

Article

On the Complexity of Stable and Biased Majority

Pablo Concha-Vega, Eric Goles, Pedro Montealegre * and Martín Ríos-Wilson

Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Santiago 7941169, Chile

* Correspondence: p.montealegre@uai.cl

Abstract: A *majority automata* is a two-state cellular automata, where each cell updates its state according to the most represented state in its neighborhood. A question that naturally arises in the study of these dynamical systems asks whether there exists an efficient algorithm that can be implemented in order to compute the state configuration reached by the system at a given time-step. This problem is called the *prediction problem*. In this work, we study the prediction problem for a more general setting in which the local functions can be different according to their behavior in tie cases. We define two types of local rules: the *stable* majority and *biased* majority. The first one remains invariant in tie cases, and the second one takes the value 1. We call this class the *heterogeneous majority cellular automata* (HMCA). For this latter class, we show that in one dimension, the prediction problem for HMCA is in **NL** as a consequence of the dynamics exhibiting a type of bounded change property, while in two or more dimensions, the problem is **P-Complete** as a consequence of the capability of the system of simulating Boolean circuits.

Keywords: cellular automata; majority rule; prediction problem**MSC:** 68Q80

Citation: Concha-Vega, P.; Goles, E.; Montealegre, P.; Ríos-Wilson, M. On the Complexity of Stable and Biased Majority. *Mathematics* **2022**, *10*, 3408. <https://doi.org/10.3390/math10183408>

Academic Editor: Adolfo Ballester-Bolínches

Received: 14 August 2022

Accepted: 15 September 2022

Published: 19 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Motivation. A *majority automata* can be defined as a two-state cellular automata (namely with states -1 and 1) where in each synchronous step, each site takes the most represented state in its neighborhood. The majority rule is one of the simplest defined rules, and appears naturally in models from physics, biology, social phenomena, and elections systems [1–6]. For that reason, a common task consists of computationally simulating the dynamics that arise given an initial finite configuration. A natural question asks what are the most efficient algorithms that can be implemented in order to compute the state configuration reached at a given time-step. This problem is known as the *prediction problem*.

The computational complexity of a computational task can be defined as the amount of resources, such as time or space, needed to solve it. One fundamental set of computational decision problems is the class **P**, which is the class of problems solvable in polynomial time on a deterministic Turing machine. The class **P** is informally known as the class of problems that admit an efficient algorithm.

Evidently, the prediction problem can be solved by simply simulating the dynamics of the cellular automata for the given number of time-steps. This is called the *trivial algorithm*. In [7] Goles et al. showed that the dynamics of the majority automaton reaches an attractor in a number of time-steps that is linear in the size of the configuration. Moreover, the reached attractor is either a fixed-point or a limit cycle of period two. This result implies that the prediction problem can be solved in polynomial time. Therefore, if we aim to solve the prediction problem more efficiently than the trivial algorithm, we would have to classify it in a proper subclass of **P**.

The class of non-deterministic logarithmic space problems **NL** is the class of decision task solvable by a non-deterministic logarithmic-space Turing machine. It is known that $\mathbf{NL} \subseteq \mathbf{P}$, and it is a widely believed conjecture that the inclusion is proper. The problems in

P that are the most likely to not belong to **NL** are the **P**-Complete problems, to which any other problem in **P** can be reduced by a logarithmic-space reduction. If we could show that some **P**-Complete problem belongs to **NL**, we would deduce that $P = NL$ [8,9].

An example of a **P**-Complete problem is the circuit value problem (CVP), which consists of determining the output of a given circuit with a particular configuration. Further, the monotone circuit value problem (MCVP), i.e., only considering the gates AND and OR, is also **P**-Complete. Intuitively, circuits are hard to compute quickly because in order to compute a given layer of gates, it is necessary to know the state of the previous ones. For a deeper understanding of circuit complexity, see [10–12].

In [9], Moore studied the computational complexity of the majority automata in a d -dimensional lattice. He showed that in three or more dimensions, the prediction problem is as hard as evaluating monotone circuits, which implies that the prediction problem is **P**-Complete. Roughly speaking, this means that in order to compute the state in a given site, the best option is to simulate the dynamics of the automaton until it reaches an attractor. However, Moore suggested that in two dimensions with von Neumann neighborhood, it would be possible to predict the dynamics much more efficiently.

The proof of **P**-completeness given in [9] consists of a series of three dimensional *gadgets* used to simulate the different parts of a monotone circuit. While it is easy to generalize these gadgets to more than three dimensions, the restriction to two dimensions seems a much more (or even impossible) task. In fact, one of the main difficulties is that the two-dimensional grid is a planar graph. The *planar monotone circuit value problem* (PMCVP) is the restriction of MCVP to planar circuits. In [13], it is shown that PMCVP belongs to the class **NC**. This class of problems are widely believed to be properly included in **P**. In addition, it contains the class **NL**. Therefore, intuitively, a proof of **P**-completeness of the two-dimensional majority should somehow manage to simulate wire crossings in a planar topology.

In the same article, Moore also studied the *biased majority automata* (called *half-or-more automata* [9]), which corresponds to the majority automata where the sites do not consider their own state in the neighborhood, and privilege state 1 in tie cases. Moore stated that the prediction problem for non-strict majority automaton is **P**-Complete in three or more dimensions, and also conjectured that in two dimensions, the problem should be solved more efficiently.

