy of one of its neighbors depending on the payoff. Since the payoff of the agents depends only on the state of their neighbors, if we do not allow error of decision (i.e. with the decision-making Rule- $K \rightarrow 0$ ), then each combination of neighboring agents will produce unambiguously the updated state of the central agent at the next iteration. Therefore, if we have the prisoner’s dilemma with Moore neighborhood as in last chapter, the state of an agent depends only on the state of all its 24 neighbors and its own.

One could obtain all of the possible results of the game with the different combinations of these 25 agents. Given that each agent can only be either a cooperator or a defector, we get  $2^{25}$  different situations that completely define the system. If now we would get the state of the current cell at the next iteration for each of these situations, then we would have formalized the game as a cellular automaton. The rules of this automaton would be each of the combinations of the neighborhood cells states. That would be  $2^{25}$  rules to follow, which would make a very complicated

---

G. Alfaro and M. A. F. Sanjuán, Classification of cellular automata based on the Hamming distance, Chaos **34**, 083129 (2024).  
<https://doi.org/10.1063/5.0227349>

cellular automaton, although one could get the same behavior with a lower number of rules by intelligently making more complicated rules.

For this study we use the algorithm developed in the investigation of the previous chapter, but this time with implement it on elementary cellular automata, the most simple cellular automata.

But, what exactly is a cellular automaton?

## 5.1 Introduction

A cellular automaton, or CA, is a collection of rules that determine the dependence of cells' states to the state of neighboring cells. Usually, CA do not have a rule for every combination of states of cells, but have a few rules that univocally tells each cell how to update. Take for example Conway's Game of Life. With only four rules that check whether a certain number of neighboring cells are alive or dead, (1 or 0), the state of each cell is determined. With only these four rules the system presents a broad complexity, with complex patterns arising from simple initial configurations.

There are different methods to increase the complexity of cellular automata. One could let the cells have more than two states, like the first ever studied cellular automata, which was developed in the late 1950s by John von Neumann [1] and had 29 states per cell. It was developed as a model for discrete liquid motion, but he was also interested in the idea of self-replication and how machines could replicate themselves. With this automata he made a universal constructor. Additionally increasing the spatial dimensions of the system would complicate the dynamics. One could also let the automata have memory by allowing the rules to depend on the state of cells at previous iterations.

This complexity arising from simple rules has given rise to the thorough research of CA in many ambitis of science. For example, in physics they have been used to model the circulation of vehicles, [2], and pedestrians [3]. In [4] a model for discrete lattice gases are cellular automata. The authors of [5] use a set of cellular automata to simulate avalanches in sand piles. Cellular automata has also been used in quantum mechanics, [6]. They have also been studied in engineering to implement logical devices as in [7]. In cryptography they are widely used as in [8, 9]. Multiple applications have also been found in biology, [10, 11, 12]

To measure this complexity we will use, again, the Hamming distance, but in order of understanding better the limitations of this distance for measuring complexity, here we study the simpler case of CA, the Elementary Cellular Automata, or ECA.

### 5.1.1 Elementary Cellular Automata

ECA are one-dimensional, binary CA that depend only on the first neighbors. Therefore all the different results of all the combinations of three cells which can have only two states are  $2^{2^3} = 256$  which is the number of different ECA rules. Due to sym-

			
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

**Table 5.1.** Truth table of ECA Rule-90. At the right, the state of the central cell at the next iteration for each of the configurations gives the name to the rule. From top to bottom:  $0 \times 1 + 1 \times 2 + 0 \times 4 + 1 \times 8 + 1 \times 16 + 0 \times 32 + 1 \times 64 + 0 \times 128 = 90$

metry reasons, of all of these rules only 88 are inequivalent. The rest produce either the same result or the complementary than one of those 88 rules.

ECA rules are recognized by a number from 0 to 255. The rules are named as follows. Take each combination of 0s and 1s of the three cells. Order them ascendingly, by the value of the binary number if the leftmost cell represents the value of  $2^2$  the central cell the value of  $2^1$  and the rightmost,  $2^0$ . Now, the state of the cell for the next iteration of the eight combinations given each rule forms a binary number. This number, translated to decimal, names the rule. This is exemplified in Table 5.1.

From the mid 1980s Stephen Wolfram classified all CA in four classes [13]. This phenomenological classification studied the behaviour of all the rules of Elementary Cellular Automata with various initial conditions. Ordered by increasing complexity, the classes are:

- (i) Class-1: The automata in this class quickly evolve to a homogeneous and stable state.
- (ii) Class-2: In this class the automata quickly evolve to a periodic state.
- (iii) Class-3: The patterns that form in the evolution of the automata in this class are pseudo-random or chaotic and never repeat, except for Poincaré recurring

times. That is, for automata with finite population size  $L$  there is only  $2^L$  possible states.

- (iv) Class-4: The automata in this class are complex in the sense that there are regions where the evolution is periodic that mix with regions that behave like those at Class-3. Since these regions are in contact they affect each other in a very long complex evolution that may end after a long time into a periodic state like automata at Class-2.

We can observe this graphically in Fig. 5.1.

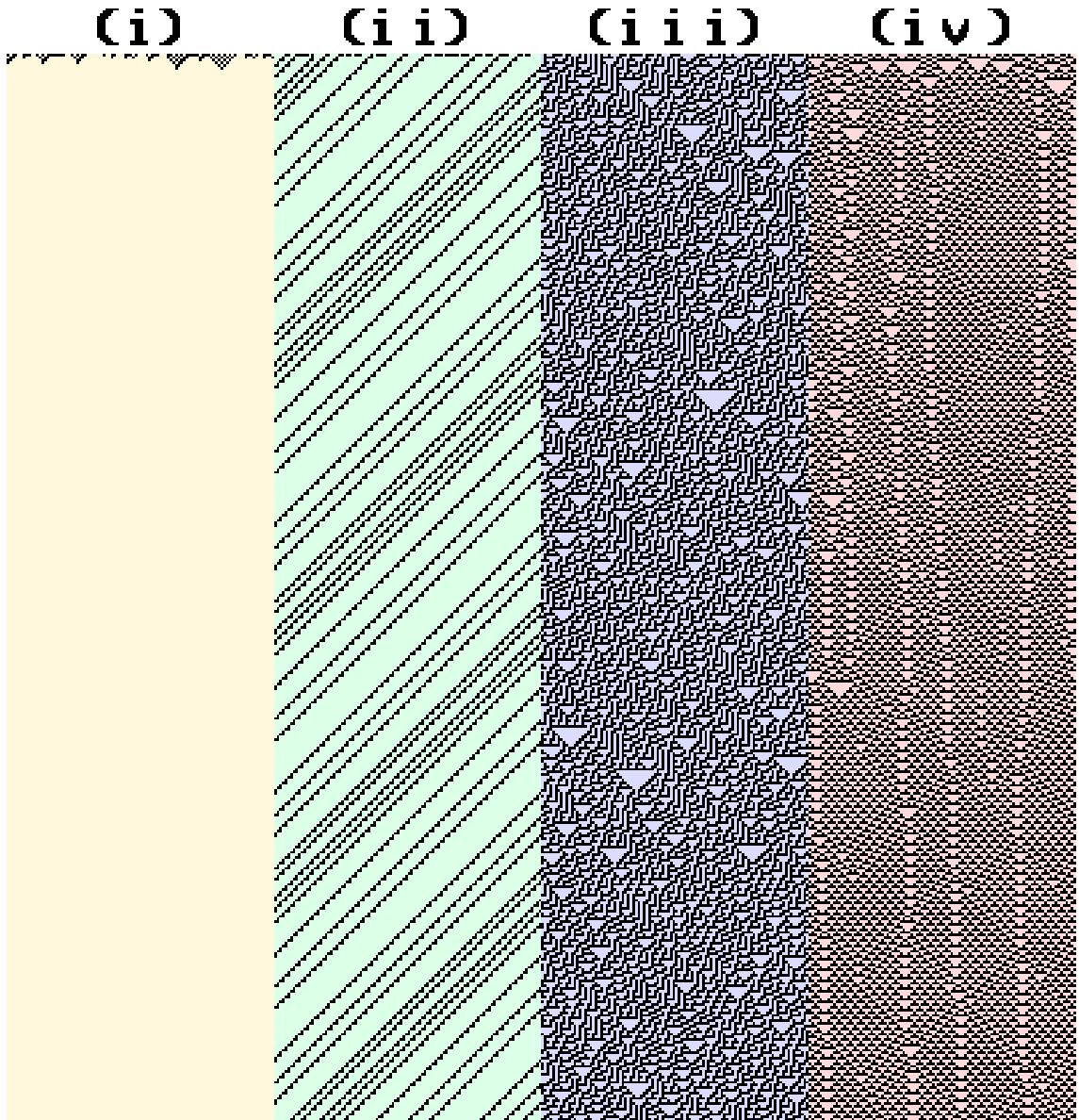
Different initial conditions may evolve to different behaviour, and be classified as a different class, so a sufficient number of initial conditions should be necessary for a cellular automaton to be classified at each class.

Class 4 is the most interesting ones, with examples like ECA Rule-110 and Conway's Game of Life, which both have been proven to be capable of universal computation [14].

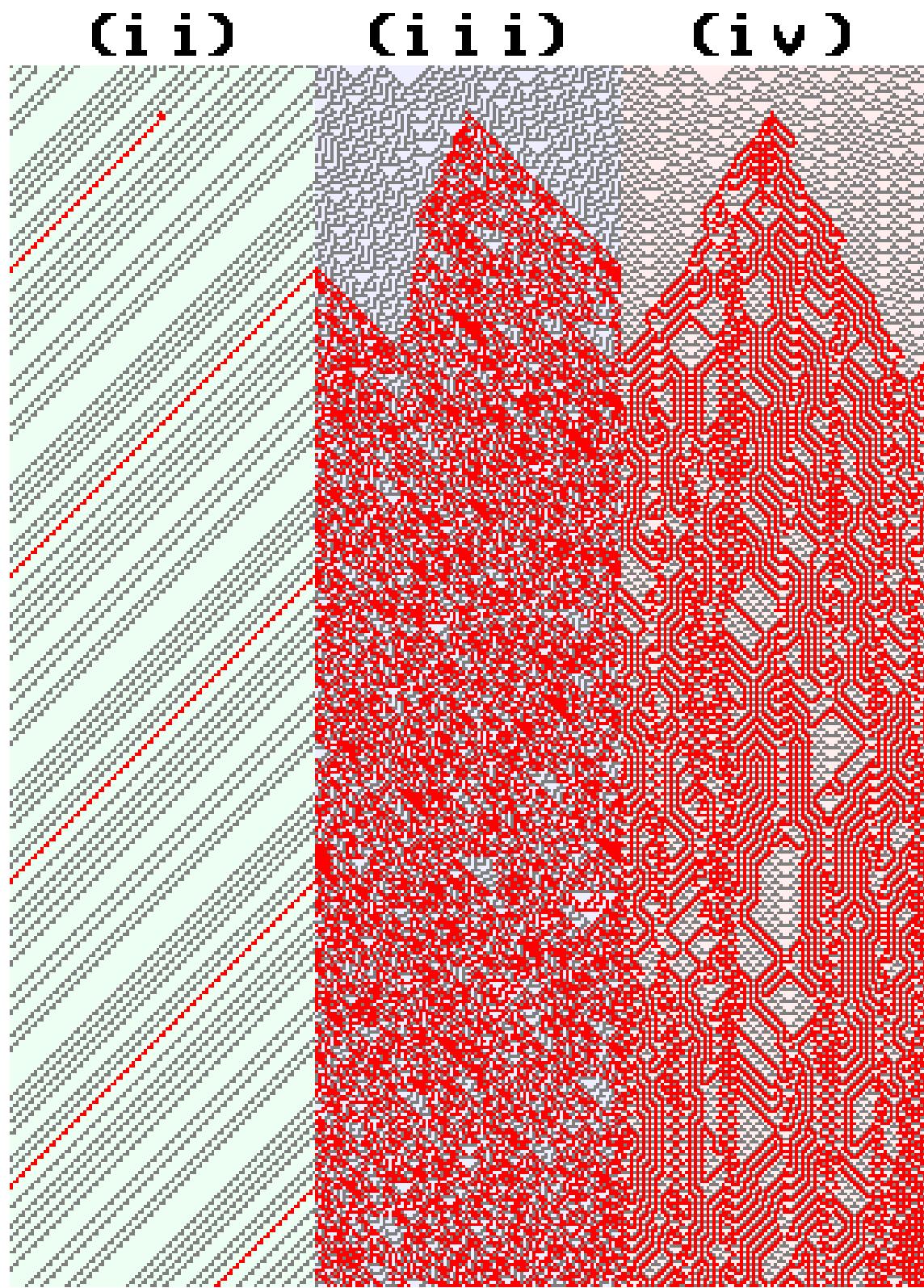
In *WolframAlpha*, a computational search engine with knowledge from experts in different fields, the ECA rules are indexed and each one is given a class. This data is collected in Table 5.2 along our own classification which we will discuss in next.

**Table 5.2.** ECA rules and classification according to Wolfram, and subclasses according to Hamming distance time serie analysis.

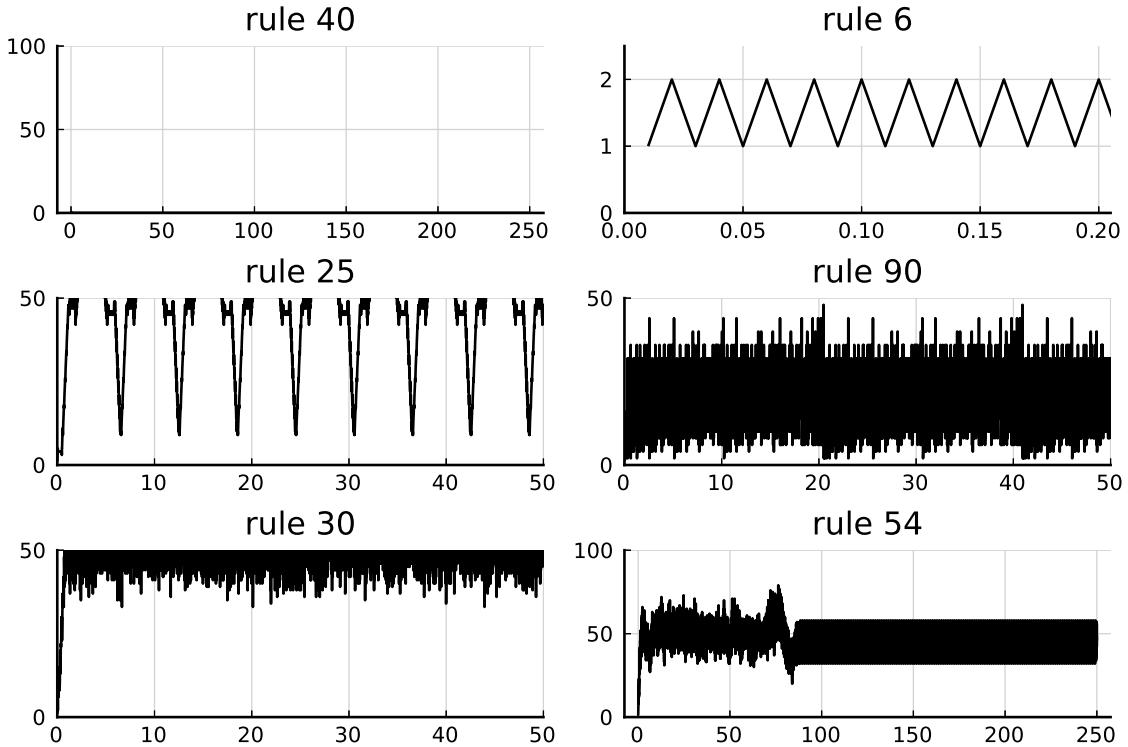
Rule	Equivalent	Class	Subclass	Rule	Equivalent	Class	Subclass
0	255	1	1	56	98, 185, 227	2	LP
1	127	2	LP	57	99	2	LP
2	16, 191, 247	2	LP	58	114, 163, 177	2	LP
3	17, 63, 119	2	LP	60	102, 153, 195	3	U
4	223	2	LP	62	118, 131, 145	2	HP
5	95	2	LP	72	237	2	LP
6	20, 159, 215	2	LP	73	109	2	T
7	21, 31, 87	2	LP	74	88, 173, 229	2	LP
8	64, 239, 253	1	1	76	205	2	LP
9	65, 111, 125	2	LP	77	-	2	LP
10	80, 175, 245	2	LP	78	92, 141, 197	2	LP
11	47, 81, 117	2	LP	90	165	3	U
12	68, 207, 221	2	LP	94	133	2	LP
13	69, 79, 93	2	LP	104	233	2	LP
14	84, 143, 213	2	LP	105	-	3	U
15	85	2	LP	106	120, 169, 225	4	C
18	183	3	C	108	201	2	LP
19	55	2	LP	110	124, 137, 193	4	T
22	151	3	C	122	161	3	C
23	-	2	LP	126	129	3	C
24	66, 189, 231	2	LP	128	254	1	1
25	61, 67, 103	2	HP	130	144, 190, 246	2	LP
26	82, 167, 181	2	LP	132	222	2	LP
27	39, 53, 83	2	LP	134	148, 158, 214	2	LP
28	70, 157, 199	2	LP	136	192, 238, 252	1	1
29	71	2	LP	138	174, 208, 224	2	LP
30	86, 135, 149	3	C	140	196, 206, 220	2	LP
32	251	1	1	142	212	2	LP
33	123	2	LP	146	182	3	C
34	48, 187, 243	2	LP	150	-	3	U
35	49, 59, 115	2	LP	152	188, 194, 230	2	LP
36	213	2	LP	154	166, 180, 210	2	LP
37	91	2	LP	156	198	2	LP
38	52, 155, 211	2	LP	160	250	1	1
40	96, 235, 249	1	1	162	176, 186, 242	2	LP
41	97, 107, 121	4	T	164	218	2	LP
42	112, 171, 241	2	LP	168	224, 234, 248	1	1
43	113	2	LP	170	240	2	LP
44	100, 203, 217	2	LP	172	202, 216, 228	2	LP
45	75, 89, 101	3	C	178	-	2	LP
46	116, 139, 209	2	LP	184	226	2	LP
50	179	2	LP	200	236	2	LP
51	-	2	LP	204	-	2	LP
54	147	T		232	-	2	LP



**Figure 5.1.** Representation of the four classes of elementary cellular automata according to Wolfram. Different colors represent the state 0 at each class. ) Class-1 with Rule-40, the automaton quickly goes to a fixed state where all cells are 0. ) Class-2 with Rule-6, the automaton quickly evolves to a periodic state. ) Class-3 with Rule-30, the automaton results in chaotic patterns that does not repeat. ) Class-4 with Rule-54, different regions can be appreciated before a periodic state is found (for longer times) . The initial cells (top) have a 50% chance of being 0 or 1 and the automaton is iterated using a periodic boundary.



