Network Decontamination with Temporal Immunity by Cellular Automata*

Yassine Daadaa, Paola Flocchini, and Nejib Zaguia

SITE, University of Ottawa, Ottawa, ON K1N 6N5, Canada
{ydaadaa,flocchin,zaguia}@site.uottawa.ca

Abstract. Network decontamination (or disinfection) is a widely studied problem in distributed computing. Network sites are assumed to be contaminated (e.g., by a virus) and a team of agents is deployed to decontaminate the whole network. In the vast literature a variety of assumptions are made on the power of the agents, which can typically communicate, exchange information, remember the past, etc.

In this paper we consider the problem in a much weaker setting; in fact we wish to describe the global disinfection process by a set of cellular automata local rules without the use of active agents. We consider the grid, which is naturally described by a 2-dimensional cellular automata, and we devise disinfection rules both in the common situation where after being disinfected a cell is prone to re-contamination by contact, and in a new setting where disinfection leaves the cells immune to recontamination for a certain amount of time (*temporal immunity*). We also distinguish between Von Neuman and Moore neighborhood, showing that, not surprisingly, a bigger neighborhood allows for a more efficient disinfection.

1 Introduction

1.1 The Problem

Consider a network where nodes are processing performing some computations, but where some of them might be contaminated (e.g., by a virus). Once contaminated, a node might behave incorrectly; furthermore it could cause its neighboring cells to become contaminated as well, thus propagating faulty computations. A pressing concern for fault tolerance and security is obviously to devise strategies to correct the faulty behavior at network sites and to neutralize the propagation of faults. To this end, nodes are endowed with antiviral software that can be activated to perform local disinfection leaving the node clean and possibly immune to further contamination for a certain amount of time. When a node is disinfected however, there is no guarantee it will not be contaminated again; when the immunity time expires in fact, it stays unprotected and becomes contaminated if any neighbour is. The problem is that nodes cannot detect the presence of a virus in themselves and in their neighborhood, and they cannot

* This work was partially supported by NSERC.

S. Bandini et al. (Eds.): ACRI 2010, LNCS 6350, pp. 287–299, 2010.

control the spread of contamination. They can however detect whether the antiviral process is active or has been activated in their neighborhood; moreover, they want to sparingly activate the anti-virus because, when active, it interrupts any other local computation (e.g., the trivial solution of simultaneously running the anti-virus in all nodes would be unfeasible as it would completely disrupt the computation). The goal is to devise a strategy that makes all nodes simultaneously clean, regardless of the initial contamination pattern, minimizing the number of nodes simultaneously running the anti-virus at any time.

Interestingly, this process can be modelled using a cellular automaton where different rules correspond to different strategies. At any point in time a node can be in one of the following three states: {*disinfecting*, *disinfected*, *unprotected*}. A *disinfecting* node is running the antiviral software that will leave the node *disinfected* (clean and protected for some time t which depends on the the antiviral properties); a *disinfected* node has run the anti-viral software; an *unprotected* node has never run the antiviral software. Both *disinfected* and *unprotected* nodes are processing and running the underlying computation. In particular, we will consider two cases regarding the local disinfecting procedure, which could leave the node: 1) disinfected but with no immunity (*basic disinfection*), 2) disinfected and immunized for a certain amount of time (*temporal disinfection*). In this context, the goal is to start the disinfection process by setting the state of some cells to *disinfecting* and letting *unprotected* cells activate disinfection on the basis of local rules so that, even in the worst case when all cells are initially contaminated, at the end of the process they are all simultaneously clean. Note that, when all nodes are simultaneously clean they will stay clean forever as we assume contamination is already present before the disinfection begins, and cannot "enter" the system. We are interested in minimizing the number of simultaneous disinfecting sites for a given immunity time or, conversely, minimizing the immunity time for a given number of simultaneous disinfecting sites. Moreover, we are interested in monotone disinfection strategies, where the anti-virus is run only once at each node, that is, once a node is *disinfected*, we must guarantee that, regardless of its protection level, it will stay clean forever.