Heterogeneous majority. In this article, we study the prediction problem on a non-homogeneous setting. We ask for the computational complexity the non-homogeneous cellular automata that combine cells having the majority rule (which in the following we call *stable majority rule*) and cells having the biased majority rule. Such automata are called the *heterogeneous majority cellular automata* (HMCA). We show that in one dimension, the prediction problem for HMCA is in **NL**, while in two or more dimensions, the problem is **P**-Complete.

Our result is based on two different techniques. For the one-dimensional case, we show that the HMCA has a property that we call *two-composition two-change*. This property means that independently of how we choose the local rules and the initial configuration, a single cell can only change its state twice when we look only to the even time-steps of the dynamic. We combine this observation with a result of [14,15], where it is shown that the prediction problem is in **NL** for a group of one-dimensional cellular automata having a property called *bounded-change*. For these cellular automata, the number of times a node can change its state does not depend on the total amount of cells. Observe that this group contains a well-studied class of rules having a property called *freezing property* in which the cells update its state according to a partial order defined on the alphabet [15].

In two dimensions, we show that the combination of stable and biased majority rules allows to simulate logic gates, as well as crossing gadgets that can be used to simulate non-planar information transmission. This implies that the prediction problem in this case is **P**-Complete.

Context. Our results are in line with a series of articles that aim to understand the computational complexity of the majority rule by studying different variants of the problem.

The freezing majority. The freezing property means that a site in state 1 remains in that state in every future time step. Freezing automata model forest fires [16], infection spreading [17], bootstrap percolation [18] and voting systems [9]. Theoretical facts about those automata can be seen in [14].

The prediction problem for the *freezing stable majority automata* was studied by Goles et al. in [19], where the authors showed that the in two dimensions, the prediction problem for the freezing majority automata is in NC. At the same time, in three or more dimensions, the prediction problem is P-Complete. Later, in [20,21], the authors showed an analogous result for the *freezing biased majority automata*. In both articles, the efficient algorithms are based in some topological properties of the set of *stable* sites, that is to say, the set of nodes initially in -1 that never switch their state. Unfortunately, these topological properties are not preserved when the freezing property is lifted. Hence, is unclear how to use the algorithms for freezing cases in non-freezing cases.

Majority automata networks. Another approach consists of generalizing the majority cellular automata from grids into an arbitrary graph. In that context, two perspectives have been taken in order to show the P-completeness. In [22], it was shown that the prediction problem for the majority rule is P-complete, even when the topology is restricted to planar graphs where every node has an odd number of neighbors. The result is based in a crossing gadget that uses a sort of *traffic light* that restricts the flow of information depending on the parity of the time-step. Then, in [23], it is shown that the prediction problem for the majority rule is P-complete when the topology is restricted to regular graphs of degree 3 (i.e., each node has exactly three neighbors). Both results are valid for the stable and biased majorities, as these rules are equal when the nodes have an odd number of neighbors.

Signed majority. In [24], the authors studied the majority rule in two dimensional grids where the edges have a *sign*. The *signed majority* consists of a modification of the majority rule, where the most represented state in a neighborhood is computed multiplying the state of each neighbor by the corresponding sign in the edge. The authors show that when the configuration of signs is the same on every site (i.e., we have an homogeneous cellular automata) then the dynamics and complexity of the signed majority is equivalent to the standard majority. Interestingly, when the configuration of signs may differ from site to site, the prediction problem is P-Complete.

Asynchronous prediction. A last variant considers the prediction problem under a sequential updating scheme. More precisely, the *asynchronous prediction* problem asks for the existence of a permutation of the cells that produces a change in the state of a given cell in a given time-step. In fact, in [9], Moore suggested in this case it holds a similar dichotomy than in the synchronous case: namely, the complexity in the two-dimensional case is lower than in three or more dimensions. This conjecture was proven in [25], where it was shown that the asynchronous prediction in two dimension is in NC, while it is NP-Complete in three or more dimensions.

Organization of the Article

The article is organized as follows. In Section 2, we begin with the main formal definitions used in the rest of the article. In Section 3, we show that in the one-dimensional case, the heterogeneous majority rule is in NL. Then, in Section 4, we give the proof of the P-completeness of the two-dimensional case. Finally, in Section 5, we give a discussion and research perspectives.

2. Preliminaries

In this section, we give the formal definitions used along the article. We begin by defining the topologies that we consider, namely the one-dimensional, and the two-dimensional grids. We continue defining the stable majority, biased majority and heterogeneous majority cellular automata. We then formally define the Prediction problem.

Cellular Automata and Grids. Cellular automata are discrete dynamical systems defined on a regular grid of cells, where each cell changes its state by the action of a local function

or automata rule, which depends on the state of the cell and the state of its neighbors. In this work, we will consider mainly two classes of grids.

First, we will consider the one-dimensional grid, where the cells are arranged in a line. The neighborhoods in this case are defined by the adjacent cells, that is to say, the left and right neighbors.

Second, we will consider the two-dimensional grid, where the cells are arranged in a rectangle of square cells, and the neighbors of each cell are simply given by the nearest cells. Thus, each cell has 4 neighbors. This definition of neighborhood is known in the literature as a Von Neumann neighborhood. This notion can be generalized to three or more dimensions in the obvious way.

In this model, each cell can only have a finite number of states. In this work, we consider only cellular automata of two states, namely -1 and 1 .

A *configuration* of the grid is a function x that assigns values in $\{-1, 1\}$ to a region in the d dimensional grid. For instance, if $d = 2$, a configuration of the two-dimensional grid is given by a square area of $n \times n$ cells. The value of the cell u in the configuration x is denoted as x_u .