**Figure 5.2.** Difference Pattern in red for Class-2, Rule-6 (ii); in Class-3, Rule-30 (iii); and in Class-4, Rule-54 (iv).

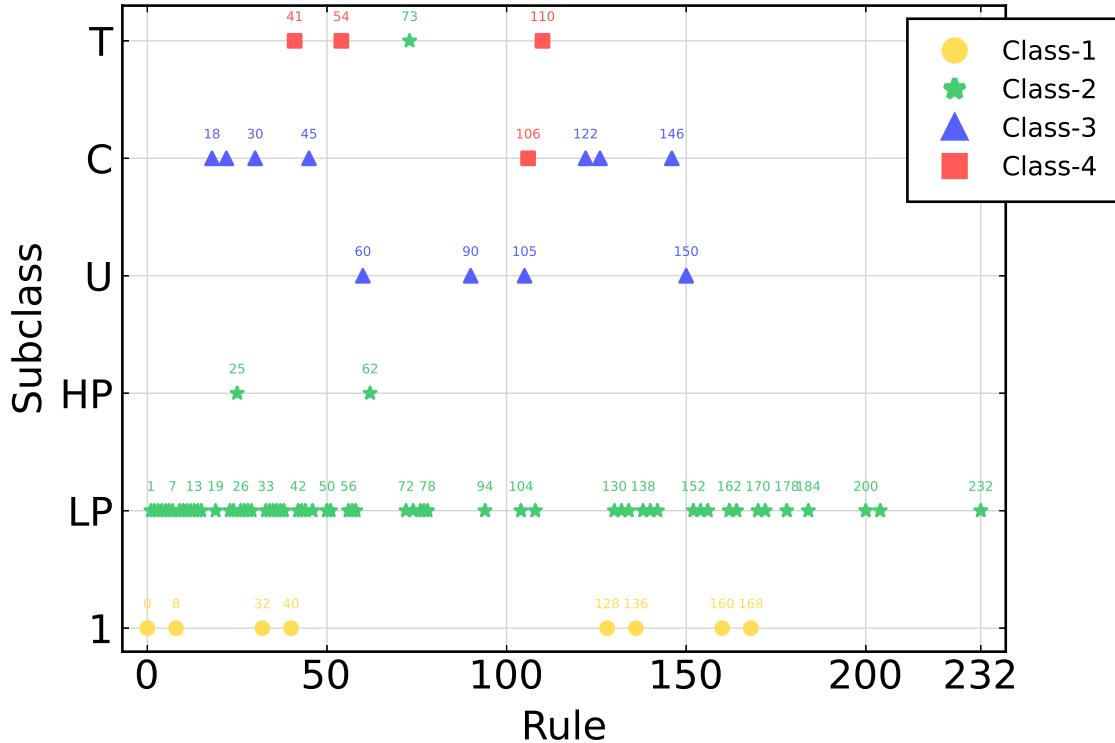


**Figure 5.3.** For 6 different rules belonging to: Class-1 (Rule-40), Class-2 subclass LP (Rule-6), Class-2 subclass HP (Rule-25), Class-3 subclass U (Rule-90), Class-3 subclass C (Rule-30) and Class-4 subclass T (Rule-54); we represent the Hamming distance over time. The horizontal axis is in units of  $L = 100$  iterations, which is the size of the population.

## 5.2 Hamming distance classification

A way to observe how susceptible a given rule is to minimal changes in the initial conditions are the difference patterns. Each class has characteristic differences when comparing its difference patterns. As we can see in Fig. 5.2 the difference pattern for Class-1 is null, periodic for Class-2, noisy for Class-3 and complex for Class-4. In our study we have focused our attention in the number of cells that are different at each iteration. This is, again, the Hamming distance, which we used in the previous chapter. Here we obtain the Hamming distance of two initially close configurations varying only in one cell versus time, and analyze the saturated time series instead of the growth.

We show the Hamming distance for 6 rules that present different behavior in Fig. 5.3. We have grouped together all rules that present the same behavior in a sub-classification of Wolfram's classification. The subclass column in Table 5.2 and Fig. 5.4 collects this information. Different initial conditions may produce a time series with different behavior. To help us to asses how complex can a rule get, we have classified the rule taking into account the time series with the most complex



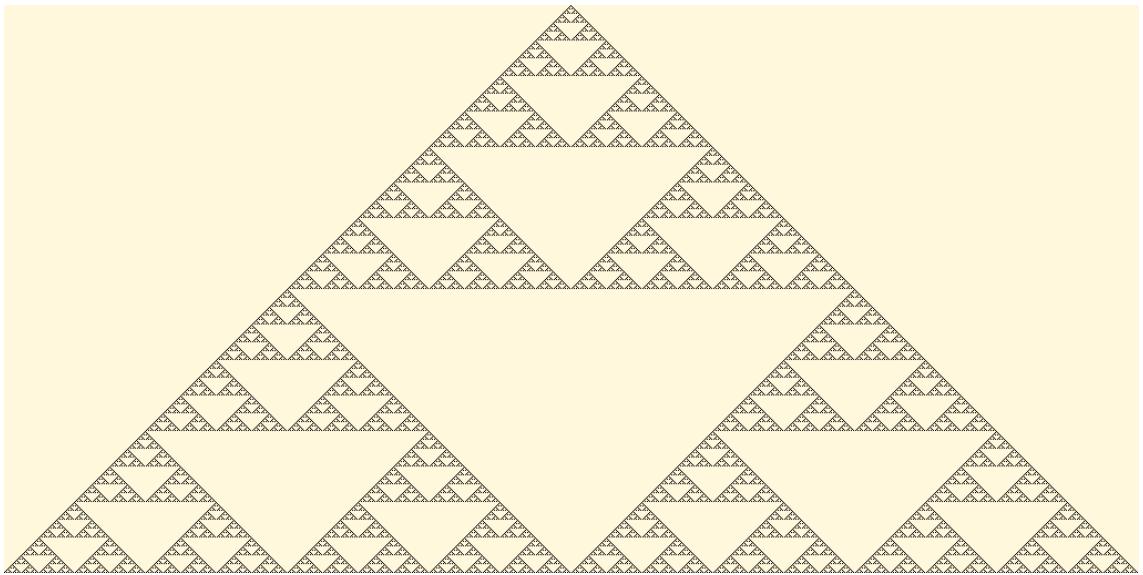
**Figure 5.4.** Hamming distance classification. Colors and symbols correspond to Wolfram’s classification, while position at the vertical axis correspond to the subclasses.

behaviour along 100 different computations.

For Wolfram Class-1, the Hamming distance time series at this class goes rapidly to 0.

For Class-2 the time series varies periodically or is constant. We have subclass LP for low periods of no more than 20 iterations, though most have period lesser than 5; and HP for high periods greater or around  $5L$  iterations. We can obtain the period o the series through its autocorrelation.

In Class-3 we have subclass U and C. Rules in subclass C give chaotic time series. Whereas those in subclass U have exactly the same time series for all initial conditions. The series is periodic, but with an extremely long period of exactly 2046 iterations. This is very close to a power of 2,  $2^{11} = 2048$ . In fact with different values of  $L$  one obtains that the periods of the Hamming distance are approximate to different powers of 2. What is more, if the populations size is set to a power of 2, the Hamming distance goes to 0 after some iterations. Looking at Fig. 5.5, where we represent the difference pattern for Rule-90, which it is the same than the pattern of evolution, we see that the patterns scale and repeat at periods that get multiplied by 2. This is due to the auto-similarity that characterizes this patterns as fractal. As we explain below, all rules in subClass-U are intrinsically fractal.

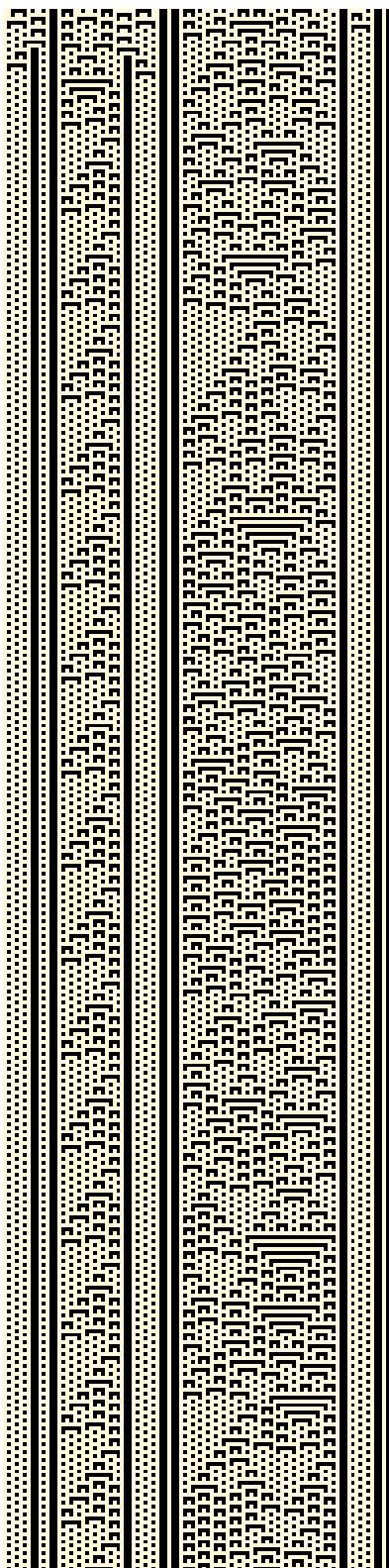


**Figure 5.5.** Evolution of the ECA Rule-90 of size 1024 starting with a single black cell, i.e. a 1, at the center. If instead there is a single 0 at the beginning, it gives the same result except with a black line at the top. This figure matches exactly with the difference pattern between two ECA's of the same rule and size where, after a brief transient (one iteration is enough), the state of the central cell is altered to be a 1. The ECA forms a fractal named Sierpiński triangle.

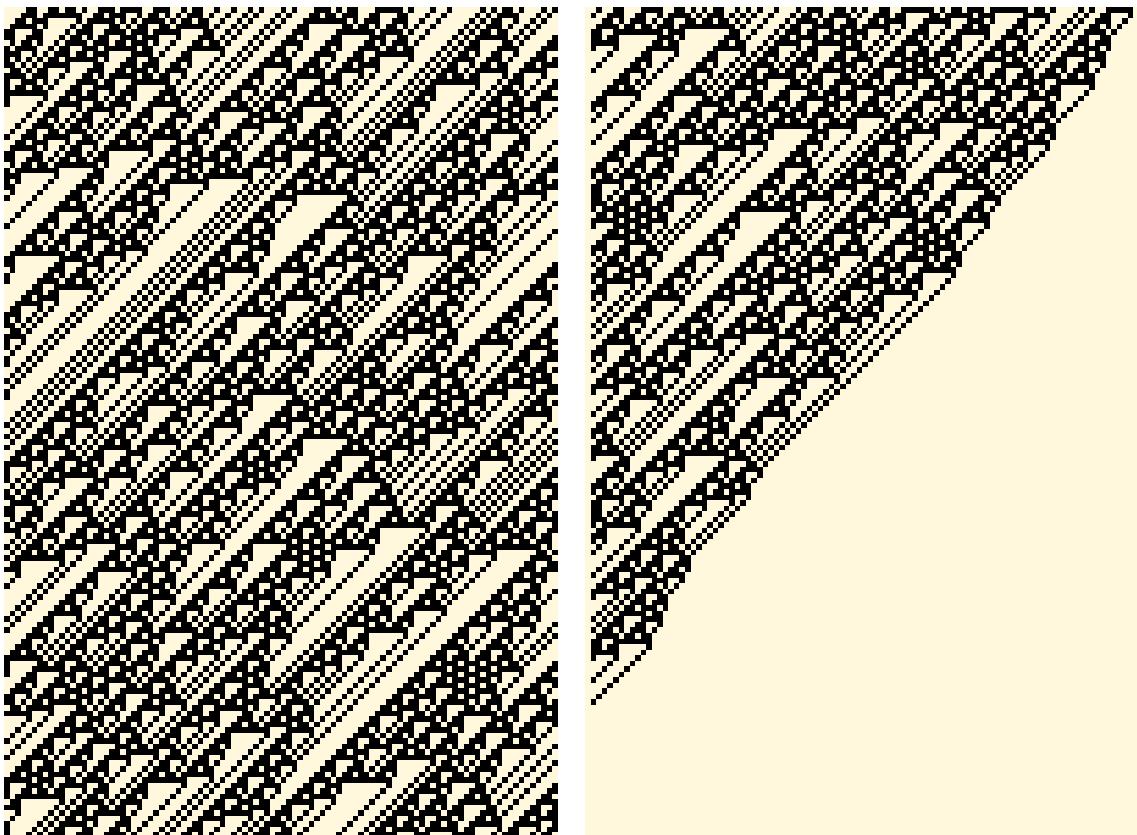
Class-3 automata evolve to complex patterns, so the fact that all initial conditions produce the same time series, is hard to believe. We can explain it analysing an intrinsic characteristic of these rules. For this explanation we will consider Rule-90 and its truth table from Table 5.1. The rest of rules in this subclass have the same explanation. Choose any row from the table. If that row has a 0 at the right side, then altering any of the trios at the left side of any row by adding modulo 2 the trio of the row you chose would not alter the result on the right. If instead it had a 1 at the right, the it would indeed alter the result. The rule itself is fractal. These only happens with rules in subclass U and marks the fractal behaviour of these rules, since, for these rules the evolution of a single 1 in a sea of 0s or vice-versa produces a fractal pattern like in Fig. 5.5, that is exactly the same as the difference pattern of the rule with any initial condition.

Finally Class-4 is represented by subclass T, in which rules present transient chaos in the Hamming distance time series. The chaotic regime holds while there is complex behavior in the patterns and finishes when the evolution saturates in a periodic state.

Each subclass belongs to one Wolfram class as appears on *WolframAlpha* except for two rules. This is the case for Rule-73, which belongs to Wolfram Class-2 but subclass T. The rule was classified in this subclass because in one, and only one, of



**Figure 5.6.** Time evolution of Rule-73 over 400 iterations. Besides being classified as Class-2 in *WolframAlpha*, this patterns belong to Class-4 more accurately.



**Figure 5.7.** Time evolution of Rule-106. At the left, the evolution under periodic boundary conditions shows patterns that never repeat for long times like those at Class-3. At the right, evolved under a fixed boundary, the zeros propagate to the left until all cells are zero, making it fall under definition of Class-1.

the 100 time series we saw transient chaos. In Fig. 5.6 we see an evolution pattern that could be classified to Wolfram Class-4, but this behaviour happens in rare occasions. Rules in the *HP* subclass are close to being from subClass-*T* also, but since the times where the patterns are complex are very narrow, we did not detect transient chaos.

Rule-106 is also an exception. We have classified it as subclass C whereas in *WolframAlpha* is classified as Class-4. Watching the pattern evolution in Fig. 5.7 we clearly see this is an error, as the patterns looks more akin to those in Class-3.

### 5.3 Discussion and conclusions

We have made an algorithm that classifies the ECA rules analyzing a time series instead of analyzing an image. Thus requires less quantity of data to analyze than Wolfram method, so it is more effective. The classification separates the rules in the same classes save for some arguably misclassified rules in the *WolframAlpha* engine.

This is not a setback because Wolfram's classification was subject of variance given different initial conditions.

With the autocorrelation of the Hamming distance we can also obtain its period, but this is not generally the period of the cellular automaton, and has a strong dependence with populations size.