We consider a common network topology: the grid, which naturally corresponds to a 2-dimensional cellular automaton $n \times n$. We first look at *basic* disinfection and we describe a simple strategy that employs the minimum possible number of disinfecting sites $k = n$ per time unit. We then turn to *temporal* disinfection and we show that, with Von Neumann neighborhood, disinfection can be achieved with $k = 1$ disinfecting site per time unit *if and only if* the immunity time is at least $4(n-1)-1$. We also extend the technique for $k = 2$ and $k = 4$. We then consider Moore neighborhoods. Also in this case we describe optimal disinfection rules for the case $k = 1$ and immunity time at least $2n-1$. The same set of rules allows disinfection with $k > 1$ simultaneously disinfecting sites and immunity time at least $\lfloor \frac{2(2n-1)}{k} \rfloor$ (see Table 1).

Due to lack of space some proofs are sketched and some omitted.

Table 1. Summary of results

Neighborhood	Number of disinfecting sites	Immunity time
Von Neumann	$k = 1, 2, 4$	$\geq \frac{4}{k}(n-1) - 1$
Moore	k	$\geq \lfloor \frac{2(2n-1)}{k} \rfloor$

1.2 Related Work

A large body of work exists on the network decontamination problem using a team of mobile agents, which appropriately move from node to neighbouring node to activate the disinfection (e.g., see [1,2,4,8,11]). In such contexts, the goal is typically to devise a strategy for the agents to collaboratively decontaminate the whole network using the smallest possible team in such a way that, once cleaned, a node does not get recontaminated. In the existing work, agents are regulated by a variety of assumptions. For example, agents usually can communicate with each other, sometimes in a face-to-face manner (i.e., when they meet at a node) or by writing messages for each other on special spaces located at the nodes; agents have local memory where they store information about their past actions or about a map of the network; agents are usually identified by distinct ids, they are sometimes coordinated by a special agent that acts as a leader, and they might be able to clone themselves. With the exception of [6,11], no immunity has ever been considered. In [11] immunity is defined differently and is related to the number of contaminated neighbours; in [6] it is defined as in this paper but it is assumed in a different setting and exclusively for the case of trees. Determining the minimum size of a decontamination team, even without any immunity, is known to be NP-hard ([12]); optimal strategies have been devised for special topologies (typical topologies are the grid, the torus, the hypercube, the chordal ring), and studies on the various models have been done to describe the computational relationship between them (e.g., see [1,2,4,8,10]). In [3] decontamination by mobile agents has been linked to cellular automata, but in a very different way: contamination was regulated by cellular automata rules (either unanimity or majority rules), decontamination (called in that context *external decontamination*) was instead performed by mobile agents.

2 Definitions

A 2-dimensional cellular automata (CA) can be described by a quadruple $C = \langle \mathbb{Z}^2, \{0, 1, \bullet\}, N, f \rangle$ where: $\mathbb{Z}^2$ represents the set of *cells* (also called *sites* or *nodes*); $\{0, 1, \bullet\}$ is the set of *states* of the cells; N is the *neighbourhood* of a cell, and $f : \{0, 1, \bullet\}^{|N|} \rightarrow \{0, 1, \bullet\}$ is the *local transition rule* (or simply *local rule*) of the automaton. In the following we will be considering both *von Neumann* and *Moore* neighborhoods at distance one. Given a cell (i, j), the *von Neumann* neighborhood consists of the cell itself, plus the four cells at distance one in the Manahattan norm. The *Moore* neighborhood, besides considering the cells of the von Neumann neighborhood, also includes the four neighboring "diagonal cells".