If $u \in \mathbb{Z}^d$, we will refer as $N(u)$ to the neighborhood of the cell u . Depending on the dimensions, a configuration x is considered to be defined over a cycle graph (one-dimensional) or a torus (two-dimensional) by identifying each cell in the boundary of x as a neighbor of the cells placed in the opposite boundary of x . In addition, for a cell $u \in \mathbb{Z}^d$, we call $x_{N(u)}$ the restriction of x to the neighborhood of u . For a cellular automaton, the size of the neighborhood of a cell is uniform, i.e., $|N(u)|$ is the same for each cell. Moreover, we have that $N(u) = N(0) + u$.

Formally, a *cellular automaton* (CA) with states Q and *local function* $f : Q^{|N(u)|} \rightarrow Q$ is a map $F : Q^{n^d} \rightarrow Q^{n^d}$ such that $F(x)_u = f(x_{N(u)})$. We call F the *global function* or the *global rule* of the CA. The dynamic is defined by assigning to the configuration x a new state given by the synchronous update of the local function on x . In addition, it is also possible to define a function F by assigning to each cell a local function. In this case, it is possible to assign different rules to each cell. We will focus, during this paper, on this particular case.

Stable and Biased Majorities. In this article, we focus on the *majority cellular automata*. In this cellular automaton, each cell has one of two possible internal states (usually represented by $\{-1, 1\}$). Each cell will change its state according to a majority local rule. In each time step, this local rule forces each cell to take the most represented state in its neighborhood. Since in this work we focus only on the case in which the amount of neighbors is even (the Von Neumann neighborhood), there are some cases in which exactly one half of the neighbors are in one state and the other half are in the other. In this *tie* case scenario, the rule is not a priori defined. However, in the literature, different approaches are considered to define transitions in tie case scenarios.

The first case is the one known as *stable* majority, in which, in the tie case scenario, the rule preserves the original state of the cell. Thus, for example, if a cell is in state 1 and there is a tie case scenario, it will remain in the state 1. Formally, the stable majority local rule is defined by:

$$f(x)_i = \begin{cases} 1 & \text{if } \sum_{j \in N(i)} x_j > 0, \\ x_i & \text{if } \sum_{j \in N(i)} x_j = 0, \\ -1 & \text{if } \sum_{j \in N(i)} x_j < 0. \end{cases}$$

The second case that is considered in this work is the one of the *biased* majority, in which, in a tie case scenario, a cell changes to a fixed state. Formally, for $s \in \{-1, 1\}$, the 1-biased majority local rule (or simply biased majority rule) is defined by:

$$f^s(x)_i = \begin{cases} 1 & \text{if } \sum_{j \in N(i)} x_j > 0, \\ s & \text{if } \sum_{j \in N(i)} x_j = 0, \\ -1 & \text{if } \sum_{j \in N(i)} x_j < 0. \end{cases}$$

Finally, a d -dimensional *heterogeneous majority automata* corresponds to a function $F : \{-1, 1\}^{\mathbb{Z}^d} \mapsto \{-1, 1\}^{\mathbb{Z}^d}$ such that for every $i \in \mathbb{Z}^d$, $F(x)_i$ is either the stable or the biased majority rule.

The prediction problem. In this paper, we focus on the prediction problem, which is a decision problem such that one asks if the state of a given cell will change after t -time steps for a given time. We provide hereunder a precise definition:

Prediction

• **Input:**

- A periodic configuration $x \in \{-1, 1\}^{[n]^d}$ with $d \geq 1$;
- An assignation of local rules for each cell $(f_i)_{i \in [n]^d}$, defining a global rule F ;
- A time $t \in \mathbb{N}$;
- A cell $i \in \mathbb{Z}^d$.

• **Question:** $F^t(x)_i \neq x_i$?

Bounded change cellular automata. A CA is k -change if the number of state changes of any cell in any orbit is upper bounded by k , formally:

$$\forall x \in \{-1, 1\}^n, \forall i \leq N : |t \in \mathbb{N} : F^{t+1}(x)_i \neq F^t(x)_i| \leq k.$$

A CA is bounded-change if it is k -change for some k . It has been proven in [15] that for any one-dimensional bounded-change CA, the prediction problem is in the NL class.

3. One-Dimensional Case

In this section, we show that when restricted to one-dimension, the heterogeneous majority cellular automata can be efficiently predicted. Indeed, we show that when restricted to $d = 1$, the problem Prediction belongs to NL. In fact, we show a stronger result: we show that if we consider the dynamics given by two consecutive iterations of an arbitrary one-dimensional HMCA, i.e., if we study the dynamics given by F^2 , where F is the global rule of the HMCA, then we have that it defines a bounded-change cellular automata.

Proposition 1 ([15]). *Restricted to one-dimensional bounded-change cellular automata, problem Prediction is in NL.*

Obviously, an HMCA is not bounded-change. For instance, if we take $d = 1$, n is even and we define the configuration x that spatially alternates 1 and -1 , we have that it induces a limit cycle of period two, independently of the local rule (biased or stable) that we choose. More precisely, if $d = 1$ and n is even, we can define for $i \in \{1, \dots, n\}$:

$$x_i = \begin{cases} -1 & \text{if } i \text{ is even} \\ 1 & \text{otherwise} \end{cases}$$

We have that $F^{t+2}(x) = F^t(x)$ for all $t \geq 0$. Nevertheless, the situation is different when we look at the evolution of the rule every two time-steps. More precisely, let F be the global function of a one-dimensional HMCA, and let us call $F^2 = F \circ F$. In the following, we show that F^2 is bounded-change.