## Bibliography

- [1] J. von Neumann, A. W. Burks, Theory of Self-Reproducing Automata. University of Illinois Press (1966).  
<https://cdn.patentlyo.com/media/docs/2012/04/VonNeumann.pdf>
- [2] D. Chowdhury, L. Santen, and A. Schadschneider, Statistical physics of vehicular traffic and some related systems. *Phys. Rep.* **329**, 199–329 (2000).  
[https://doi.org/10.1016/S0370-1573\(99\)00117-9](https://doi.org/10.1016/S0370-1573(99)00117-9)
- [3] C. Burstedde, K. Klauck, A. Schadschneider, and J. Zittartz, Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A* **295**, 507–525 (2001). [https://doi.org/10.1016/S0378-4371\(01\)00141-8](https://doi.org/10.1016/S0378-4371(01)00141-8)
- [4] D. H. Rothman and J. M. Keller, Immiscible cellular-automaton fluids. *J. Stat. Phys.* **52**, 1119–1127 (1988). <https://doi.org/10.1007/BF01019743>
- [5] L. Kadanoff, S. R. Nagel, L. Wu, and S. Zhou, Scaling and universality in avalanches. *Phys. Rev. A* **39**, 6524 (1989).  
<https://doi.org/10.1103/PhysRevA.39.6524>
- [6] D. A. Meyer, From quantum cellular automata to quantum lattice gases. *J. Stat. Phys.* **85**, 551–574 (1996).
- [7] P. Tougaw, L. Douglas, and S. Craig, Logical devices implemented using quantum cellular automata. *J. Appl. Phys.* **75**, 1818–1825 (1994).  
<https://doi.org/10.1007/BF02199356>
- [8] S. Nandi, B. K. Kar, and P. Pal Chaudhuri, Theory and applications of cellular automata in cryptography. *IEEE Trans Comput* **43** 1346–1357 (1994).  
<https://doi.org/10.1109/12.338094>
- [9] J. Machiacao, A. G. Marco, and O. M. Bruno, Chaotic encryption method based on life-like cellular automata. **39**, 12626–12635 (2012).  
<https://doi.org/10.1016/j.eswa.2012.05.020>
- [10] J. Lechleiter, S. Girard, E. Peralta, and D. Clapham, Spiral calcium wave propagation and annihilation in *Xenopus laevis* oocytes. *Science* **252**, 123–126 (1991). <https://doi.org/10.1126/science.2011747>

- [11] K. C. De Carvalho and T. Tomé, Probabilistic cellular automata describing a biological two-species system. *Mod. Phys. Lett. B* **18**, 873–880 (2004).  
<https://doi.org/10.1142/S0217984904007396>
- [12] M. Redeker, A. Adamatzky, and G. J. Martínez, Expressiveness of elementary cellular automata. *Int J Mod Phys C* **24**, 1350010 (2013).  
<https://doi.org/10.1142/S0129183113500101>
- [13] S. Wolfram, Universality and complexity in cellular automata. *Physica D* **10**, 1–35 (1984). [https://doi.org/10.1016/0167-2789\(84\)90245-8](https://doi.org/10.1016/0167-2789(84)90245-8)
- [14] M. Cook, Universality in Elementary Cellular Automata. *Complex Syst.* **15**, 1–40 (2004).  
<https://content.wolfram.com/sites/13/2018/02/15-1-1.pdf>

## Chapter 6

# Escaping from transient chaos with partial control

*“Fantasy is hardly an escape from reality. It’s a way of understanding it.”*

-Lloyd Alexander

Partial control has been developed as a control method generally used in chaotic transients, where the controller does not fix a given orbit, but instead chooses a set of points to avoid. In this chapter, we introduce this concept to further develop it in the two following chapters, where we use the partial control method to solve a game between two players.

### 6.1 Introduction

The partial control method was firstly introduced to solve the Yorke’s Game of Survival in [1] where a game is defined between a “protagonist” and an “antagonist”. The goal of the protagonist was to survive in a chaotic transient region indefinitely. In the former paper’s case, the region was  $[0, 1]$  with the slope three tent map  $f(x) = \mu(1 - |x|) - 1$  as the dynamics of the system. The antagonist role was interpreted by the effect of an additive noise, which bound was larger than the control bound of the protagonist.

This map is well known and the iteration of the map in the sequence  $x_{i+1} = f(x_i)$  for slopes greater than two, presents transient chaos. This is because the map is only positive for the region  $[-1, 1]$  and for those slopes, the orbits leave that region

---

G. Alfaro, R. Capeáns, and M. A. F. Sanjuán, Forcing the escape: Partial control of escaping orbits from a transient chaotic region, Nonlinear Dyn. **104**, 1603–1612 (2021).  
<https://doi.org/10.1007/s11071-021-06331-4>

after falling near 0.5 and escape towards  $-\infty$ . Through the iteration of the map all initial points eventually escape, since only the zero-measure Cantor ternary set remains an indefinite amount of time in the chaotic transient. The escape times depend on the initial conditions and present fractality. Therefore they are highly unpredictable in the disturbed case as can be seen on Fig. 6.1. Orbits that start near the Cantor set take many iterations to escape without disturbance. Although in Fig. 6.1 we used the logistic map, the graphic is similar for the tent map.

Given that all points eventually escape and that the noise bound is greater it is unlikely that the protagonist could survive. However, with the partial control method, it is possible to find sets of points called the safe set for each combination of control bound and noise bound. Given a noise bound and a control bound there is a unique set that enables the protagonist to stay in the region if the initial conditions belongs to the set, for as long as control is active. The sets were firstly found by using the sculpting algorithm [2] and later the safe functions were developed which enable to find multiple safe sets for different control bounds more efficiently [3].

But what drives the study presented in this chapter is not to maintain the orbits in the chaotic region, but to take out the unpredictable manner in which orbits escape from the region. We aim to control the orbits in order of being certain to know how the orbits will escape. We will get some functions similar to the safe functions that allows us either to escape from the chaotic transient region in the least iterations possible, or to set the exact number of iterations that passes before the orbits escape. We called these functions the escape functions, and the sets, the escape sets.

## 6.2 Escape from a region with partial control

Given a map  $f(x)$ , with a bounded additive noise  $\xi_n$  that affects the map at each iteration  $n$ , we eject a bounded additive control  $u_n$ , which, as we mentioned earlier, can be smaller than the noise bound. We define the iterated map that acts on  $q \in Q$ ,  $Q$  being a region in phase space as:

$$\begin{aligned} q_{n+1} &= f(q_n) + \xi_n + u_n \\ |\xi_n| &\leq \xi_0 \\ |u_n| &\leq u_0. \end{aligned} \tag{6.1}$$

The goal is to find the correct sequence of controls  $(u_1, u_2, \dots, u_m)$  to ensure that at the  $m^{th}$  iteration, with  $m \leq N$ , the point  $q_{m+1}$  is outside of the region  $Q$ .  $N$  is the maximum number of iterations it will take to achieve the goal, but in some cases, depending on the initial condition, the orbit can be expelled sooner. When setting a very small  $N$ , the control bound should be high, or otherwise achieving the goal may not be possible. On the other hand, by setting a large  $N$  the control bound can be smaller and smaller.

To apply this control technique one should firstly choose the region  $Q$  from where the orbits must escape. Then one should compute  $N$  quick escape functions  $U_k$ ,  $k = 1 : N$ , which depend on the value  $\xi_0$ . These functions tell how much control is needed to expel the orbit in  $k$  iterations or less.

Then, by setting the control bound  $u_0$  compute the escape sets  $E_k$  for every escape function. Every point which belongs to this set can be thrown out from the region  $Q$  in  $k$  or fewer iterations. Each escape set contains all the points which have a value of  $U_k$  lower than  $u_0$ . Since  $E_N$  may not gather every point in the region  $Q$ , if the orbit starts in one of those point that are not gathered in  $E_N$ , the escape is not guaranteed to be made in  $N$  iterations, and could take larger times.

Now, when all escape sets are computed, the control strategy is to push the point  $f(q_i) + \xi_i$  towards the escape set with the lower index  $k$  that is within control reach.

We also developed a similar algorithm to make the escape from the region  $Q$  in an orderly manner, i.e., to escape in exactly  $N$  iterations, no more, no less. The procedure is the same as we have explained above, except that the escape functions are different and for the control strategy we have to push the first point towards  $E_N$ , the second iteration towards  $E_{N-1}$  and so on, until the controlled orbit gets to  $E_1$  after which the application of the map, noise, and control, the point is outside region  $Q$ .

Finally we presented a practical case in which we shift the trajectory from one chaotic region to another periodically, resulting in a quasiperiodic orbit.

## 6.3 Application of the method

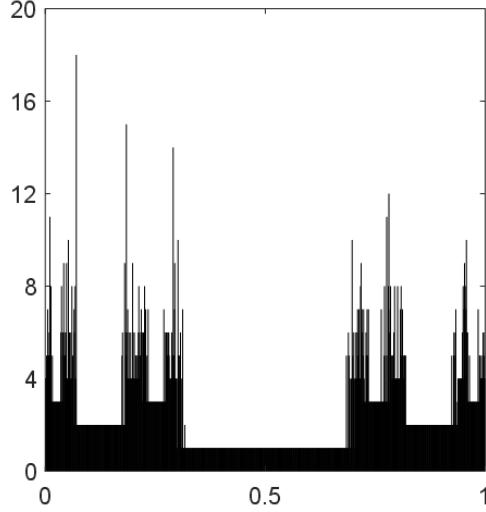
To give an example on how to apply the algorithms, we have chosen the logistic function  $f(x) = \mu x(1 - x)$ . For  $\mu > 4$  there is a chaotic transient on the region  $Q = [0, 1]$ .

In the following subsections we present the algorithms to calculate the escape functions and sets in order to achieve three distinct goals. First, to make the orbits escape region  $Q$  as quickly as possible setting a maximum number of iterations  $N$ . Next, to set an exact number of iterations  $N$  after which the orbit must be outside region  $Q$ . Finally, to make a quasiperiodic orbit that shifts between two chaotic transients at fixed intervals  $N_1$  and  $N_2$  iterations.

### 6.3.1 Escaping as quick as possible

We want to make orbits to escape in  $N$  iterations or less. The larger the control bound is, the quicker some orbits will escape and the escape sets will be broader. This will be helpful if one wishes to avoid as much as possible chaotic orbits which can be dangerous in some situations. The authors of [4] state that “transient chaos may be a dangerous and unwanted state of a vibro-impact system”.

Since we are making numerical calculations, we must use a grid on region  $Q$ . We divide the region in  $M$  equal parts, so that the region transforms in a collection of



**Figure 6.1.** Escape times from the chaotic transient of the logistic map with  $\mu = 4.7$  and a noisy disturbance bounded to  $\xi_0 = 0.03$ . Horizontal axis represents the starting points for the orbit and the vertical axis, the number of iterations of a random orbit before it escaped the chaotic transient region  $Q = [0, 1]$ .

equidistant points  $q[i]$ ,  $i = 1 : M$ . Also the value of the noise is discretized into  $W$  values ranging from  $-\xi_0$  to  $\xi_0$ . The map then becomes:

$$q[j] = f(q[i]) + \xi[s] + u[i, s, j], \quad (6.2)$$

where  $q[j]$  is the arrival point with  $j = 1 : M$ . Now we are ready to compute the escape functions.

Firstly we compute the escape function  $U_1$ . This function tells us the control necessary for every point in the region  $Q$  to escape in the next iteration.

$$U_1(q[i]) = \max_s(\min(f(q[i]) + \xi[s] + \epsilon - 0, 1 + \epsilon - (f(q[i]) + \xi[s]))), \quad (6.3)$$

where 0 and 1 are the limits of the region  $Q$  and  $\epsilon$  is a small number to ensure the escape and is convenient to be  $1/M$ .

If  $f(q[i]) + \xi[s]$  is beyond  $Q$  then change the function value to 0, since there is no need for control.

Then, we calculate the following functions recursively with this formula:

$$\begin{aligned} u[i, s, j] &= q[j] - f(q[i]) - \xi[s] \\ U_{k+1}^*[i] &= \max_s \left( \min_j \left( \max(|u[i, s, j]|, U_k[j]) \right) \right) \\ U_{k+1}[i] &= \min \left( U_k[i], U_{k+1}^*[i] \right), \end{aligned} \quad (6.4)$$

where  $u[i, s, j]$  is the control to take the resulting point, after the application of the map and noise, to each point in the grid. The last equation ensures that the escape can be done in less than  $N$  iterations.

We give an example with  $N = 3$  in Fig. 6.2. The escape sets in blue are the regions of the escape functions in red that have lower value than the control bound. Each index  $k$  in  $E_k$  tells in how many iterations is possible to escape.

Now in order to escape as quick as possible, the controller must push the iterated point of the orbit towards the lowest indexed escape set reachable.

By increasing the number of iterations we allow for the orbit to escape  $N$ , the escape sets will become larger. For  $N \rightarrow \infty$  the set becomes the entire region except the zero-measure Cantor ternary set. This will not be the case for the following case in which the orbit must escape at an exact number of iterations.

### 6.3.2 Escaping at an exact number of iterations

Now we ask ourselves if it is possible to set the exact number of iterations before an orbit is expelled from the chaotic transient. This seems a difficult task when considered the restriction of limiting control to be smaller than noise, since some points naturally escape in one iteration, while others take a lot of time to do so.

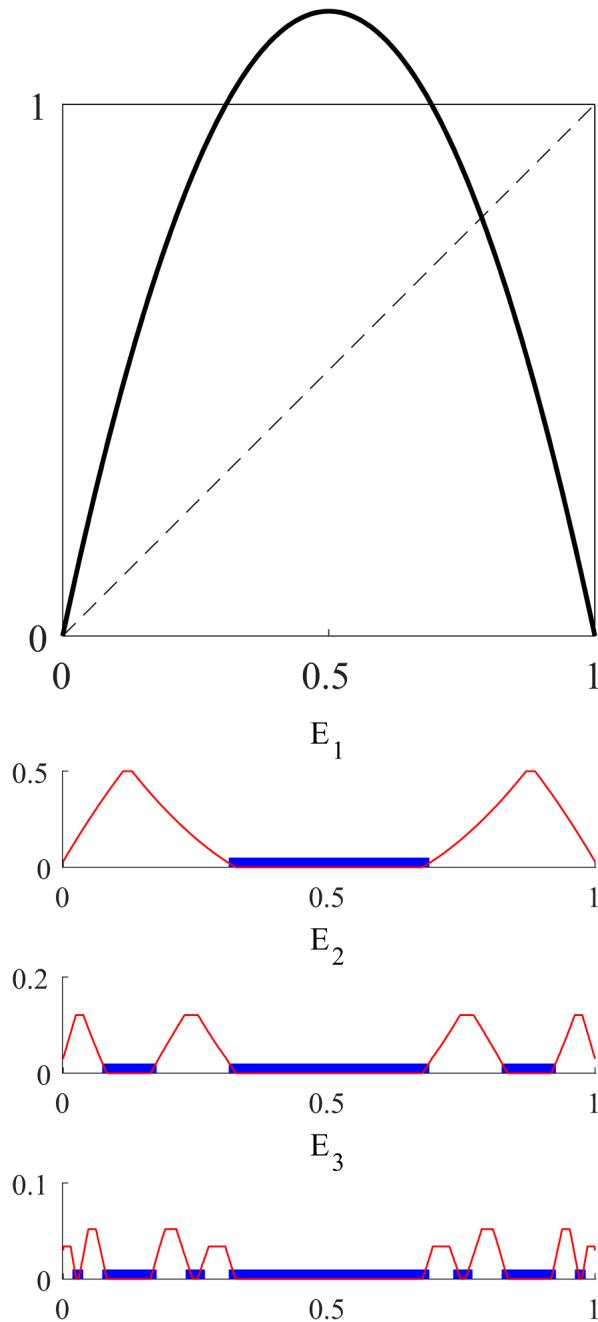
However, we repeat the same line of work as the previous case, by constructing the exact escape functions and later by getting the exact escape sets. When inspecting the logistic map, points at the center will escape naturally in the next iteration, so probably these points won't be in the exact escape sets for large values of their index. On the other hand points near the Cantor ternary set seem to be out of bounds for the sets with small index. This landscape will be reflected on the exact escape functions which are constructed differently than in the previous case.

In spite of the task seeming harder, the algorithm to calculate the escape functions is very similar and even simpler.  $U_1$  is calculated the same obviously. Then the rest of the functions are computed exactly as equations 6.4 but without the last equation. In this case, we do not check whether the function is smaller than the function with lower index since we cannot allow for an earlier escape.