Given an initial configuration, C^0, that is a mapping $C^0 : \mathbb{Z}^2 \rightarrow \{0, 1, \bullet\}$, cell states are synchronously updated at each time step by the local transition rule applied to their neighbourhoods. A configuration is the resulting map $C^t : \mathbb{Z}^2 \rightarrow \{0, 1, \bullet\}$ at any time t. In the following we will denote a transition rule by indicating the neighborhing states in clock-wise order starting from the left neighour, for example, in the case of Von Neumann neighborhoods we will use the following notation: $(x^t_{i,j}, x^t_{i-1,j}, x^t_{i,j+1}, x^t_{i+1,j}, x^t_{i.j-1}) \rightarrow x^{t+1}_{i,j}$, and we will denote an arbitrary state by an asterisk $(*)$. A *finite* 2-dimensional Boolean cellular automaton has a finite number of non-zero states in an infinite quiescent background. That is, $C^t(z) = 0$ for all but finitely many $z \in \mathbb{Z}^d$. In this case, let $n \times n$ be the size of the finite lattice initially containing non-zero states; moreover, let us indicate as (i, j) a cell in column i, row j with respect to the finite lattice indicating the left-bottom corner with $(0, 0)$. *Border* cells are cells $(i, n-1)(0, j), (i, 0), (n-1, j)$ with $0 < i, j < n-1$, the four *corner* cells are $(0, 0), (0, n-1), (n-1, 0), (n-1, n-1)$, all other cells are called *internal.* A *Circular* cellular automata can be thought of a 2-dimensional grid where the last node of a row (resp. column) is connected to the first.

3 Basic Disinfection

In this section we consider basic disinfection in finite 2-dimensional CAs, where the disinfecting process does not provide any type of immunity and a cell could be contaminated as soon as one of its neighbours is. In this case any disinfection has to forbid a *disinfected* cell to ever be in contact with a *unprotected* one, which is potentially contaminated.

Notice that with basic disinfection, a single disinfecting site per time unit would obviously not be sufficient to perform decontamination because immediately after becoming *disinfected* a site would inevitably be exposed to a *unprotected* site as at most one of its neighbors could become *disinfecting.* The question is what is the minimum number of disinfecting sites which could guarantee disinfection without recontamination. We prove in the following that n disinfecting sites are necessary and sufficient.

Theorem 1. *Optimal basic disinfection can be achieved in a finite CA with Von Neumann neighborhood using n simultaneous disinfecting sites.*

Proof (Sketch). The proof that n is sufficient is constructive. The idea is to place the n initial disinfecting sites in the cells of the first column of the CA and to have the cells act according to the following very simple rules (see Table 2): regardless of the neighbourhing cells' state, a *disinfecting* cell becomes *disinfected*, an *unprotected* cell becomes *disinfecting* at time $t + 1$ if its left neighbour is *disinfecting* at time t. The effect of the local rules is the sequential disinfection of columns. In fact, it is easy to see by induction on the time steps that the CA is fully disinfected in n time steps with no *disinfected* site ever in contact with an *unprotected* one.

Table 2. Basic disinfection in Finite CAs with Von Neumann neighborhood. All missing combinations of states do leave the cell unchanged.

Basic-Finite

`Initial disinfecting sites:` $(0, j)$, $0 \leq j \leq n-1$

Configuration	Next State
$\{\bullet, *, *, *, *\}$	0
$\{1, \bullet, *, *, *\}$	$\bullet$

To prove optimality, consider a time t during the disinfection process when there are h disinfected cells. To avoid recontamination at time t, these cells must be surrounded by disinfecting cells (or by quiescent cells outside the border of the CA); we will call such cells *protective cells.* It is easy to see that, if the disinfected cells form separate continuous blocks, the amount of necessary protective cells is never smaller than if the disinfected cells belonged to a single block. Moreover, it has been shown in [9] that for a block of size h at least $\min\{n, \lfloor\frac{1+\sqrt{1+8h}}{2}\rfloor\}$ protective cells are mecessary. Since $\lfloor\frac{1+\sqrt{1+8h}}{2}\rfloor < n$ implies $h < \lfloor\frac{n(n-1)}{2}\rfloor$, we have that less than n simultaneous disinfecting cells can protect less than half the cells of the CA, which then cannot be fully disinfected.