Lemma 1. Let F be a one-dimensional HMCA. Then F^2 is a two-change one-dimensional cellular automata.

Proof. For simplicity, in the following, we identify state -1 with 0 . Our proof consists of showing that, if a cell has a state transition from 0 to 1 under F^2 , then this cell will remain fixed in state 1 on every following iteration of F^2 . This means that, in total, a cell can have at most two state transitions under F^2 .

Let us denote by u an arbitrary cell such that there exists an even time-step t so that $x_u^t = 0$ and $x_u^{t+2} = 1$. We claim that $x_u^{t+2k} = 1$ for every $k > 1$. We separate the proof of our claim in five cases, depending on the local functions of the cells adjacent to u . Let us call ℓ and r , respectively, the left and right neighbors of u .

In our figures, cell u is highlighted, and the local rules of the cells are specified by a “ b ” or an “ s ”, meaning “biased” and “stable” majorities, respectively. For instance, the case bbs is the case where u and ℓ have the biased majority, and r has the stable majority.

In our cases, we repeatedly use the following observations. First, if in a given time-step, two adjacent cells are in state 1 , then they will remain in state 1 in every future time-step independently of the local functions. Second, if, in a given time-step, two adjacent cells are in state 0 , and the local function of both cells have is stable majority, then both cells will remain in state 0 in every future time-step. Now we are ready to tackle the cases.

Case 1: bbb , bbs and sbb . Let us consider the case where u and ℓ have the biased majority rule (case sbb is symmetric to bbs). Since u is in state 1 at time-step $(t + 2)$, then at time-step $t + 3$, cell ℓ will be in state 1 , making u stay in state 1 in time-step $t + 4$. Inductively, we conclude that $x_u^{t+2k} = 1$ for every $k > 1$.

	ℓ	u	r
	b	b	b/s
t	.	0	.
$t + 1$	.	.	.
$t + 2$	.	1	.
$t + 3$	1	.	.
$t + 4$	.	1	.

Case 2: bsb . This case is similar to Case 1. If cell u reaches state 1 at time-step $(t + 2)$, then both neighbors will be in state 1 on time-step $t + 3$, which turns u into 1 in time-step $t + 4$. Inductively, we conclude that $x_u^{t+2k} = 1$ for every $k > 1$.

	ℓ	u	r
	b	s	b
t	.	0	.
$t + 1$	.	.	.
$t + 2$	.	1	.
$t + 3$	1	.	1
$t + 4$	.	1	.

Case 3: sbs . First notice that if r and ℓ are in state 0 on t , then the three cells will remain fixed in state 0 in every future time-step, which contradicts the choice of u and t . Then, at least one neighbor of u is in state 1 on t .

Without loss of generality, let us suppose that $x_\ell^t = 1$. This implies that $x_u^{t+1} = 1$, since u has the biased majority rule. Moreover, since $x_u^{t+2} = 1$, necessarily one neighbor of u is in state 1 at time-step $t + 1$. This implies that in time-step $t + 1$, cell u and one of its neighbors are simultaneously in state 1 . This implies that $x_u^{t+k} = 1$ for every $k > 0$.

	ℓ	u	r
	s	b	s
t	1	0	.
$t+1$	1	1	.
$t+2$	1	1	.

	ℓ	u	r
	s	b	s
t	1	0	.
$t+1$	.	1	1
$t+2$	.	1	1

Case 4: *sss*. If the three cells are ruled by the stable majority, and at least one neighbor is in state 0 at time-step t , then u will get fixed in 0, contradicting the choice of u and t . Then, we assume that $x_\ell^t = x_r^t = 1$. This implies that $x_u^{t+1} = 1$. Since $x_u^{t+2} = 1$, then at least one of the neighbors of u is in state 1 at time-step $(t+1)$ (in our figure below, we assume without loss of generality that $x_r^{t+1} = 1$). Then, we have two adjacent cells in state 1, making $x_u^{t+k} = 1$ for every $k > 0$.

	ℓ	u	r
	s	s	s
t	1	0	1
$t+1$	.	1	1
$t+2$	.	1	1

Case 5: *bss* and *ssb*. Let us study the case *bss* (the case *ssb* is symmetric). Observe that $x_r^t = 1$, because otherwise we will have two adjacent stable cells in 0 in the same time-step, fixing them in every future time-step, contradicting the choice of t and u . Since $x_u^{t+2} = 1$, at least one of the neighbors of u has to be in 1 in time-step $t+1$.

	ℓ	u	r
	b	s	s
t	.	0	1
$t+1$	.	.	1
$t+2$	.	1	.

	ℓ	u	r
	b	s	s
t	.	0	1
$t+1$	1	.	.
$t+2$	.	1	.

If $x_u^{t+1} = 1$, then we have that u and one of its neighbors are in state 1 in $t+1$, implying that u and that neighbor get fixed in state 1 on every future time-step.

	ℓ	u	r
	b	s	s
t	1	0	1
$t+1$	.	1	1
$t+2$	1	1	1

	ℓ	u	r
	b	s	s
t	1	0	1
$t+1$	1	1	.
$t+2$	1	1	.

If $x_u^{t+1} = 0$, then necessarily ℓ and r are in state 1 at time-step $(t+1)$. Since r has a stable-majority local rule and $x_u^t = 0$, then necessarily the right neighbor of r , namely w , is in state 1 at time-step t . This means that $x_r^{t+k} = x_w^{t+k} = 1$ for every $k > 0$. This implies that u and r are in state 1 at time-step $t+2$, meaning that $x_u^{t+1+k} = 1$ for every $k > 0$.