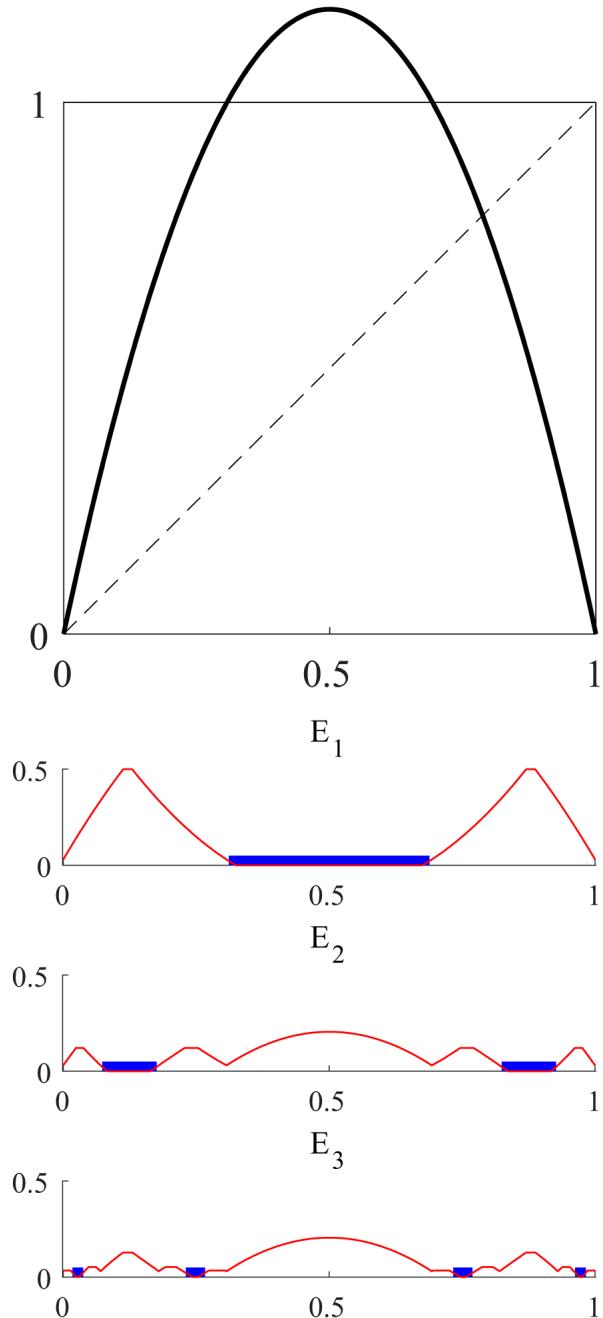
$$\begin{aligned} u[i, s, j] &= q[j] - f(q[i]) - \xi[s] \\ U_{k+1}[i] &= \max_s \left( \min_j \left( \max(|u[i, s, j]|, U_k[j]) \right) \right) \end{aligned} \quad (6.5)$$

Following this formula, we arrive to the exact escape functions and sets in Fig. 6.3. We can see that the sets are smaller, because the set  $E_k$  doesn't contain the contents of  $E_{k-1}$ , unlike the previous case. Each exact escape set  $E_k$  is the complementary of the  $k-th$  approximate Cantor ternary set  $C_k$ , though a little bit decreased or increased depending on whether the noise is greater than the control or vice versa.

Now, to escape in exactly  $N$  iterations, we must simply take the first iterated point towards  $E_{N-1}$ . This will be possible only if the initial point started in  $E_N$ .



**Figure 6.2.** On top, logistic map  $\mu x(1-x)$  with  $\mu = 4.7$ . On the bottom, escape functions (red) and escape sets (blue) for escaping in  $N = 3$  or less iterations as quick as possible. The functions were computed for a noise bound of  $\xi_0 = 0.03$  and the sets for a control bound of  $u_0 = 0.022$ . The vertical axis shows the minimum control needed for orbits starting at each starting point to stay in the chaotic transient region  $Q = [0, 1]$  indefinitely, which mark the quick escape functions in red. The quick escape sets  $E_k$  are highlighted in blue and their heights are trivial. The points in these sets can be controlled with the given control bound to escape in at maximum  $i$  iterations.



**Figure 6.3.** On top, logistic map  $\mu x(1 - x)$  with  $\mu = 4.7$ . On the bottom, escape functions (red) and escape sets (blue) for escaping in  $N = 3$  or less iterations as quick as possible. The functions were computed for a noise bound of  $\xi_0 = 0.03$  and the sets for a control bound of  $u_0 = 0.022$ . The vertical axis shows the minimum control needed for orbits starting at each starting point to stay in the chaotic transient region  $Q = [0, 1]$  indefinitely. The blue lines mark the exact escape sets  $E_k$ , their heights are meaningless, the points in  $E_k$  can escape in exactly  $k$  iterations with the given control bound.

Afterwards we will push the iterated point one escape set further at a time until  $E_1$  is reached. Then, after iterating the map the orbit will be out of the chaotic region or the border will be within reach of control.

### 6.3.3 Shifting chaotic transients

Thanks to the previous control method we can accurately control the escape from a chaotic transient in an orderly manner. This can be helpful in a case where a rigorous control of time is needed. This next case goes forward in this direction, by setting the frequencies of time at which a controlled orbit shifts from one chaotic transient to another. With this control technique one can build a chaotic clock in which a desired chaotic system changes significantly at some desired rates and still maintaining its chaotic behavior.

This can be useful in multistable chaotic systems. In these kind of systems, varying a parameter one can merge various attractors into a larger one with different regions when their basin boundary collides. The orbit shifts chaotically from one region to the other continuously. One example of this motion is the Lorenz system [5], and others are given in [6, 7, 8, 9].

With the following control method, one could make this shifts to appear periodically as long as control is made while maintaining the chaotic motion through the rest of the orbit. The orbit will therefore be quasi-periodical.

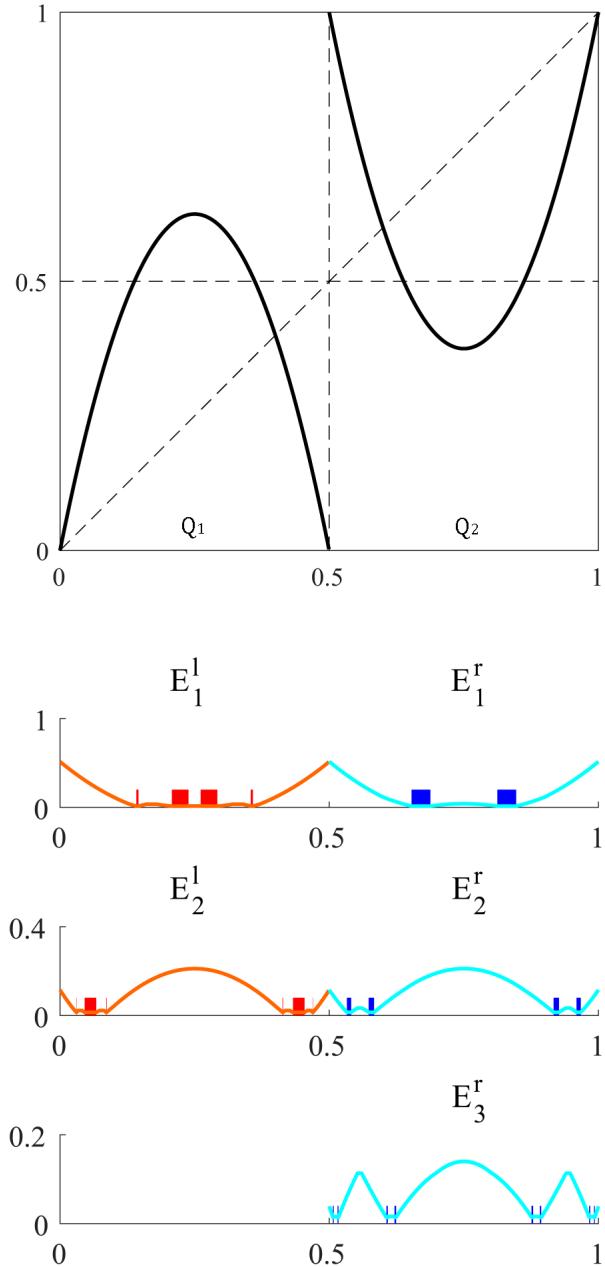
We will develop this method on a simple model to show the procedure. On the top graphic of Fig. 6.4 we show the custom made double parabola map:

$$f(x) = \begin{cases} -\mu(x^2 - \frac{1}{2}x) & \text{if } x < 0.5, \\ 1 + \mu(x^2 - \frac{3}{2}x + \frac{1}{2}) & \text{if } x \geq 0.5, \end{cases} \quad (6.6)$$

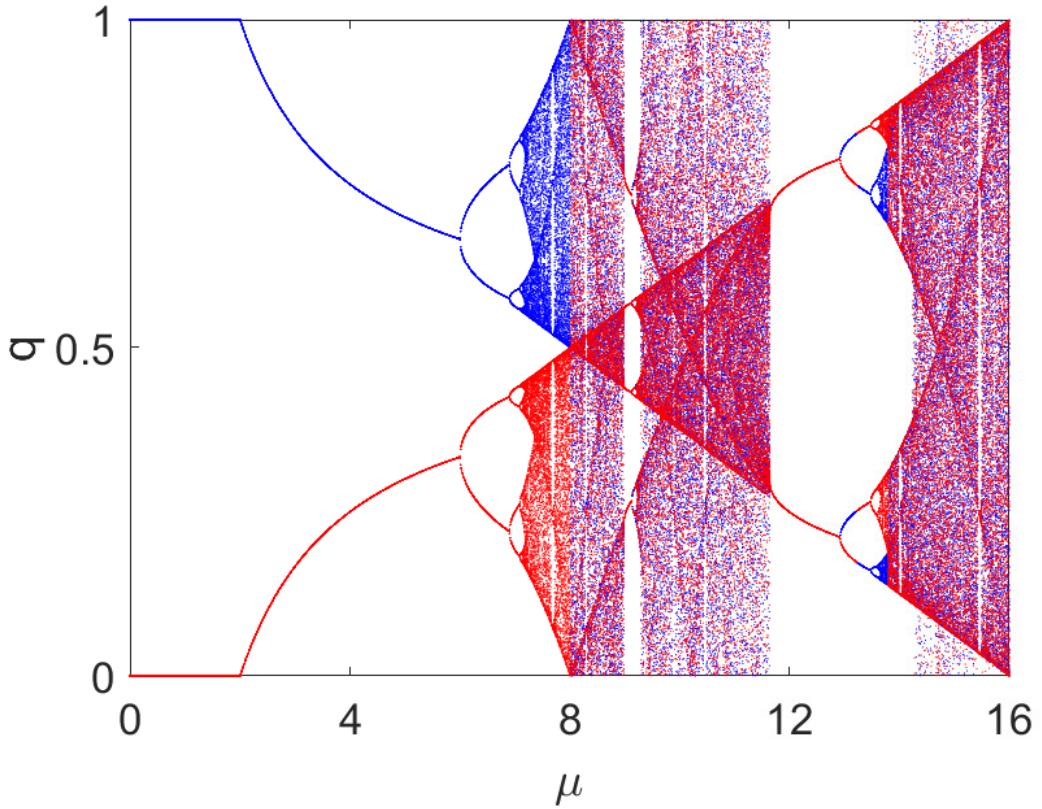
The bifurcation diagram and Lyapunov exponent for the iterated map  $x_{n+1} = f(x_n)$  are shown in Figures 6.5 and 6.5 respectively. It shows that there is a chaotic behavior for most values of  $\mu \geq 7.14$  and for values of  $7.14 \leq \mu < 8$  there are two separated attractors that merge at  $\mu = 8$  into a larger one. Then, for values close to that, the orbit shifts chaotically between the left and right regions.

Figure 6.7 shows one typical orbit and the controlled one. The uncontrolled trajectory is characterized by chaotic shifts from the left region to the right one, while the controlled trajectory results in a quasi-periodic signal with periodic shifts between the two regions. Also, we can see that the magnitude of the control made at each iteration is no more than a certain bound, which is lower than the noise bound.

To control this orbits we have used the same approach as the previous cases. In this case through the shift functions and sets. which are similar than the escape sets. This time we have to take into account how to stay in a region and how to leave from it and go to the other. So, we need to calculate the control  $u_{in}^{l,r}$  as the



**Figure 6.4.** On top, the custom made double parabola map for  $\mu = 10$ . It present two differentiated zones at left and right of the middle point at 0.5. At the bottom we present the shift functions as lines and sets as blocks. The shift fucntions  $U_k^{l,r}$  tells how much control is needed at least to keep the orbit in the left or right region for  $k$  iterations before shifting to the other region. In this case we have calculated the functions and sets in order to keep the orbit  $N^l = 2$  iterations at the left and  $N^r = 3$  iterations to the right.



**Figure 6.5.** Bifurcation diagram for the double parabola map choosing two different initial conditions. Te points in red correspond to starting the trajectories in  $Q_l$  and when in blue, the trajectories started in  $Q_r$ . For  $\mu > 8$  a global attractor merges and both regions are accessible to the trajectories.

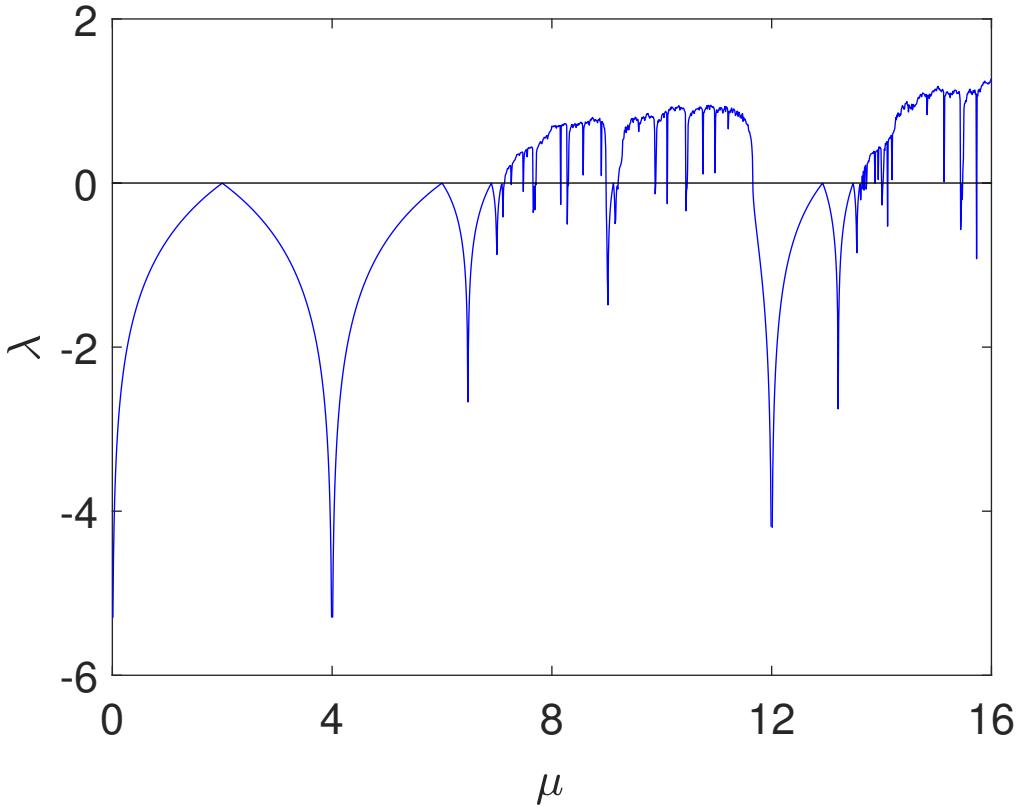
control needed to get an iterated point in region  $Q^{l,r}$  to another desired point in the same region.

$$u_{in}^{l,r}[i, s, j] = q^{l,r}[j] - f(q^{l,r}[i]) - \xi[s].$$

And to change regions, the control  $u_{out}^{l,r}$  tells the distance between one point from region  $Q^{l,r}$  to another from the opposite region  $Q^{r,l}$ .

$$u_{out}^{l,r}[i, s, j] = q^{r,l}[j] - f(q^{l,r}[i]) - \xi[s].$$

Taking as a seed  $U_1^l = 0$ , we can calculate the shift functions  $U_k^{l,r}$  following Algorithm 6.3.1. This algorithm calculates the amount needed to control the orbit  $k$  iterations at the left, or right, region before shifting to the other region. The shift sets  $E_k^{l,r}$  will be the set of points that have a lower value of the function than their bound of control. The sets  $E_k^{l,r}$  are ordered like the exact escape sets from previous section, with the index  $k$  indicating the number of iterations needed until the orbit will shift to the other region.

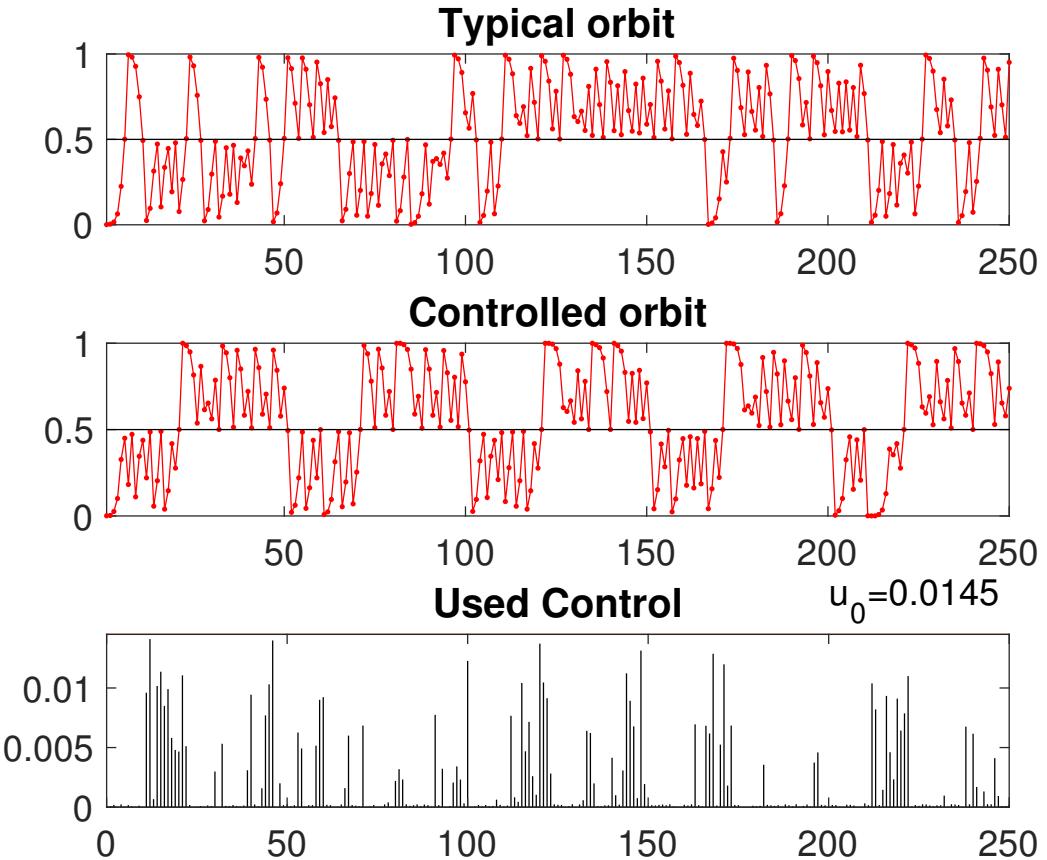


**Figure 6.6.** Evolution of the Lyapunov exponent versus  $\mu$  for the double parabola map. For positive values of the exponent, the dynamics are chaotic.