4 Temporal Disinfection

The case of temporal disinfection is quite different and more interesting. We must design a set of local rules and choose the location of the initial *disinfecting* sites in such a way that during the evolution of the CA, a *disinfected* node never comes into contact with an *unprotected* one after its immunity time has expired We distinguish here the two types of neighborhood: Von Neumann and Moore, noticing that the amount of influence from neighbourhing cells highly impacts the efficiency solutions.

4.1 Von Neumann Neighbourhood

We first show that, if we impose the use of a single disinfecting cell per time unit, temporal disinfection can be achieved if and only if the immunity time is $\geq 4(n-1)-1$. In the following, and in the rest of the paper, when we say that disinfection "propagates", we indicate that a *disinfecting* cell becomes *disinfected* and one or more of its neighbours become *disinfecting*, thus simulating the propagation of disinfection from cell to neighbouring cell(s).

Theorem 2. *With a single disinfecting site per time unit, temporal disinfection is possible if and only if the immunity time is* $\geq 4(n-1)-1$.

Proof (Sketch). Disinfection is achieved by placing the initial disinfection site at a corner and by applying the set of rules indicated in Table 3 (a) (Single-Temporal-vn). It is easy to see that the effect of the local rules is the spiral

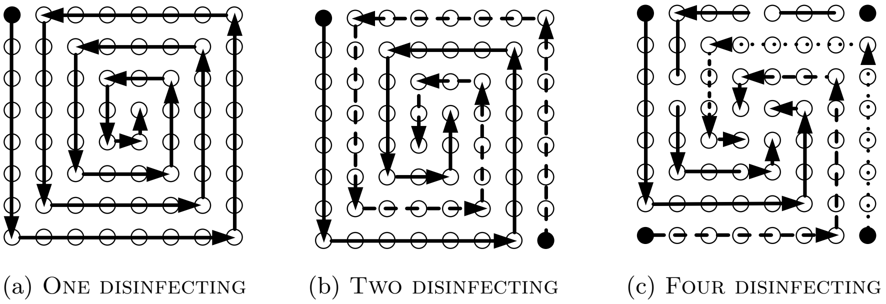


Fig. 1. Propagation of disinfection with temporal immunity and Von Neumann neghborhood

propagation of the disinfecting site (see Figure 1) where a *disinfected* cell is never in contact with an *unprotected* one for more than $4(n-1)-1$ time units, resulting in a situation where all nodes are simultaneously clean after $n \times n$ time units.

We now show that there exist no initial placement for which a correct set of rules exists when the immunity time smaller than $4(n-1)-1$, where a correct set of rules is such that it maintains a single disinfecting site per time unit, and eventually disinfects the whole network never leaving a disinfected site in contact with an *unprotected* one for more than $4(n-1)-1$ time units after it has been cleaned.

We have three possible placements of the initial disinfecting site: corner border, and internal. Let us first consider the case when the initial disinfecting site is internal, in this case we will show that, regardless of the immunity time, there exist no set of rules that allows to maintain a single disinfecting site. Let us call *forbidden rule* a rule that cannot be present in a correct set of rules. First of all notice that $r_0 = \{\bullet, *, *, *, *\} \rightarrow 0$ is necessarily part of the set of rules to insure the presence of a single disinfecting site. Moreover, to allow the spread of disinfection to a single new site, one (and only one) of the following four rules must be present ($r_1 = \{1, \bullet, 1, 1, 1\} \rightarrow \bullet$, $r_2 = \{1, 1, \bullet, 1, 1\} \rightarrow \bullet$, $r_3 = \{1, 1, 1, \bullet, 1\} \rightarrow \bullet$, $r_4 = \{1, 1, 1, 1, \bullet\} \rightarrow \bullet$). Without loss of generality, let r_1 be present and let r_2, r_3, r_4 be forbidden. The combination of r_0 and r_1 makes the disinfecting site propagate to the right until reaching the border. At this point, for insuring the propagation of the disinfecting site, one (and only one) of the following three rules must be added to the set: $r_5 = \{0, 0, 1, \bullet, 1\} \rightarrow \bullet$, $r_6 = \{1, 1, 1, 0, \bullet\} \rightarrow \bullet$). $r_7 = \{1, 1, \bullet, 0, 1\} \rightarrow \bullet$). We can show that r_5 is forbidden (the disinfecting site would propagate back to the original site where however, because of the fact that r_2 ,r_3, and r_4 are forbidden the only possible movement would be to the right again giving rise to an oscillation). Let, w.l.g, r_6 be present and r_7 be forbidden. Because of r_6 being present, rule $r_8 = \{1, 1, 1, \bullet, 0\} \rightarrow \bullet$ must be forbidden, otherwise more than one site would be in disinfecting state. The combination of r_0 and r_6 makes the disinfecting site propagating up until reaching the down neighbour of the corner, at this point, since r_3 is forbidden, we need to add a rule to ensure propagation, and the only possible one is $r_9 = \{1, 1, 0, 0, \bullet\} \rightarrow \bullet$ and