	ℓ	u	r	w
	b	s	s	
t	0	0	1	1
$t+1$	1	0	1	1
$t+2$	.	1	1	1
$t+3$	1	1	1	1

On all cases, we conclude that $x_u^{t+2k} = 1$ for every $k > 1$. Therefore, the cellular automata given by global function F^2 is 2-change. $\square$

Theorem 1. Prediction is in NL for every one-dimensional HMCA.

Proof. Let (F, x, t, i) be an input of the Prediction problem, where F is defined by some assignation of local rules $\mathcal{F}$. Let us denote G the global function F^2 . Observe that as a consequence of Proposition 1, there exists a nondeterministic logarithmic-space algorithm $\mathcal{A}$ solving the Prediction problem. Suppose first that t is even. In this case, we simply run $\mathcal{A}$ on input $(G, x, t/2, i)$. Observe that $G^{t/2}(x)_i = F^t(x)_i$. This implies that the output of $\mathcal{A}$ on input $(G, x, t/2, i)$ is the answer of Prediction on input (F, x, t, i) .

Suppose now that t is odd. In this case, we run $\mathcal{A}$ on inputs

$$(G, x, (t-1)/2, i-1), (G, x, (t-1)/2, i) \text{ and } (G, x, (t-1)/2, i+1).$$

From the outputs given by $\mathcal{A}$, we can deduce the values of $G^{(t-1)/2}(x)_{i-1}$, $G^{(t-1)/2}(x)_i$ and $G^{(t-1)/2}(x)_{i+1}$. These values correspond to the states of cells $i-1$, i and $i+1$ in $F^{t-1}(x)$. Finally, using the local function of i , we can compute $F_i^t(x)$ and decide the output. We deduce that Prediction is in NL. $\square$

4. Two-Dimensional Case

In this section, we give evidence that Prediction is harder to compute in the two-dimensional case than in the one-dimensional case. More precisely, we show that with the right combination of local functions and cell states, it is possible to simulate any monotone Boolean circuit in the two-dimensional grid. Hence, the prediction problem for the two-dimensional HMCA is P-Complete.

Theorem 2. *When restricted to two-dimensional HMCA with the von Neumann neighborhood, the Prediction problem is P-Complete.*

Proof. To show that PRE is P-complete we reduce the monotone circuit value problem to it. To do so, we simulate the input and the different parts of a Boolean circuit using a certain combination of biased and stable rules on a certain initial configuration.

In fact, according to [24], it is enough to construct a series of *gadgets*, namely *wires*, conjunction and disjunction gates, together with a cross-over and signal multipliers. These *gadgets* have to be of constant size and have to have two inputs and two outputs consistently fulfilling that the inputs are in the west and north side and the outputs are in the east and south side of the gadget. Furthermore, any rotation of these orientation constraints is valid. Additionally, inputs and outputs have to be placed in such a way that when two gadgets are put together, one of the outputs of one matches one of the inputs of the other. We follow this approach and build gadgets satisfying these conditions.

First, we explain how to simulate *wires*, i.e., gadgets that allow us to propagate signals through the grid. Wires are made by three rows (columns) of cells with the biased majority rule, two of them fully in the state $+1$, while the other one remains in the state -1 . To have a TRUE signal, turn the first two cells of one side of the wire to $+1$. An example of a wire is depicted in Figure 1.



Figure 1. Examples of the functioning of wires. On the (left), we have wire initialized as a FALSE signal. In the (middle), we have a wire initialized as a TRUE signal. On the (right), we have a wire with a TRUE signal after four time steps.

In all our figures, we use the colors black (■) and white (□) to represent states -1 and $+1$ of the cells with the stable majority rule, while cyan (■) and orange (■) represent states -1 and $+1$ for the cells with the biased majority rule. Notice that for a wire to work, it needs a background of stable cells in state -1 .

Next, we show how to simulate logic gates. Since we only simulate monotone circuits, we need to simulate conjunction and disjunction gates with fan-in 2 and fan-out 2. Our gadgets, which we call *logic gadgets*, are depicted in Figure 2.

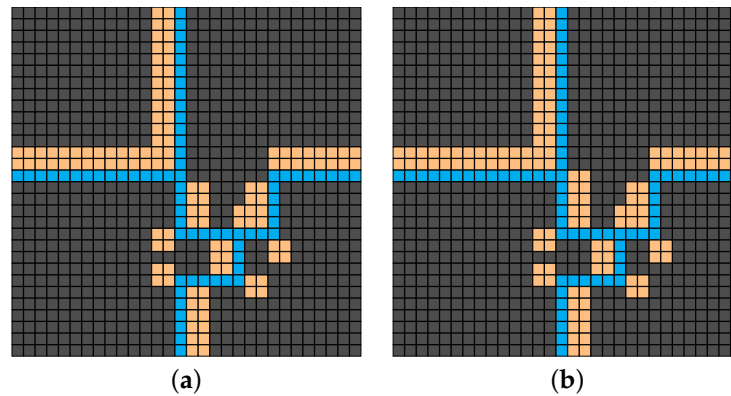


Figure 2. Logic gadgets that simulate conjunction gates (a) and disjunction gates (b).

Conjunction gates output TRUE only when both inputs are TRUE. In Figure 3, we show the evolution of a conjunction gadget in different combinations of input values. The disjunction gate construction is similar (see Figure 2b), but its outputs are TRUE with at least one TRUE input.