An example is given in Fig. 6.4. We calculated the shift sets in order for the orbit to remain  $N^l = 3$  at the left side, and  $N^r = 2$  at the right side. Following the example, if an orbit starts at a point in  $E_3^l$  it will arrive at  $E_2^l$  and then  $E_1^l$  through control after which, it can be controlled to shift to  $E_2^r$ , then repeating the same process on the right side until the orbit can be shifted to the left. This process can be continued indefinitely. If one is fortunate and the orbit starts at a point with low value of the shift function, the control can be made lower than the noise.

## 6.4 Discussion and conclusions

We have applied the partial control method with the goal of driving off trajectories from a chaotic transient region. We developed two methods of doing this. Firstly the controller may want to expel the trajectory as swift as possible. Alternatively, the control can be made in an orderly manner, choosing the exact number of iterations that the trajectory stays at the transient region from the beginning of control to the moment the trajectory leaves the region.



**Figure 6.7.** Controlled and uncontrolled orbits of the double parabola map for  $\mu = 8.1$  in red. The typical orbit describes the motion of a chaotic system where the attractor has different regions. The orbits stay an indefinite amount on each side of the attractor shifting chaotically. This is typical of a multistable chaotic system in the moment when its different attractors have merged into a larger one. We were able to control the dynamics in the map so it keeps the chaotic behavior in each side of the attractor, but have controlled the frequency of the shifts. As we can see the controlled orbits spends  $N^l = 20$  iterations at the left side and  $N^r = 30$  at the right one. The control made at each iteration was at most the value of its bound  $u_0 = 0.0145$  which is lower than the noise bound  $\xi_0 = 0.015$

Another case was studied in which the controller shifts periodically between two regions of a multistable chaotic system. In these kind of systems the trajectory shift in chaotic manner from one region to the other, but with the control method provided here, we controlled the frequency of these shifts. The result is a quasi-periodic orbit in which there is a periodical shifting through the two regions, but the orbit inside each region remains chaotic.

---

**Algorithm 6.3.1** Iterative process to calculate the shift functions  $U_k^{l,r}$  to stay in the left or right side of the double parabola map a number of iterations  $N^l$  and  $N^r$  respectively.

---

```

while  $U_k^{l,r}$  are different than previous iteration do
  for  $k = 1 : N^l - 1$  do
     $U_{k+1}^l(q[i]) = \max_s \left( \min_j \left( \max(u_{in}^l[i, s, j], U_k^l(q[j])) \right) \right)$ 
  end for
   $U_1^r(q[i]) = \max_s \left( \min_j \left( \max(u_{out}^r[i, s, j], U_{N^l}^l(q[j])) \right) \right)$ 
  for  $k = 1 : N^r - 1$  do
     $U_{k+1}^r(q[i]) = \max_s \left( \min_j \left( \max(u_{in}^r[i, s, j], U_k^r(q[j])) \right) \right)$ 
  end for
   $U_1^l(q[i]) = \max_s \left( \min_j \left( \max(u_{out}^l[i, s, j], U_{N^r}^r(q[j])) \right) \right)$ 
end while

```

---

The control method does not provide a given orbit to follow strictly, but instead analyses the system and finds sets of points reachable within control limits that serve to complete the goal. The controller can then choose from those points, whether randomly or with other priorities in mind, where to direct the orbit. The outstanding fact about this control method is that there are occasions in which the maximum control can be set to be lesser than a known noise bound.

A simple example is given for each of the three cases. The logistic map was studied in the first two cases and for the control of a multistable system, we crafted a custom made map consisting of two disjointed parabolas with opposite convexity.

For all the cases we managed to achieve the goal at some set of initial conditions even when control was lesser than noise. Increasing the control ensures that more initial conditions are able to be controlled.

## Bibliography

- [1] J. Aguirre, F. d’Ovidio, and M. A. F. Sanjuán Controlling chaotic transients: Yorke’s game of survival Phys. Rev. E **69**, 016203 (2004). <https://doi.org/10.1103/PhysRevE.69.016203>
- [2] J. Sabuco, S. Zambrano, M. A. F. Sanjuán, and J. A. Yorke, Finding safety in partially controllable chaotic systems. Commun Nonlinear Sci Numer Simulat **17**, 4274–4280 (2012). <https://doi.org/10.1016/j.cnsns.2012.02.033>
- [3] R. Capeáns, J. Sabuco, M. A. F. Sanjuán, A new approach of the partial control method in chaotic systems. Nonlinear Dyn. **98**, 873–887 (2019). <https://doi.org/10.1007/s11071-019-05215-y>

- [4] V.A. Bazhenov, O. S. Pogorelova, and T.G. Postnikova, Transient chaos in platform-vibrator with shock. Strength Mater. Theory Struct. **106**, 22-40 (2021). <https://doi.org/10.32347/2410-2547.2021.106.22-40>
- [5] E. N. Lorenz, Deterministic nonperiodic flow. J. Atmos. Sci. **20**, 130-141 (1963). [https://doi.org/10.1175/1520-0469\(1963\)020<0130:DNF>2.0.CO;2](https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2)
- [6] Y. F. Jin, *Stochastic resonance in an under-damped bistable system driven by harmonic mixing signal*, Chinese Phys. B **27**, 050501 (2018). <https://doi.org/10.1088/1674-1056/27/5/050501>
- [7] E. L. Rempel and A. C.-L. Chian, *Intermittency induced by attractor-merging crisis in the Kuramoto-Sivashinsky equation*, Phys. Rev. E **71**, 016203 (2005). <https://doi.org/10.1103/PhysRevE.71.016203>
- [8] A. L. Liverati, I. L. Caldas, C. P. Dettmann, and E. D. Leonel, *Crises in a dissipative bouncing ball model*, Phys. Lett. A **379**, 2830–2838 (2015). <https://doi.org/10.1016/j.physleta.2015.09.016>
- [9] S. Vaidyanathan, S. T. Kingni, A. Sambas, M. A. Mohamed, and M. Mamat, *A new chaotic jerk system with three nonlinearities and synchronization via adaptive backstepping control*, Int. J. Eng. Technol. **7**, 1936–1943 (2018). <https://doi.org/10.14419/ijet.v7i3.15378>

## **Chapter 7**

# **Two-player Yorke's game of survival in chaotic transients**

*"As a general rule, the most successful man in life is the man who has the best information."*

-Benjamin Disraeli

In the previous chapter, we have developed a method of partial control that followed the line of work initiated in [1] followed by papers on the issue [2, 3, 4] among others. The problem introduced by James Yorke was devised as a kind of game where one player is at a huge disadvantage but, nonetheless, can achieve its goal. From there on the studies derived in a control method that did not force a single trajectory as the solution, but provided a set of points, the *safe sets*, where the controller was safe; hence the name *partial control*.

On the other hand, game theory provides powerful tools for analyzing strategic interactions across diverse fields, from social sciences to economics and physics [5, 6, 7]. While classical game theory typically focuses on equilibrium states, many real-world situations involve chaotic systems, where extreme sensitivity to initial conditions and inherent unpredictability create fundamental challenges for strategic decision-making. The nonlinear nature of these systems makes traditional game-theoretic approaches insufficient, as small perturbations can lead to dramatically different outcomes. What's more, the intrinsic characteristics of a game are susceptible to change in real life, and these changes can result from the players' decisions [8, 9].

---

G. Alfaro, R. Capeáns, M. A. F. Sanjuán, Two-player Yorke's game of survival in chaotic transients. Phys. Rev. E (2025).

<https://doi.org/10.48550/arXiv.2501.12188>

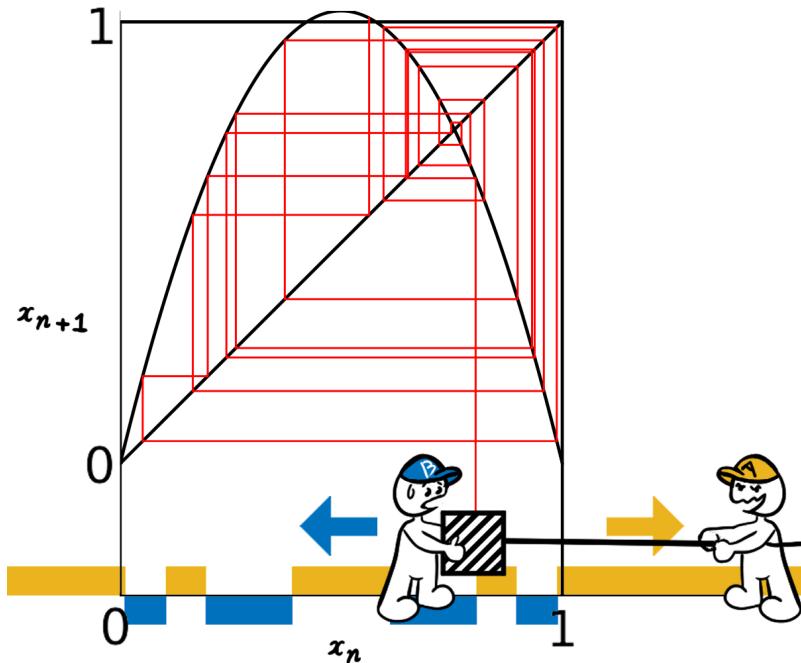
Game theory has also proven valuable in chaos control [10], where control problems naturally emerge as competitive scenarios between opposing objectives. This framework reveals how controllers must optimize their strategies while dealing with three key challenges: (1) the unpredictable nature of chaotic dynamics, (2) the system constraints, and (3) the actions of other controllers who must carefully choose their actions to achieve their own goal. This interaction between different control agents adds a strategic dimension that goes beyond traditional chaos control methods.

Since this thesis is oriented towards game dynamics, the opportunity to rejoin these two branches of the problem led to the publication of a paper, which we collect in this chapter. In this paper we present a two-player game of control where each player has conflicting objectives. One player wants to control the trajectory of a given dynamical system towards one region and the opponent towards a different one. Through the analysis of the game with the partial control tools we achieve the solution of the game through all initial conditions. We will construct the *winning sets* (what was previously called *safe sets*), as the initial conditions that guarantee victory for each player.

## 7.1 Introduction

To illustrate the game we studied a paradigmatic dynamical system, the logistic map  $f(x) = \mu x_n(1 - x_n)$ . The system presents transient chaotic dynamics for  $\mu > 4$  where all orbits starting at the region  $Q = [0, 1]$  eventually leave the region in a finite time. When one player aims to stay at the region  $Q$  indefinitely and the other aims to drive the trajectory off, the game gets very interesting. In one hand, the game is asymmetric since the player who wants to leave region  $Q$  is in advantage. On the other hand, we found that there are initial conditions where the player that aims to conserve the trajectory in the transient region can do so even when their control is lesser than the opponent's control.

In many games the order of play is important. This game is no exception, here the importance lies in the information that the second player has when they see the action of the opponent. Knowing where the opponent is going to push the trajectory to will affect the decision of control of the second player, giving them an advantage that the first player lacks. To study this effect, we devised three scenarios. In the first game the player that aims to keep the trajectory in the region  $Q$  knows the action of the rival. In the second one, the informed player is the one who intends to expel the trajectory from the region. Finally, in the third game, no player knows the rival's action. The lack of knowledge in this last game will translate in the unsettled solution of the game. Regions where no player has the victory assured appear as a consequence of these lack of knowledge.



**Figure 7.1.** Players  $A$  and  $B$  compete for controlling the trajectory on the chaotic and escaping logistic map  $x_{n+1} = \mu x_n(1 - x_n)$ . Player  $B$  tries to maintain the trajectory on the region  $Q = [0, 1]$ , while fighting against the dynamics of the map, since for  $\mu > 4$  all points except a zero measure Cantor set eventually escape from the region. The red line shows an example of one orbit that ends up escaping. Furthermore, player  $B$  also fights against the opponent, player  $A$ , who aims to expel the trajectory from region  $Q$  in a finite time. To achieve their goals the players can control the trajectory with a given control bound. The yellow and blue rectangles are the *winning sets* of players  $A$  and  $B$ , respectively. These are the initial conditions that guarantee victory for each player.

## 7.2 The game

Consider two individuals pulling an object. Player  $A$  aims to pull the object to their side, while player  $B$  pulls it toward their own. In the absence of any other factors, the stronger player will prevail by exerting more force. However, the game becomes much more interesting when the movement of the object is influenced by a natural dynamic governed by a rule defined by a function  $f$ . In this case, the players must not only consider their opponent's force but also account for the inherent motion of the object. This becomes more challenging when the movement that describes  $f$  is chaotic. In such scenarios, even slight variations in how the players pull can result in entirely different and unpredictable outcomes, as illustrated in Fig. 7.1.

With this in consideration, we formulate our game. Let  $Q$  represent a region in the phase space, with a map  $f$  acting on the state space  $X$ . The initial state  $x$  of the game begins within  $Q$ . Player  $B$ 's objective is to keep the trajectory within  $Q$ , while player  $A$ 's goal is to drive the trajectory outside of  $Q$ . Each iteration of the

game involves players  $A$  and  $B$  selecting their respective bounded controls,  $u_n^A$  and  $u_n^B$ . At each discrete time step  $n$ , the state  $x_n$  evolves according to this dynamics

$$x_{n+1} = f(x_n) + u_n^A + u_n^B, \quad (7.1)$$

where  $u_n^A$  and  $u_n^B$  represent the control actions of players  $A$  and  $B$ , respectively. These controls are bounded by

$$|u_n^A| \leq u_0^A, \quad |u_n^B| \leq u_0^B. \quad (7.2)$$

Player  $B$  wins if the trajectory remains inside  $Q$  indefinitely, whereas player  $A$  wins if the trajectory exits  $Q$ . Once the trajectory has left  $Q$ , player  $B$  has no chance of returning it to the region.

A crucial factor in solving this game is the order of play. To address this, we consider three distinct scenarios summarized in Table 7.1. In game  $A^-B^+$ , player  $B$  has the advantage of knowing the actions of player  $A$  before making their move. This sequential structure allows player  $B$  to respond optimally to the choices of their rival. In game  $A^+B^-$ , player  $A$  has the advantage, with complete knowledge of moves by player  $B$  before acting. Finally, in game  $A^-B^-$ , both players make their decisions simultaneously, without any knowledge of the opponent actions.

Game Type	Order of Play	Information Available
Game $A^-B^+$	$B$ plays after $A$	$B$ knows $A$ 's action
Game $A^+B^-$	$A$ plays after $B$	$A$ knows $B$ 's action
Game $A^-B^-$	Simultaneous play	Both ignore the opponent's action

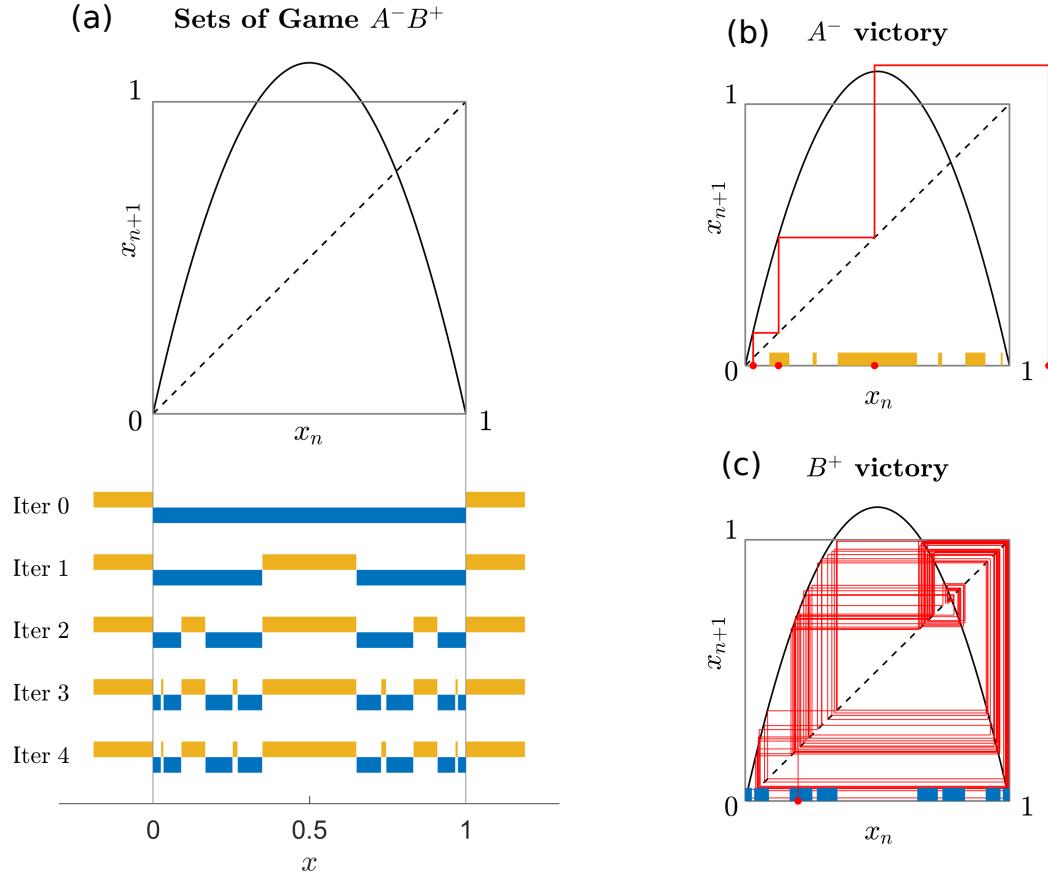
**Table 7.1.** Three different scenarios analyzed in this paper based on the order of play, i.e., what information is available to each player.