Table 3. Rule Tables for 1/2/3 disinfecting sites in a Finite CA with temporal immunity and Von Neumann neighborhood. All missing combinations of states for the neighbourhood of cell (i, j) do leave $x_{i,j}$ unchanged.

Single-Temporal-vn
Init: **a corner**

Configuration	Next State
$\{\bullet, *, *, *, *\}$	0
$\{1, \bullet, 1, 1, 0\}$	$\bullet$
$\{1, \bullet, 1, 0, 0\}$	$\bullet$
$\{1, 0, \bullet, 1, 1\}$	$\bullet$
$\{1, 0, \bullet, 1, 0\}$	$\bullet$
$\{1, 0, \bullet, 0, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 1\}$	$\bullet$
$\{1, 1, 0, \bullet, 1\}$	$\bullet$
$\{1, 1, 1, 0, \bullet\}$	$\bullet$
$\{1, 1, 0, 0, \bullet\}$	$\bullet$
$\{1, 0, 0, 0, \bullet\}$	$\bullet$

a)

Two-Temporal-vn
Init: **two opposite corners**

Configuration	Next State
$\{\bullet, *, *, *, *\}$	0
$\{1, \bullet, 1, 1, 0\}$	$\bullet$
$\{1, \bullet, 1, 0, 0\}$	$\bullet$
$\{1, 0, \bullet, 1, 1\}$	$\bullet$
$\{1, 0, \bullet, 1, 0\}$	$\bullet$
$\{1, 0, \bullet, 1, 1\}$	$\bullet$
$\{1, 1, 0, \bullet, 1\}$	$\bullet$
$\{1, 0, 0, \bullet, 1\}$	$\bullet$
$\{1, 1, 1, 0, \bullet\}$	$\bullet$
$\{1, 1, 0, 0, \bullet\}$	$\bullet$
$\{1, \bullet, 0, 0, \bullet\}$	$\bullet$
$\{1, 0, \bullet, \bullet, 0\}$	$\bullet$
$\{1, 0, \bullet, 0, \bullet\}$	$\bullet$

b)

Four-Temporal-vn
Init: **all corners**

Configuration	Next State
$\{\bullet, *, *, *, *\}$	0
$\{1, \bullet, 1, 1, 0\}$	$\bullet$
$\{1, \bullet, 1, 0, 0\}$	$\bullet$
$\{1, \bullet, 1, 1, 0\}$	$\bullet$
$\{1, 0, \bullet, 1, 1\}$	$\bullet$
$\{1, 0, \bullet, 1, 0\}$	$\bullet$
$\{1, 0, \bullet, 1, 1\}$	$\bullet$
$\{1, 1, 0, \bullet, 1\}$	$\bullet$
$\{1, 0, 0, \bullet, 1\}$	$\bullet$
$\{1, 1, 1, 0, \bullet\}$	$\bullet$
$\{1, 1, 0, 0, \bullet\}$	$\bullet$
$\{1, \bullet, 1, \bullet, 0\}$	$\bullet$
$\{1, \bullet, 0, \bullet, 1\}$	$\bullet$
$\{1, 0, \bullet, 1, \bullet\}$	$\bullet$
$\{1, 1, \bullet, 0, \bullet\}$	$\bullet$
$\{1, \bullet, \bullet, \bullet, \bullet\}$	$\bullet$