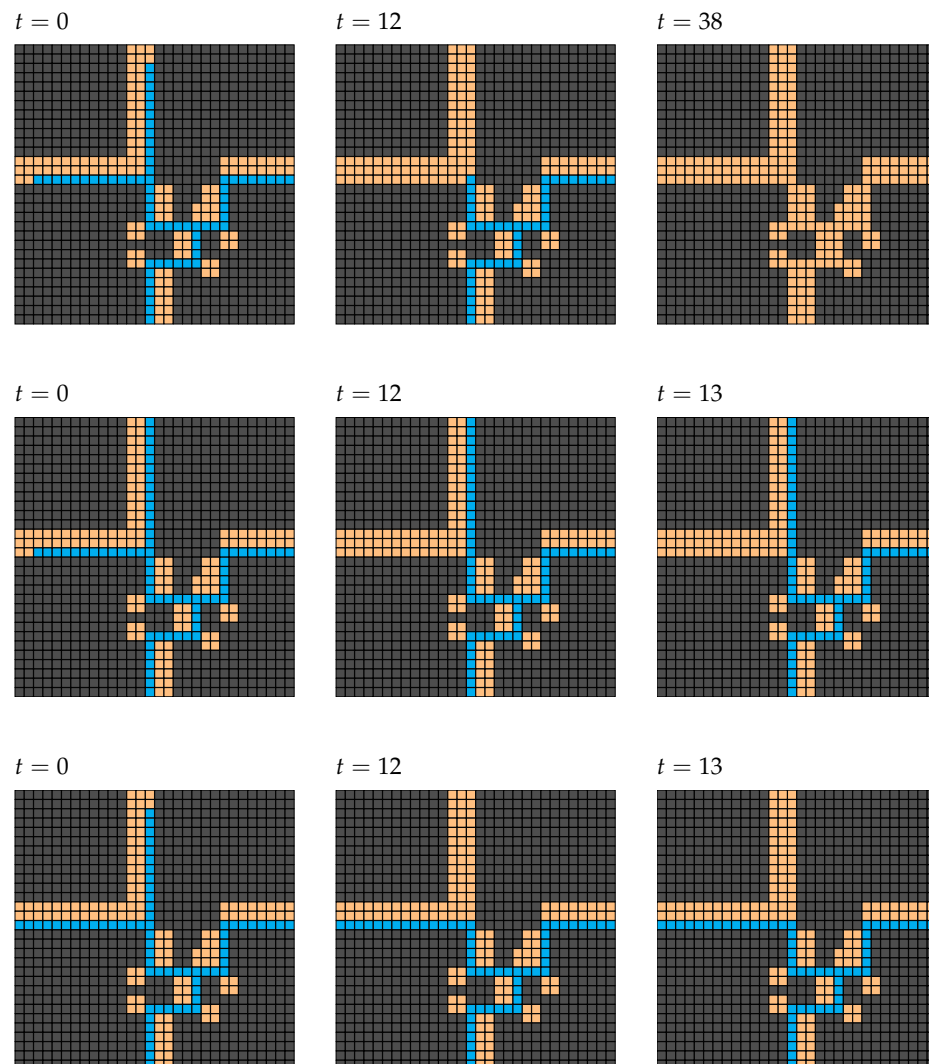


Figure 3. Behavior of the conjunction gadget when two inputs are TRUE (**top**), when the left input is TRUE (**middle**) and when only the top input is TRUE (**bottom**).

To simulate any circuit, we need to cross signals, but since the HMCA space is a planar graph, it is not evident that we can do it. In order to do so, we created a cross-over gadget (Figure 4) that can distinguish one signal from the other by using the oscillating behavior of the HMCA as “traffic lights”, letting the west signal go through its east output only and the north signal through the south output only. For an example, see Figure 5. In the case both signals are TRUE, the cross-over gadget works as a conjunction gate. No prior signal coordination is needed.

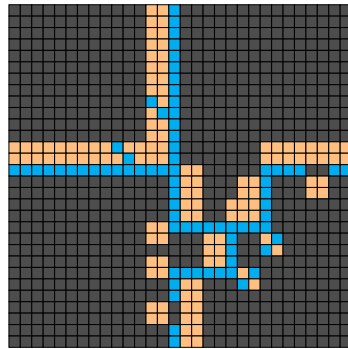


Figure 4. Cross-over gadget.