### 7.3 Solving the game: The winning sets

Given the opposing objectives described in the previous section, we define a region  $R$ , containing  $Q$ , where the winning sets  $W$  are determined for both players. These sets represent whether a given initial condition guarantees victory (value 1) or not (value 0). For player  $B$ , the initial condition is  $W^B(x) = 1$  for  $x \in Q$  and 0 otherwise. Conversely, for player  $A$ , the initial condition is  $W^A(x) = 0$  for  $x \in Q$  and 1 otherwise.

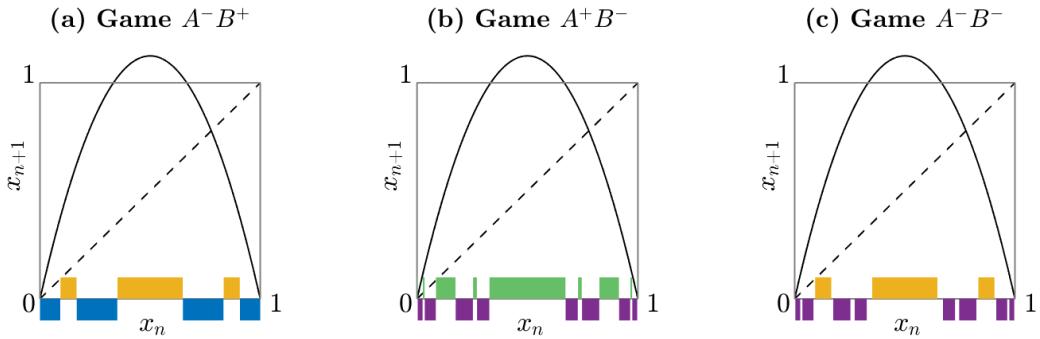
The computation process for each winning set depends on whether the player is informed by knowing the opponent's move, or ignorant. This procedure is summarized in Table 7.2. Each set can be computed independently, as outlined below:

- **Player  $A^-$  (ignorant player  $A$ ):** To compute the set of  $W^{A^-}$ , we have to evaluate for each  $x \in Q$  the image  $f(x) + u^A + u_0^B$  for all possible controls



**Figure 7.2.** Winning sets and controlled trajectory for a game between an ignorant player  $A$  and an informed player  $B$ . (a) Steps of the algorithm to compute the winning sets for the ignorant player  $A$ ,  $W^{A^-}$ , in yellow, and for the informed player  $B$ ,  $W^{B^+}$ , in blue. The sets were computed in the region  $R \in [-0.3, 1.3]$ , with  $Q \in [0, 1]$ ,  $u_0^A = 0.016$ ,  $u_0^B = 0.038$ . The algorithm converges in 3 iterations since the sets for the third iteration are identical to those of the forth. (b) Controlled trajectory of the game when the initial condition belongs to player's  $A$  winning set. Player  $A$  acts first, without knowledge of player  $B$ 's move, so they choose its control at each step to reach the closest point belonging to the shrunk set  $(W^{A^-} - u_0^B)$ , accounting for the worst possible subsequent action of player  $B$ . After 3 iterations the trajectory has left region  $Q$ , so player  $A$  wins. (c) The game now starts in a point that belongs to player's  $B$  winning set. Player  $B$  acts second, with the knowledge of player  $A$ 's move, so selects their control to reach the closest point belonging to set  $W^{B^+}$ . The trajectory stays indefinitely inside region  $Q$ , so player  $B$  wins as long as control is maintained.

$u^A \in [-u_0^A, u_0^A]$  and  $u^B \in [-u_0^B, u_0^B]$ . We select only those  $u^A$  where all possible images  $f(x) + u^A + [-u_0^B, u_0^B]$  fall within  $W^{A^-}$ . Points  $x \in Q$  satisfying this condition update  $W^{A^-}(x) = 1$ . All these initial conditions are able to be



**Figure 7.3.** The logistic map and winning sets to control trajectories of the logistic map with  $\mu = 4.5$  and control bounds  $u_0^A = 0.015$  for player  $A$ , and  $u_0^B = 0.040$  for player  $B$ . Each winning set at the bottom show the initial conditions that guarantee victory to each player. Each player has two winning sets whether they are informed (+) or ignorant (-) to their opponent's move. The winning sets of player  $A$  are colored in green when they are informed and in yellow when they are ignorant. On the other hand, those of player  $B$  are colored in blue when they are informed and purple when they are ignorant. For simplicity, we only represent the winning sets in the interval  $Q$ . (a) Ignorant player  $A$  against informed player  $B$ . (b) Now the informed player is  $A$  who plays against the ignorant player  $B$ . (c) Neither player knows each other actions. In this case, there are regions that do not belong to any player's winning set, meaning that both player could win when starting at those initial conditions, but the victory is uncertain because players cannot guarantee victory against any move from the opponent. These “no winning regions” represent the difference between the respective informed winning sets and the ignorant winning sets.

controlled to escape after one iteration of the map. But there will be other points that, after more than one iterations of the map and control, will be able to escape. To find this points we must repeat the algorithm, thus updating  $W^{A^-}$  until it converges.

This process is equivalent to performing the following morphological operations: first shrinking  $W^{A^-}$  by  $u_0^B$ , then dilating by  $u_0^A$ . For any  $x \in Q$  whose  $f(x)$  falls within the shrunk set, we set  $W^{A^-}(x) = 1$ . The process is repeated with this new  $W^{A^-}$  until it converges.

- **Player  $A^+$  (informed player  $A$ ):** To compute the set of  $W^{A^+}$ , we analyze  $f(x) + u^B + u^A$  for all possible controls. For each  $x \in Q$ , we evaluate  $f(x) + [-u_0^B, u_0^B] + u^A$  and determine if there exists some  $u^A \in [-u_0^A, u_0^A]$  capable of forcing escape for all possible images. If such  $u^A$  exists, then  $W^{A^+}(x) = 1$ . This process is performed for all  $x \in Q$ , updating the set  $W^{A^+}$ . The algorithm is repeated until  $W^{A^+}$  converges.

Morphologically, this process is equivalent to first dilating  $W^{A^+}$  with  $u_0^A$  and

then shrinking the resulting set with  $u_0^B$ . For any  $x \in Q$  whose  $f(x)$  falls within this shrunk set, we set  $W^{A+}(x) = 1$ .

- **Player  $B^-$  (ignorant player  $B$ ):** To compute the set of  $W^{B^-}$ , we evaluate for each  $x \in Q$  the image  $f(x) + u^B + u_0^A$  for all possible controls  $u^B \in [-u_0^B, u_0^B]$  and  $u^A \in [-u_0^A, u_0^A]$ . We select only those  $u^B$  where all possible images  $f(x) + u^B + [-u_0^A, u_0^A]$  fall within  $W^{B^-}$ . Points  $x \in Q$  not satisfying this condition update  $W^{B^-}(x) = 0$ . Since player  $B$  wants to keep the orbit in  $Q$  forever, the algorithm must be repeated until  $W^{B^-}$  converges.

The process is equivalent to iteratively performing the following morphological operations: first shrinking  $W^{B^-}$  by  $u_0^A$ , then dilating by  $u_0^B$ . For any  $x \in Q$  whose  $f(x)$  falls outside the shrunk set, we set  $W^{B^-}(x) = 0$ . The process is repeated with this new  $W^{B^-}$  until it converges.

- **Player  $B^+$  (informed player  $B$ ):** To compute the set of  $W^{B^+}$ , we analyze  $f(x) + u^A + u^B$  for all possible controls. For each  $x \in Q$ , we evaluate  $f(x) + [-u_0^A, u_0^A] + u^B$  and determine if there exists some  $u^B \in [-u_0^B, u_0^B]$  capable of guaranteeing containment for all possible images. If such  $u^B$  does not exist, then  $W^{B^+}(x) = 0$ . This process is performed for all  $x \in Q$ , updating the set  $W^{B^+}$ . The algorithm is repeated until  $W^{B^+}$  converges.

Morphologically, this process is equivalent to first dilating  $W^{B^+}$  with  $u_0^B$  and then shrinking the resulting set with  $u_0^A$ . For any  $x \in Q$  whose  $f(x)$  falls outside this shrunk set, we set  $W^{B^+}(x) = 0$ .

## 7.4 Game of survival in the logistic map

To demonstrate our method, we explore a game based on the logistic map,  $f(x) = \mu x_n(1 - x_n)$  with  $\mu = 4.5$  and  $Q = [0, 1]$ . This map provides an ideal scenario due to its transient chaotic behavior within  $Q$ : the trajectories exhibit chaotic dynamics before eventually leaving the region. This creates an asymmetric situation where player  $B$ , aiming to keep the trajectory inside the region, must contend with both the natural behavior of the system and the opposing player's objective of pushing the trajectory out. While this example is specific, our approach can be generalized to other maps  $f$  and regions  $Q$  in the phase space.

In this setup, both players are allowed to apply control at each step to influence the trajectory. However, the control actions of both players are constrained by their respective limits,  $u_0^A$  and  $u_0^B$ . The state of the system after each iteration is given by

$$x_{k+1} = f(x_k) + u^A + u^B, \quad (7.3)$$

where  $u^A$  and  $u^B$  denote the control that the players exert.

We present a specific example of the game with control bounds  $u_0^A = 0.016$  and  $u_0^B = 0.038$ . In Fig. 7.2, we illustrate how the winning sets are formed. As

Player	Initial Set	Morphological Operations
$A^-$	$W^{A^-}(x) = \begin{cases} 0 & x \in Q \\ 1 & \text{otherwise} \end{cases}$ $W_{\text{new}}^{A^-} = W^{A^-}$	<ol style="list-style-type: none"> <li>Shrink <math>W^{A^-}</math> by <math>u_0^B</math> to obtain <math>W_{\text{shrunk}}^{A^-}</math></li> <li>Dilate <math>W_{\text{shrunk}}^{A^-}</math> by <math>u_0^A</math> to obtain <math>W_{\text{dilated}}^{A^-}</math></li> <li><math>\forall x \in Q</math>, if <math>f(x)</math> falls in <math>W_{\text{dilated}}^{A^-}</math>, set <math>W_{\text{new}}^{A^-}(x) = 1</math></li> <li><math>W^{A^-} = W_{\text{new}}^{A^-}</math>. Go to step 1 and repeat the process</li> </ol>
$A^+$	$W^{A^+}(x) = \begin{cases} 0 & x \in Q \\ 1 & \text{otherwise} \end{cases}$ $W_{\text{new}}^{A^+} = W^{A^+}$	<ol style="list-style-type: none"> <li>Dilate <math>W^{A^+}</math> by <math>u_0^A</math> to obtain <math>W_{\text{dilated}}^{A^+}</math></li> <li>Shrink <math>W_{\text{dilated}}^{A^+}</math> by <math>u_0^B</math> to obtain <math>W_{\text{shrunk}}^{A^+}</math></li> <li><math>\forall x \in Q</math>, if <math>f(x)</math> falls in <math>W_{\text{shrunk}}^{A^+}</math>, set <math>W_{\text{new}}^{A^+}(x) = 1</math></li> <li><math>W^{A^+} = W_{\text{new}}^{A^+}</math>. Go to step 1 and repeat the process</li> </ol>
$B^-$	$W^{B^-}(x) = \begin{cases} 1 & x \in Q \\ 0 & \text{otherwise} \end{cases}$ $W_{\text{new}}^{B^-} = W^{B^-}$	<ol style="list-style-type: none"> <li>Shrink <math>W^{B^-}</math> by <math>u_0^A</math> to obtain <math>W_{\text{shrunk}}^{B^-}</math></li> <li>Dilate <math>W_{\text{shrunk}}^{B^-}</math> by <math>u_0^B</math> to obtain <math>W_{\text{dilated}}^{B^-}</math></li> <li><math>\forall x \in Q</math>, if <math>f(x)</math> falls in <math>W_{\text{dilated}}^{B^-}</math>, set <math>W_{\text{new}}^{B^-}(x) = 1</math></li> <li><math>W^{B^-} = W_{\text{new}}^{B^-}</math>. Go to step 1 and repeat the process</li> </ol>
$B^+$	$W^{B^+}(x) = \begin{cases} 1 & x \in Q \\ 0 & \text{otherwise} \end{cases}$ $W_{\text{new}}^{B^+} = W^{B^+}$	<ol style="list-style-type: none"> <li>Dilate <math>W^{B^+}</math> by <math>u_0^B</math> to obtain <math>W_{\text{dilated}}^{B^+}</math></li> <li>Shrink <math>W_{\text{dilated}}^{B^+}</math> by <math>u_0^A</math> to obtain <math>W_{\text{shrunk}}^{B^+}</math></li> <li><math>\forall x \in Q</math>, if <math>f(x)</math> falls in <math>W_{\text{shrunk}}^{B^+}</math>, set <math>W_{\text{new}}^{B^+}(x) = 1</math></li> <li><math>W^{B^+} = W_{\text{new}}^{B^+}</math>. Go to step 1 and repeat the process</li> </ol>

**Table 7.2.** Morphological operations for each player's winning set computation.

the process progresses, the winning sets evolve with each iteration: player  $A$ 's set grows, while player  $B$ 's set shrinks. Additionally, two controlled trajectories are shown. The first one, in Fig. 7.2(b), starting in player  $A$ 's winning set, where the controller successfully pushes the trajectory out of region  $Q$ . For the second one, in Fig. 7.2(c), the trajectory starts in player  $B$ 's winning set, so they can keep the trajectory within the region.

Figure 7.3 displays the outcomes of three different examples of games, depending on whether each player has knowledge of the other's actions. In Fig. 7.3(a), player  $A$  moves first, followed by player  $B$ , therefore player  $A$  has advantage by knowing player  $B$ 's actions. Figure 7.3(b) reverses the order, with player  $B$  moving first and player  $A$  second. Finally, Fig. 7.3(c) shows the case where both players are unaware of each other's moves, representing a simultaneous play situation.

This last game is noteworthy. In this case, regions in the state space appear

that do not belong to any winning set. These areas arise because neither player can secure a win without considering the opponent's strategy. This uncertainty stems from the fact that each player is unaware of the other's choice when making their decision. As a result, the outcome depends on the specific combination of moves made by both players, and the players may not always respond optimally to the opponent's move.

This uncertainty stands in contrast to the games in Fig. 7.3(a) and (b), where the presence of informed players leads to clear winning sets, leaving little ambiguity about the game's outcome for all initial conditions.

### 7.4.1 Exploring possible games in the $(u_0^B, u_0^A)$ space

The results of these games are heavily influenced by the range of control available to each player. Figure 7.4 offers a detailed exploration of potential game scenarios within the parameter space  $(u_0^B, u_0^A) \in [0, 0.2] \times [0, 0.1]$ . It distinguishes between three types of games:  $A$  ignorant vs  $B$  informed,  $A$  informed vs  $B$  ignorant, and both players ignorant.

In these diagrams, certain regions represent scenarios where one player is guaranteed to win regardless of the starting conditions, while other regions show outcomes that depend on the initial setup. In particular, when both players lack information (Fig. 7.4(c)), there are regions labeled as  $NW$  (no winner) that represent cases where the outcome is indeterminate for certain starting conditions.