c)

the disinfection will reach the corner. At this point, for insuring the propagation of the disinfecting site, one (and only one), of the following two rules must be added to the set: ($r_{10} = \{0, 1, \bullet, 0, 0\} \rightarrow \bullet$), and $r_{11} = \{1, 1, 0, \bullet, 1\} \rightarrow \bullet$). Rule r_{10} is forbidden because the disinfecting site would propagate back to the the first disinfected site on the border and only an oscillation will be permitted (all the other rules are forbidden). Following a similar reasoning, one can see that the disinfecting site is forced to propagate left until reaching the corner with the necessary inclusion of rule $r_{12} = \{1, 0, 0, \bullet, 1\} \rightarrow \bullet$, then down following another necessary rule $r_{14} = \{1, 0, \bullet, 1, 1\} \rightarrow \bullet$. At the next step, however two sites ($(0, n-3)$ and $(1, n-2)$) will become simultaneously disinfecting (due to rules r_1 and r_{14} respectively), which contradicts our hypothesis. It follows that starting from an internal cells we cannot decontaminate all sites with a single disinfecting site per time unit.

If the initial disinfecting site is a border site or a corner, one can show, using similar arguments, that all combinations of possible rules either lead to an impossibility, or to a disinfection requiring an immunity higher than $4(n-1)-1$.

Interestingly, the strategy can be generalized when we allow two or four simultaneous disinfecting sites. In the first case we achieve disinfection initially placing the two disinfecting sites in two opposite corners and having the rules of Table 3 (*b*) describe the local behavior of the cells; in the second case the four disinfecting sites are initially placed in the four corners and the rules are described in Table 3 (c). The global behavior of the automata corresponds to the propagation of the disinfection in spirals (see Figure 1 b) and c)), and it is easy to see that the immunity time is at least $2(n-1)-1$ for the case of two disinfecting sites, and $(n-1)-1$ for four. In fact, we can then conclude:

Theorem 3. *With $k = 1, 2, 4$ disinfecting site per time unit, temporal disinfection is possible if and only if the immunity time is $\geq (4-k)(n-1)-1$.*

4.2 Moore Neighbourhood

The bounds are different if we increase the neighborhood to include diagonal neighbours. In fact, we can show that with a single disinfecting site optimality is reached when the immunity time is at least $2n-1$. We now describe a general set of rules which achieves optimality in the particular case of a single agent placed on a corner, but which works correctly also with more disinfecting sites.

Table 4. Rule Table for Finite CA with temporal immunity and Moore neighborhood. All missing combinations of states for the neighbourhood of cell (i,j) do leave $x_{i,j}$ unchanged.

Temporal-Moore

Configuration	Next State
$\{\bullet, *, *, *, *, *, *, *, *\}$	0
$\{1, \bullet, 0, 1, 1, 1, 1, 1, 0\}$	$\bullet$
$\{1, \bullet, 0, 0, 0, 1, 1, 0, 0\}$	$\bullet$
$\{1, \bullet, 0, 1, 1, 1, 0, 0, 0\}$	$\bullet$
$\{1, \bullet, 0, 0, 0, 1, 1, 1, 0\}$	$\bullet$
$\{1, \bullet, 0, 1, 1, 1, 1, 1, \bullet\}$	$\bullet$
$\{1, \bullet, 0, 1, 0, 0, 0, 1, 0\}$	$\bullet$
$\{1, \bullet, 0, 1, 0, 0, 0, 0, 0\}$	$\bullet$
$\{1, \bullet, 0, 0, 0, 0, 0, 1, 0\}$	$\bullet$
$\{1, \bullet, \bullet, 1, 1, 1, 1, 1, 0\}$	$\bullet$
$\{1, \bullet, 0, 1, 0, 0, 0, 1, \bullet\}$	$\bullet$
$\{1, \bullet, \bullet, 1, 0, 0, 0, 1, 0\}$	$\bullet$