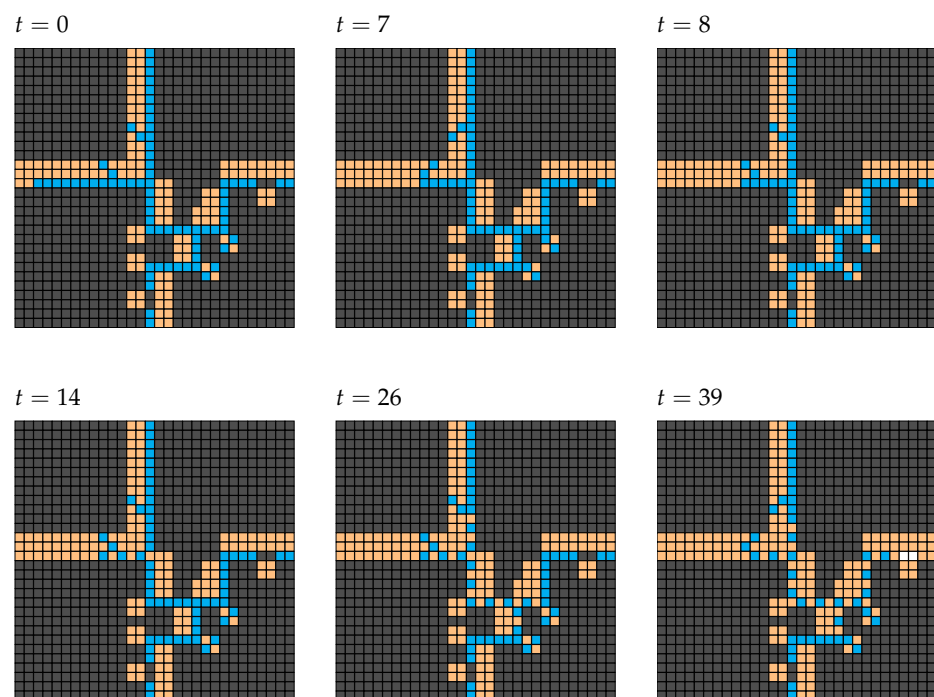


Figure 5. Cross-over gadget behavior with one TRUE input.

Finally, as the space is an undirected graph, but circuits are defined over directed graph, we need diodes making the signals going only in directions west-east and north-south (Figure 6). For a better understanding of the diode behavior, see Figures 7 and 8.

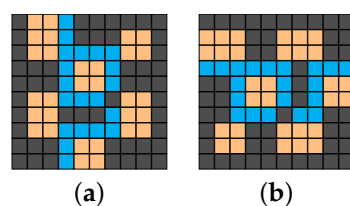


Figure 6. Diode gadgets. The vertical diode allows signal to pass only from top to bottom, while the horizontal diode allows the signal to go only from left to right. (a) Vertical diode; (b) horizontal diode.

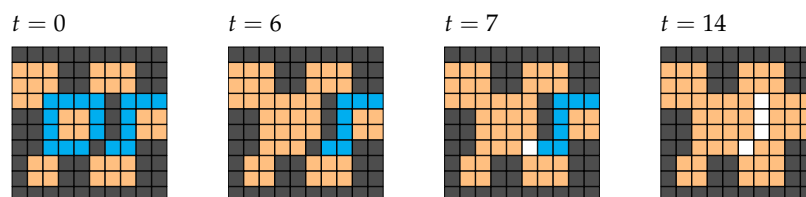


Figure 7. Horizontal diode behavior when a signal comes from the west.

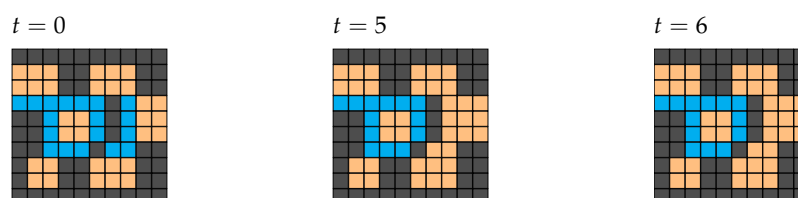


Figure 8. Horizontal diode behavior when a signal comes from the east.

These diodes have to be appended to every gadget's outputs. In Figure 9, we depict the complete conjunction, disjunction and crossing gadgets, including the output diodes. Observe that, in at most 39 time-steps, the gadgets will produce the output values. In fact, the latter upper-bound is uniform for all the gadgets. $\square$

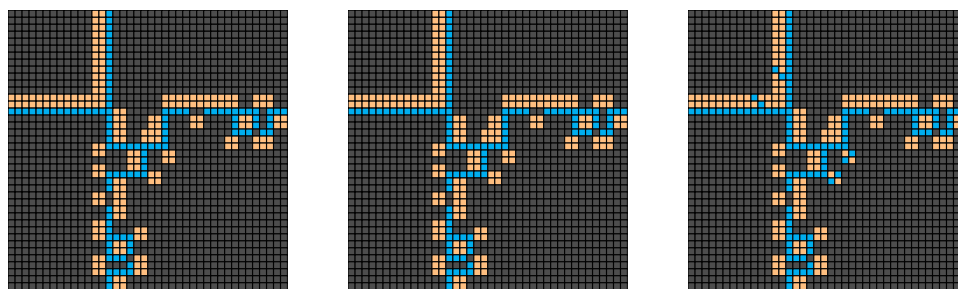


Figure 9. Disjunction (left), Conjunction (middle) and crossing gadgets (right) including their output diodes.

5. Conclusions

In this paper, we study the computational complexity of the HMCA in one and two-dimensional grids. For the one-dimensional HMCA, the prediction problem turns out to be in the class **NL** while in two dimensions it is **P-Complete**.

On the upper-bound side, we showed that the dynamic given by two iterations of the HMCA induce a 2-change CA. Obviously, this property is preserved when a homogeneous (biased or stable) one-dimensional majority cellular automata is considered. We believe that such a property can be extended to more dimensions. Nevertheless, it is unlikely that such techniques can be extended to show non-trivial complexity upper-bounds for the Prediction problem. Indeed, there exist two-dimensional bounded-change (even 1-change) cellular automata that are **P-Complete**(see, for instance, [21]).

Our proof of **P-completeness** shows that it is possible to simulate cable crossings in a two-dimensional grid under monotone dynamics. Whether this property can be preserved

using a uniform majority automata is a difficult open question that remains open. In addition, a related question that could be interesting to explore is whether we can simulate circuits using another type of tie-breaking rules, such as the unstable majority, in which, in the tie case, a cell will always change its state. Finally, another interesting open question is the one related to the simulation capabilities of HMCA equipped with other neighborhood topologies, such as the Moore neighborhood or the Toom neighborhood.