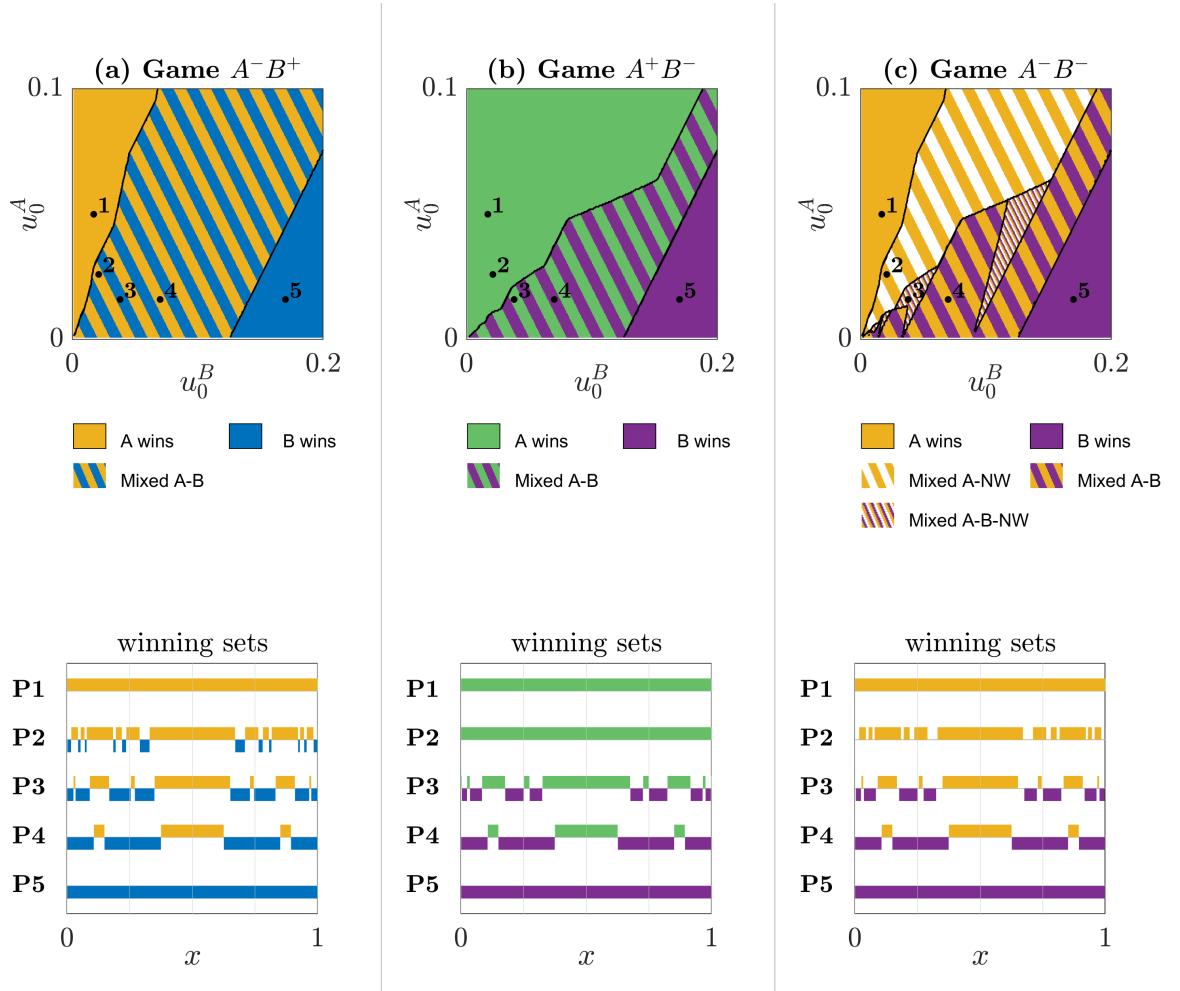
The lower panels (Figs. 7.4(d-f)) provide concrete examples of winning sets for five different parameter combinations, referred to as games  $P1$  through  $P5$ . These examples illustrate how winning sets change as the parameters vary across the identified regions of the parameter space.

Figure 7.5 further analyzes these findings by showing the proportion of region  $Q$  occupied by each player's winning set. Notably, Fig. 7.5(c) highlights the percentage of initial conditions where neither player can secure a guaranteed win when both are unaware. The color gradients in this figure depict a gradual shift in game outcomes as control bounds change.

These findings emphasize how the relative strength of the players' control bounds shapes not only who emerges victorious but also whether a definitive victory is possible. When the control bounds are closely matched, the results often depend on initial conditions or remain uncertain.

### 7.4.2 Boundary games

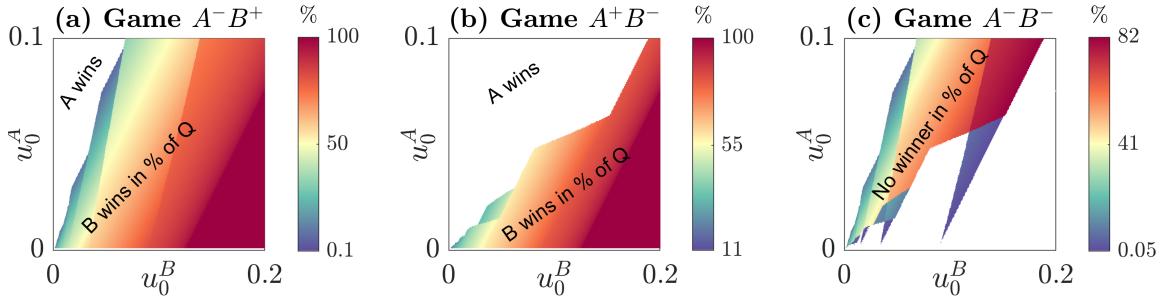
The dynamics of the game inherently favor player  $A$ , whose objective is to exit region  $Q$ , a process that occurs naturally due to the transient nature of the system. However, this does not mean player  $B$  is without hope. With sufficient control, at some cases even with control lower than that of player  $A$ , player  $B$  can still achieve their goal. The boundary games depicted in Fig. 7.6 examine the critical



**Figure 7.4.** Different solutions depending on the values of  $u_0^A$  and  $u_0^B$ . Panels (a), (b), and (c) show a diagram in which solid colored regions represent parameter combinations where the informed/ignorant player  $A$  wins for all initial conditions at solid colors green/yellow and the informed/ignorant player  $B$  at solid blue/purple. Striped regions indicate parameters where the winner depends on the initial conditions. There are also regions with white strips marked as  $NW$  (no winning), where the victory is uncertain at some initial conditions. Below each diagram we show the winning sets, i.e., the initial conditions that guarantee victory for each players, for each kind of game at 5 different regions of control bound. These points, written as  $(u_0^B, u_0^A)$ , are:  $P1 = (0.017, 0.050)$ ,  $P2 = (0.021, 0.026)$ ,  $P3 = (0.038, 0.016)$ ,  $P4 = (0.070, 0.016)$ , and  $P5 = (0.170, 0.016)$ .

limits of control at which player  $B$ 's victories begin to emerge. This boundary, illustrated in Figs. 7.6(a) and 7.6(c) within the  $(u_0^B, u_0^A)$  parameter space, marks a pivotal transition, crossing this boundary creates initial conditions where  $B$ 's victory becomes possible.

The structure of these boundaries is particularly noteworthy, as shown in Figs. 7.6(b)



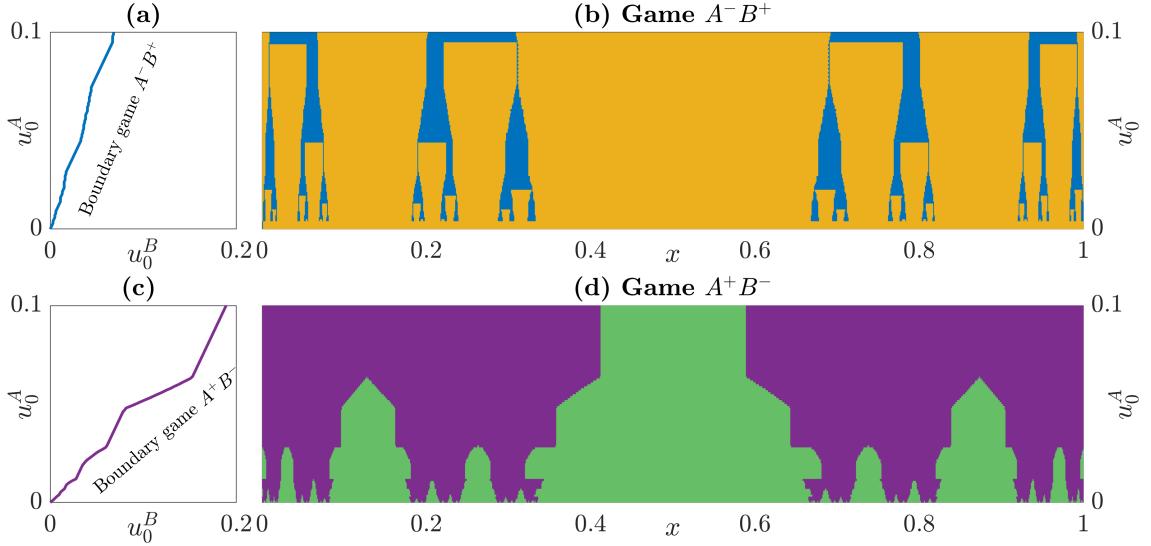
**Figure 7.5.** The first two panels show the percentage of the region that occupies the winning set of player  $B$  respect region  $Q$  whether they are informed (+) or ignorant (-). The color is white when the percentage is zero. The last panel represents the percentage of initial conditions that do not assure a winner when both players are ignorant. All figures show similar details at different scales, suggesting self similarity in the system dynamics.

and 7.6(d), which present the winning sets along these transitions for two distinct game types. In the case of player  $A$  ignorant vs player  $B$  informed (Fig. 7.6(b)), we see a highly intricate pattern of victories for player  $B$  represented as blue regions surrounded by the domain of player  $A$ . These patterns exhibit increasing detail at finer scales, hinting at a self-similar structure. By contrast, in the scenario where player  $A$  is informed and player  $B$  is ignorant, depicted in Fig. 7.6(d), the arrangement takes on a different form. The winning regions of player  $B$ , in purple, are interspersed with the ones of player  $A$ , in green, in a complex but distinct alternating pattern.

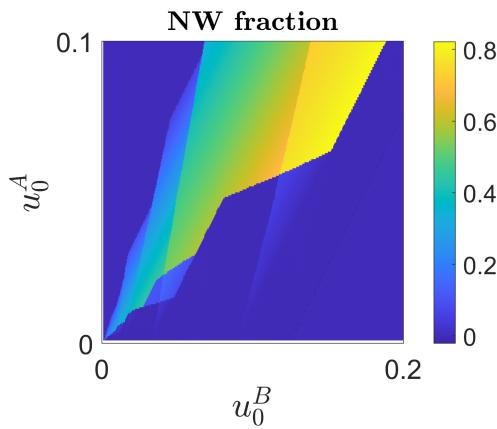
### 7.4.3 Is it worth being informed?

The advantage of playing second in this game is evident because they have the knowledge of the opponent's actions, as this allows them to counteract any move the rival makes. Consequently, the ignorant sets are strictly contained within the informed sets. Interestingly, when plotting the difference between the proportions of each player's informed and ignorant sets (Fig. 7.7), we observe a large region where this difference is zero. Notably, this region corresponds to the same area in Fig. 7.5 where no player can guarantee victory. This relationship is clearly demonstrated by the following equations:

$$\left. \begin{array}{l} A^+ + B^- = 1 \\ A^- + B^+ = 1 \\ A^- + B^- + NW = 1 \end{array} \right\} \Rightarrow NW = A^+ - A^- = B^+ - B^- \quad (7.4)$$



**Figure 7.6.** Panels (a) and (c) show the game boundary for which there is a shift between player  $A$  winning the game at all initial conditions (left of the line) and there is a mixed victory among different initial conditions (right of the line). This lines are the boundaries of panels (a) and (b) from the previous figure. Panels (b) and (d) show the winning sets within the control bounds along the game boundary. In colors yellow and blue the points in the  $A$  ignorant and  $B$  informed winning sets respectively and in green and purple,  $A$  informed and  $B$  ignorant. In panel (b) we can clearly see increasing and almost self-similar details when the scales decreases.



**Figure 7.7.** The color on the heatmap represents the difference between the informed and ignorant sets fraction. The no winning fraction ( $NW$ ) satisfies the relation  $NW = A^+ - A^- = B^+ - B^-$ , that is, the difference is the same for both players. Therefore they will fight for being the one informed at the same parameter regions. However for many points the difference is 0, so being informed or being ignorant results in the same winning set.

Here,  $P^+$  represents the proportion of region  $Q$  occupied by the winning set of the informed player  $P$ , while  $P^-$  denotes the same proportion for the ignorant player. The term  $NW$  (no winning) corresponds to the proportion of blank space when both players are ignorant.

If players could compete for the right to act as the informed player, it is reasonable to assume that there would be a cost associated with gaining this advantage. In such a scenario, it may not always be worthwhile to incur the cost of being informed, because in a significant portion of cases, the outcome remain unchanged. However, if minimizing the average control effort is a priority, being informed offers a substantial advantage in nearly all situations. We have checked that the minimal control required to obtain the same results is lower for informed players.

## 7.5 Discussion and conclusions

We proposed a novel two-player game of survival in a transient chaotic region. In the game, two players are confronted against each other to control the trajectory of a chaotic dynamical system. The players had opposing goals, each one aiming to get the trajectory to different regions. Through the partial control method we got the set of initial points that guaranteed the victory for each player. Through partial control algorithms we were able to construct winning sets that provided the initial conditions where each player had victory assured. Then, the controlled trajectories are not unique, but at each iteration of the game, the system can be controlled to a range of possible points within the winning sets, giving more flexibility to the controllers.

As an example for our game, we analyzed the logistic map. Here one player aims to stay at the transient chaotic region and the rival wants to drive the trajectory out of there. This system unveils an interesting aspect about the complexity of the dynamical system. Because even when the dynamical system plays against the conservative player and their opponent has greater control capabilities, the player can thrive and achieve their objective.

We also found that the information each player has plays a substantial role in the game. The player that plays first, and therefore knows the action of the rival, will undoubtedly be at an advantage. Therefore, this knowledge was crucial to victory at some cases, but at some other cases, surprisingly, the information was of little use to the informed player.

Finally another interesting result is that when no player knows the action of their rival, regions appear where victory is uncertain. The game is unresolved there and the outcome will depend on the sequence of actions of the players.

## Bibliography

- [1] J. Aguirre, F. d’Ovidio, and M. A. F. Sanjuán, Controlling chaotic transients: Yorke’s game of survival. Phys. Rev. E 69:016203 (2004). <https://doi.org/>

- 10.1103/PhysRevE.69.016203
- [2] Juan Sabuco, Miguel A. F. Sanjuan, and James A. Yorke Dynamics of partial control Chaos 22:047507 (2012). <https://doi.org/10.1063/1.4754874>
  - [3] R. Capeáns, J. Sabuco, and M. A. F. Sanjuán, Beyond partial control: controlling chaotic transients with the safety function. Nonlinear Dyn. 107:2903–2910 (2022). <https://doi.org/10.1007/s11071-021-07071-1>
  - [4] R. Capeáns, J. Sabuco, and M. A. F. Sanjuán, A new approach of the partial control method in chaotic systems. Nonlinear Dyn. 98:873–887 (2019). <https://doi.org/10.1007/s11071-019-05215-y>
  - [5] P. Kollock, Social dilemmas: the anatomy of cooperation. Annu. Rev. Sociol. **24**, 183–214 (1998). <https://doi.org/10.1146/annurev.soc.24.1.183>
  - [6] Y. Xiao, Y. Peng, Q. Lu, and X. Wu, Chaotic dynamics in nonlinear duopoly Stackelberg game with heterogeneous players. Physica A **492**, 1980–1987 (2018). <https://doi.org/10.1016/j.physa.2017.11.112>
  - [7] W. Hu, G. Zang, H. Tian, and Z. Wang, Chaotic dynamics in asymmetric rock-paper-scissors games. IEEE Access **7**, 175614–175621 (2019). <https://doi.org/10.1109/ACCESS.2019.2956816>
  - [8] E. Akiyama and K. Kaneko, Dynamical systems game theory and dynamics of games, Physica D **147**, 221–258 (2000).
  - [9] E. Akiyama and K. Kaneko, Dynamical systems game theory II A new approach to the problem of the social dilemma Physica D **167**, 36–71 (2002). [https://doi.org/10.1016/S0167-2789\(00\)00157-3](https://doi.org/10.1016/S0167-2789(00)00157-3)
  - [10] J. R. Marden and J. S. Shamma, Game Theory and Control. Annu. Rev. Control. Robotics Auton. Syst. **1**, 105–134 (2018). <https://doi.org/10.1146/annurev-control-060117-105102>

# **Chapter 8**

# **Results and Discussion**

*"Strange game. The only winning move is not to play."*

-Lawrence Lasker and Walter F. Parkes,  
WarGames

We have studied a specific evolutionary game, the public goods game, and introduced modifications to examine time-dependent effects that occur in real-life scenarios. Both effects, oscillations in return efficacy and delays in punishing defectors, were found to hinder cooperation. This is because cooperators require a certain level of stability to ensure their efforts are worthwhile. Additionally, delayed punishment is less effective, as immediate consequences are more likely to achieve the intended deterrent effect.

Then we have measured complexity in the prisoner's dilemma and the public goods game with the Hamming distance metric. This has helped us understanding the chaotic pattern formations present when simulating the games spatially, with local interactions. We have obtained a measure that may correlate to the Lyapunov time. This measure tells us how much time passes until two configurations are significantly different, and also help us determining the velocity of propagation of changes in the configurations of cooperators and defectors.

Each of the analyzed games can be seen as an enormous cellular automata, and therefore, we have also analyzed the complexity of the simplest of them to understand the roots of the problem. These are the elementary cellular automata, which can be divided in 4 classes.

We have obtained an analogue classification to that of Wolfram, but ours focuses on the behavior of the Hamming distance of two close configurations. Class 1 consist of cellular automata that nullifies the difference between two configurations, while in class 2 the difference becomes too small and does not change over time. Therefore, we can say that the cellular automata in these two classes are stable. On the other hand, cellular automata in classes 3 and 4 are unstable, with changes rapidly

propagating. In class 3 the Hamming distance behaves chaotically representing the random like patterns that form when plotting the cells' states while in class 4 the Hamming distance presents transient chaos, which explains the transition between complex patterns to periodical ones but with large periods characteristic of the cellular automata in this class.

By studying a control-based game, we found that success largely depends on the effective use of information. Players competing in the game have a higher chance of winning if they possess knowledge of their opponent's moves and both players' control capabilities. Additionally, a precise understanding of the system in which they are playing is crucial.

A surprising fact of the game is that the player that has the most challenging goal, to stay inside a transient chaotic region, can sometimes do so even when their control capabilities are inferior to those of their opponent.

Before studying that game, we have familiarized with the partial control, by extending the method to the control of escaping trajectories in a quick or in an orderly way. This study was also used when developing the algorithms to solve the competitive game of control.

Through the interconnected study of game dynamics, complexity and control of chaos, we have obtained a broader understanding on how seemingly simple rules can create rich and complex behaviors. Nonetheless, through a precise understanding of the system in play, players can take advantage of the characteristics of the game dynamics and achieve their goals. Therefore, the study of game dynamics through tools that come from the complex systems domain, provides powerful insights that can be used to solve many real-world problems.

I hope that the breakthroughs and methods presented in this doctoral thesis will contribute to future research in the fields of game theory, complex systems, and control.

# **Chapter 9**

# **Conclusions**

Here, we schematically summarize the main conclusions of the thesis:

- We have studied two alterations of the public goods game with punishment. We concluded that when investing among peers, instability in the returns, characterized by an oscillation on the enhancement factor, hinders cooperation. Moreover, the delayed punishment of defectors also makes defectors thrive.
- We have analyzed the complexity in spacial evolutionary games, even though the system is globally stable since no significant variations of strategy frequencies occur, the local dynamics can be unstable. We obtained a method to measure the complexity of this interactions through the study of the divergence of the Hamming distance of two initially close configurations. Thanks to this tool we arrived at the typical system's time for the two configurations to be far apart.
- We correctly classified all elementary cellular automata with the Hamming distance measure of diverging configurations. The first two classes are stable to local variations while Class-3 and 4 are unstable. The distance in Class-3 behaves chaotically while Class-4 presents transitory chaos underlying the phenomenon of edge of chaos.
- We found a new application to partial control in order to expel trajectories from a transient chaotic region. We have been able to do so with two different approaches. The first one, by accelerating the escape for the trajectories to be expelled in the least time possible. Second, by controlling the escape in order to know how much time passes since we eject control until the trajectory has escaped. Furthermore, we controlled trajectories in multistable systems by setting the time the orbit stays at each region, instead of chaotically shifting to one another.
- Through the control partial analysis we obtained a useful tool in game theory. We designed a game of control and survival between two confronted players. With the partial control tool helps players can make decisions that gives them the victory at certain initial conditions. We analyzed different cases regarding how much information each player has.



# Resumen de la tesis en castellano

*"Que la realidad te parezca absurda es la mayor crítica que le puedes hacer"*

-Laura Fernández

## 9.1 Introducción

Esta tesis trata sobre juegos evolutivos, complejidad y control del caos. Estos temas, que pueden parecer dispares, han sido interconectados en los diferentes capítulos presentes aquí. El hilo conductor entre las distintas disciplinas es que todos han sido estudiados mediante herramientas propias del análisis de los sistemas complejos y la dinámica no lineal.

La Teoría de Juegos Evolutiva (Evolutionary Game Theory, EGT) es analizada profundamente en ramas de la física por su dinámica compleja. Al ser un estudio más adecuado para grandes poblaciones, los sistemas estudiados con estas técnicas suelen tener propiedades emergentes. Estas propiedades emergentes no pueden ser explicadas considerando únicamente a los individuos que forman el sistema, sino que hay que tratar con la población entera para que aparezcan y poder entenderlos. Como consecuencia, estos sistemas suelen tener propiedades no lineales muy interesantes para la física de sistemas complejos.

Al principio de la tesis se ha estudiado el juego de los bienes públicos. En este juego social, los jugadores tratan de hacer grupos con los que invertir para aumentar sus recompensas. Cada individuo puede elegir cooperar con una unidad monetaria al bien público o abstenerse. Indiferentemente a si han cooperado o no, todos los jugadores se reparten el resultado de la inversión, con su debido aumento gracias a las ganancias, al final de cada ronda. Según la estrategia que adapta cada individuo y la que adaptan sus vecinos, este obtiene un *payoff*, es decir, un saldo que determina la aptitud o *fitness* del individuo en el juego. Mediante un proceso evolutivo consistente en la imitación del individuo más apto, el sistema evoluciona según se iteran turnos del juego.

Aparte de las dos estrategias ya mencionadas, cooperar o no hacerlo, se ha añadido una tercera estrategia. Existen individuos que, aparte de cooperar, pagan

una tasa adicional para castigar a los individuos que no cooperan. Los impagadores que sean detectados por estos castigadores deberán pagar una multa, reduciendo así su *payoff*.

En la investigación del primer capítulo se ha analizado la variación en el juego de los bienes públicos al introducir componentes dependientes del tiempo. Primero se ha considerado una oscilación en el parámetro que controla la eficacia de las inversiones. Esta oscilación trata de imitar las fluctuaciones propias de los mercados o de la naturaleza. Se ha observado que esta oscilación afectaba negativamente a la cooperación entre individuos.

También se ha investigado el efecto de introducir un retardo a la hora de castigar a los impagadores. Este retardo causa que los impagadores aumenten, demostrando así que los castigos más eficaces son aquellos que son inmediatos.

Tras haber resuelto estas cuestiones nos propusimos analizar la complejidad del juego, además de otro juego social, el dilema del prisionero. Otros estudios demuestran que se producen comportamientos caóticos espacio-temporales considerando la dinámica de formación de patrones cuando se observa la disposición de cooperadores y no cooperadores en el espacio.

Con la ambición de cuantificar esta complejidad, hemos considerado la divergencia de la distancia de Hamming entre dos configuraciones inicialmente muy próximas. Esta distancia, usada comúnmente en ciencias de la computación, establece la diferencia en número de caracteres entre dos grupos del mismo tamaño. Observando su comportamiento hemos podido observar que la distancia crece cuando tratamos con parámetros del sistema en los que existen comportamientos caóticos.

En el caso en el que no se utilice ningún componente aleatorio, los juegos anteriormente estudiados se pueden considerar como autómatas celulares. Eso sí, el número de reglas que tendrían estos autómatas celulares sería muy grande. En concreto el autómata tendría  $2^N$  reglas, siendo  $N$  el número de vecinos que influyen en el *payoff* de cada agente. Debido a esto, se ha visto conveniente analizar la complejidad de todos los autómatas celulares elementales. Además se ha elaborado una clasificación análoga a la de Wolfram para estos autómatas. Esta clasificación depende del comportamiento de la distancia de Hamming entre dos configuraciones muy próximas al principio.

En otra línea diferente de la investigación se extendió el estudio del control parcial para controlar el escape de una trayectoria de su región caótica transitoria. El control parcial es un método de control en el que se evita ciertas zonas del espacio de fases de un sistema. De esta manera, por ejemplo, en un sistema con dinámicas de caos transitorio, se puede conseguir que la trayectoria permanezca en la región caótica transitoria de manera indefinida. En uno de los estudios realizados en esta tesis se ha considerado exactamente lo contrario. Es decir, hemos tratado de sacar al sistema de esa región controladamente utilizando las mismas herramientas del control parcial.

El sistema naturalmente escapa de la región, pero lo que hemos querido hacer es tratar de hacerlo de manera controlada. Para ello nos propusimos dos objetivos:

hacer que la trayectoria escape lo más rápidamente posible, o que escape en un número de iteraciones a elegir por el controlador.

Tras conseguirlo con éxito nos fijamos un tercer objetivo más ambicioso: hacer que una trayectoria en un sistema multiestable pase de una región a otra de manera controlada. De esta forma, las trayectorias que, sin control transitan de manera caótica de una región a otra, permanecerán en cada región el tiempo que decidamos.

Por último se ha diseñado un juego de supervivencia entre dos jugadores enfrentados entre sí. Cada jugador tiene un objetivo distinto, permanecer en diferentes zonas del espacio de fases. Hemos ilustrado el juego con un ejemplo, el mapa logístico. Debido a la naturaleza transitoria de este sistema, este mapa se plantea muy interesante. Cuando establecemos que un jugador tiene como objetivo escapar de la región caótica transitoria y el oponente trata de permanecer en esa zona, el jugador que quiere escapar tiene ventaja. Las trayectorias naturalmente escaparán dándole la victoria. Gracias a las técnicas de control parcial, hemos obtenido las condiciones iniciales para las que cada jugador tiene asegurada la victoria, los sets de victoria, *winning sets*.

Con las herramientas y descubrimientos realizados durante el transcurso de esta tesis doctoral, esperamos haber contribuido a las ramas de los juegos evolutivos, los autómatas celulares y el control del caos, y haber facilitado futuras investigaciones en estos temas.

## 9.2 Metodología

Para realizar esta tesis hemos empleado un análisis teórico y computacional. Con ayuda de métodos analíticos u simulaciones numéricas, resolvemos y corroboramos predicciones teóricas. A continuación describimos los métodos utilizados.

Se han realizado simulaciones Monte Carlo para modelos basados en agentes. De esta manera tenemos un modelo evolutivo de individuos que juegan a juegos sociales. Hemos utilizado una red cuadrada en la que cada agente está colocado y juega con sus vecinos inmediatos. Según el método de Monte Carlo, se eligen individuos al azar para que adopten la estrategia de un vecino aleatorio. Si el *payoff* del vecino es mayor, es probable que adopte su estrategia. Las simulaciones han sido realizadas en el lenguaje de programación *Julia*.

En un momento de la investigación hemos diseñado un mapa personalizado para aplicarle el método de control parcial. Para caracterizar la dinámica de este nuevo mapa se ha realizado un diagrama de bifurcación, que determina la estabilidad del mapa ante diferentes valores de los parámetros que determinan el sistema, en nuestro caso, un solo parámetro.

Además se obtuvo el máximo exponente de Lyapunov para el mapa. Este exponente cuantifica la naturaleza caótica de un sistema. Cuando el exponente toma valores positivos se considera que sus órbitas serán caóticas, y cuando el exponente es negativo o nulo, el sistema es estable, caracterizado por órbitas periódicas. El máximo exponente de Lyapunov se puede obtener con la siguiente fórmula cuando

el sistema es discreto en el tiempo.

$$\lambda(x_0) = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=0}^{n-1} \log \left| \frac{df(x_i)}{dt} \right|.$$

El diagrama de bifurcación y el máximo exponente de Lyapunov fueron computados con *MATLAB*.

También se ha aprovechado otra herramienta que permite cuantificar el comportamiento caótico. Este método ha sido escasamente utilizado en la literatura, pero se nos ofrece muy útil en nuestra investigación. Es un algoritmo que utiliza la distancia de Hamming para medir divergencias en los patrones de distribución de diferentes estrategias en juegos o estados de un autómata celular. Se examinaron dos configuraciones inicialmente muy próximas y medimos la separación a medida que el tiempo de la simulación iba aumentando.

### 9.3 Resultados y Discusión

En esta tesis hemos realizado aportaciones en los campos de dinámica evolutiva de juegos sociales con la que hemos ganado entendimiento en temas como la cooperación entre individuos. Al hacer oscilar un parámetro que controla la efectividad de inversiones económicas observamos lo siguiente. Cuando el parámetro aumenta hay un aumento de los cooperadores, pero cuando el parámetro disminuye, los cooperadores flaquean y aumenta mucho más el número de impagadores. Este efecto puede ser explicado fácilmente, ya que una pobre estabilidad en los rendimientos de una inversión la hace más arriesgada, y, por tanto, menos apetecible. Además se ha comprobado que un castigo tardío no es tan eficaz como un castigo más inmediato.

También hemos hecho progresos relacionados con la cuantificación de la complejidad en sistemas que presentan caos espacio-temporal. Hemos sido capaces de analizar la complejidad en la dinámica de patrones de disposición de diferentes estrategias para dos juegos sociales: el juego de los bienes públicos y el dilema del prisionero. Ambos resultaron presentar complejidad tras medir la divergencia entre configuraciones muy parecidas al principio. Se ha observado que la distancia de Hamming entre las dos configuraciones crece rápidamente para aquellos parámetros en los que se observa comportamiento caótico espacio-temporal.

Además tras el estudio de la distancia de Hamming en autómatas celulares se obtuvo una clasificación análoga a la de Wolfram. Para la Clase-1 la distancia era nula y para la Clase-2 la distancia permanecía constante, lo cual muestra que los autómatas de ambas clases son sistemas simples. Esto se puede apreciar observando los patrones uniformes que forman los autómatas de estas clases. En cambio, en la Clase-3, al mostrar la distancia de Hamming en función del tiempo, esta se comporta de manera caótica. Esto explica los patrones que parecen aleatorios que se observan al representar los estados de estos autómatas celulares. La última clase, la Clase-4, también se comporta de manera caótica, pero al cabo de un tiempo alcanza un

equilibrio. La distancia tomará un valor fijo o formará una oscilación periódica. Este comportamiento caótico transitorio de la Clase-4 se podría relacionar con el fenómeno de *edge of chaos* (límite del caos), y se manifiesta en la combinación de zonas con cierto orden y otras zonas complejas al contemplar los estados del autómata.

La investigación referente a controlar trayectorias para que escapen rápidamente u ordenadamente con control parcial fue exitosa. Se obtuvo el control necesario para expulsar las trayectorias y, para algunas condiciones iniciales, el control resultó ser menor que el ruido propio del sistema. Además se consiguió transformar trayectorias caóticas en quasi-periódicas en sistemas multiestables.

Finalmente se diseño un juego entre dos controladores y fue resuelto mediante técnicas de control parcial. En el juego, uno de los controladores tiene el objetivo de permanecer en una región caótica transitoria y el otro hace todo lo posible por expulsar la trayectoria de allí. Se obtuvieron los sets de victoria, esto es, las condiciones iniciales en las que cada controlador consigue su objetivo.

En el caso de que un jugador conozco los movimientos del rival antes de ejercer su control, los sets de ambos jugadores son complementarios, llenando así todo el espacio de fases. En cambio, si ambos jugadores desconocen las acciones del oponente, puede haber casos en el que queden huecos en los que ningún controlador tenga asegurada la victoria. Es importante resaltar que esto no siempre ocurre, y a veces no hay diferencia entre estar informado y no estarlo. Esto se debe a que las zonas donde el jugador tiene asegurada la victoria son exactamente iguales cuando el controlador está informado que cuando no. Sin embargo, cuando el controlador sí está informado, el valor de control que tiene que hacer es menor.

Un resultado sorprendente es el hecho de que el jugador que está en desventaja, debido a la dinámica natural del mapa con caos transitorio, pude conseguir su objetivo en algunos casos incluso con una cota de control menor que la del oponente.

### 9.3.1 Conclusiones

Las principales conclusiones del estudio realizado en la tesis doctoral son:

- En el estudio de los efectos dependientes del tiempo en el juego de los bienes públicos se determinó que la inestabilidad de las inversiones, caracterizada por una oscilación en el parámetro que multiplica las aportaciones de los individuos, perjudica a la cooperación entre estos para invertir conjuntamente. Además, un castigo tardío a los impagadores favorece su crecimiento.
- Hemos obtenido una herramienta que cuantifica comportamientos caóticos espacio-temporales de juegos en los que los individuos se reparten de manera espacial. Aunque el sistema esté en equilibrio globalmente, es decir, la frecuencia de poblaciones no se altera significativamente con el tiempo, la dinámica local puede ser inestable. De esta manera cambios mínimos en las configura-

ciones iniciales de estrategias producen cambios completamente diferentes tras un tiempo característico del sistema, que hemos podido medir.

- Hemos clasificado correctamente todos los autómatas celulares elementales mediante el uso de la misma herramienta. Las Clases-1 y 2 resultan presentar estabilidad mientras que las Clases-3 y 4 presentan comportamiento caótico. En la Clase-3 los autómatas producen caos para siempre mientras que en la Clase-4 se ha observado caos transitorio.
- Hemos elaborado una nueva aplicación al control parcial para controlar el escape de trayectorias de una zona con caos transitorio. Se consiguen expulsar a las trayectorias de la manera más rápida posible o de una manera controlada, sabiendo el tiempo que va a pasar hasta que escape. Además, se ha conseguido crear trayectorias que transitan de una zona a otra periódicamente en sistemas caóticos multiestables.
- Usando técnicas de control parcial conseguimos resolver un juego novel de control y supervivencia entre dos jugadores. Se obtuvieron las regiones en las que cada jugador gana el juego dependiendo de la información que cada uno posee.