Author Contributions: All authors contributed equally to the conceptualization, methodology, investigation, writing, review and editing of the article. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Centro de Modelamiento Matemático (CMM), FB210005 BASAL funds for centers of excellence from ANID-Chile. FONDECYT 11190482 (P.M.), FONDECYT 1200006 (E.G., P.M.), ANID FONDECYT Postdoctorado 3220205 (M.R-W).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Schelling, T.C. *Micromotives and Macrobehavior*; WW Norton & Company: New York, NY, USA, 2006.
- Castellano, C.; Fortunato, S.; Loreto, V. Statistical physics of social dynamics. *Rev. Mod. Phys.* **2009**, *81*, 591. [\[CrossRef\]](#)
- Hegselmann, R. Modeling social dynamics by cellular automata. *Comput. Model. Soc. Process.* **1998**, 37–64.
- Ancona, C.; Iudice, F.L.; Garofalo, F.; De Lellis, P. A model-based opinion dynamics approach to tackle vaccine hesitancy. *Sci. Rep.* **2022**, *12*, 11835. [\[CrossRef\]](#) [\[PubMed\]](#)
- Agur, Z.; Daniel, Y.; Ginosar, Y. The universal properties of stem cells as pinpointed by a simple discrete model. *J. Math. Biol.* **2002**, *44*, 79–86. [\[PubMed\]](#)
- Yassine, D. *Network Decontamination with Temporal Immunity*; University of Ottawa (Canada): Ottawa, ON, USA, 2012.
- Goles, E.; Olivos, J. Periodic behaviour of generalized threshold functions. *Discret. Math.* **1980**, *30*, 187–189. [\[CrossRef\]](#)
- Greenlaw, R.; Hoover, H.; Ruzzo, W. *Limits to Parallel Computation: P-Completeness Theory*; Oxford University Press, Inc.: New York, NY, USA, 1995.
- Moore, C. *Majority-Vote Cellular Automata, Ising Dynamics, and P-Completeness*; Working papers; Santa Fe Institute: Santa Fe, NM, USA, 1996.
- Epstein, R.L.; Carnielli, W.A. *Computability Computable Functions, Logic, and the Foundations of Mathematics*; Wadsworth Publ. Co.: Belmont, CA, USA, 2000.
- Wegener, I. *The Complexity of Boolean Functions*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1987.
- Vollmer, H. *Introduction to Circuit Complexity: A Uniform Approach*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1999.
- Goldschlager, L.M. The Monotone and Planar Circuit Value Problems Are Log Space Complete for P. *SIGACT News* **1977**, *9*, 25–29. [\[CrossRef\]](#)
- Goles, E.; Ollinger, N.; Theyssier, G. Introducing Freezing Cellular Automata. In Proceedings of the Cellular Automata and Discrete Complex Systems, 21st International Workshop (AUTOMATA 2015), Turku, Finland, 8–10 June 2015; TUCS Lecture Notes; Volume 24, pp. 65–73.
- Theyssier, G.; Ollinger, N. Freezing, Bounded-Change and Convergent Cellular Automata. *Discret. Math. Theor. Comput. Sci.* **2022**, *24*. [\[CrossRef\]](#)
- Karafyllidis, I.; Thanailakis, A. A model for predicting forest fire spreading using cellular automata. *Ecol. Model.* **1997**, *99*, 87–97. [\[CrossRef\]](#)
- Fuentes, M.; Kuperman, M. Cellular automata and epidemiological models with spatial dependence. *Phys. A Stat. Mech. Appl.* **1999**, *267*, 471–486. [\[CrossRef\]](#)
- Chalupa, J.; Leath, P.L.; Reich, G.R. Bootstrap percolation on a Bethe lattice. *J. Phys. C Solid State Phys.* **1979**, *12*, L31. [\[CrossRef\]](#)
- Goles, E.; Montealegre-Barba, P.; Todinca, I. The complexity of the bootstrapping percolation and other problems. *Theor. Comput. Sci.* **2013**, *504*, 73–82. [\[CrossRef\]](#)
- Goles, E.; Maldonado, D.; Montealegre, P.; Ollinger, N. On the computational complexity of the freezing non-strict majority automata. In Proceedings of the International Workshop on Cellular Automata and Discrete Complex Systems, Milan, Italy, 7–9 June 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 109–119.
- Goles, E.; Maldonado, D.; Montealegre, P.; Ollinger, N. On the complexity of the stability problem of binary freezing totalistic cellular automata. *Inf. Comput.* **2020**, *274*, 104535. [\[CrossRef\]](#)
- Goles, E.; Montealegre, P. The complexity of the majority rule on planar graphs. *Adv. Appl. Math.* **2015**, *64*, 111–123. [\[CrossRef\]](#)

-
23. Goles, E.; Montealegre, P. Computational complexity of threshold automata networks under different updating schemes. *Theor. Comput. Sci.* **2014**, *559*, 3–19. [[CrossRef](#)]
 24. Goles, E.; Montealegre, P.; Perrot, K.; Theyssier, G. On the complexity of two-dimensional signed majority cellular automata. *J. Comput. Syst. Sci.* **2018**, *91*, 1–32. [[CrossRef](#)]
 25. Goles, E.; Montealegre, P. The complexity of the asynchronous prediction of the majority automata. *Inf. Comput.* **2020**, *274*, 104537. [[CrossRef](#)]