Configuration	Next State
$\{1, \bullet, \bullet, 0, 0, 0, 0, 0, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 0, 0, 0, 0, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 1, 1, 1, 1, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 1, 1, 0, 0, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 0, 0, 0, 1, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 1, 1, 1, \bullet, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 0, 0, 0, \bullet, 0\}$	$\bullet$
$\{1, 0, 0, 1, 0, 0, 0, \bullet, 0\}$	$\bullet$
$\{1, 0, 0, 0, 0, 0, 0, \bullet, 0\}$	$\bullet$
$\{1, 0, 0, 0, 0, 1, 1, \bullet, 0\}$	$\bullet$
$\{1, 0, 0, 1, 1, 1, 1, \bullet, 0\}$	$\bullet$

Any number of *disinfecting* sites (greater than 1) are initially placed on the first column at distance greater than 1 from each other (i.e, no two consecutive cells are initially disinfecting). Careful inspection of the rules in Table 4 shows that disinfection "propagates" vertically to unprotected cells (an *unprotected* cell becomes *disinfecting* when its down/up neighbour, or both, are *disinfecting*) until they reach either another disinfecting cell or a clean one, in which case they "propagate" to the next column (an *unprotected* cell becomes *disinfecting* when its left neighbour is *disinfecting* and at least one of its left diagonal left neighbours are *disinfected*). Figure 2 shows the effect of the local rules on some partial configurations. Figure 3 describes instead the global propagation path of the disinfection. In the following we prove the correctness of this set of rules.

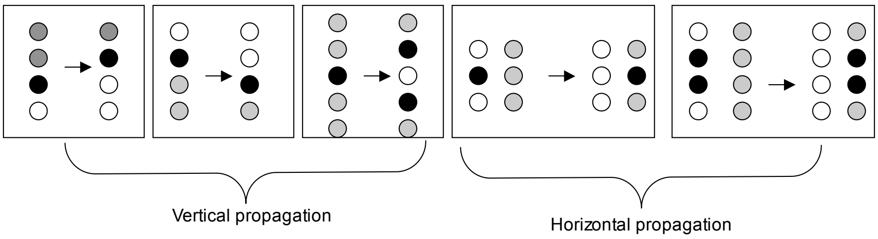
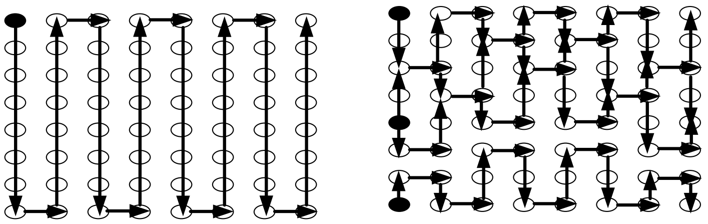


Fig. 2. Examples of vertical and horizontal propagation. Black cells are *disinfecting*, grey cells are *unprotected*, white cells are *disinfected*.



(a) One disinfecting (b) Several disinfecting

Fig. 3. Propagation of disinfection with temporal immunity and Moore neghborhood

Theorem 4. *Let d_{max} be the maximum distance between two consecutive* disinfecting *cells or between a* disinfecting *cell and a corner at time 0. In the initial number of disinfecting sites is at least 2, the set of rules* Temporal-Moore *achieves disinfection with immunity time $t = d_{max}$.*

Proof (Sketch). To prove the theorem, we need to prove that disinfection is achieved monotonically. That is, that once *disinfected*, every cell stays clean until the end of the process, when all cells are clean. Let us call *entry points* of disinfection in a column, the cells in such a column that become disinfecting due to a left disinfecting neighbour (horizontal propagation). We now prove by induction on the number of columns that, for each column i there is a time t when: (i) all cells in column i are either *disinfected* or *disinfecting*; (ii) all cells in column $i-1$ (for $i > 0$) are clean; (iii) by time $t + d_{max}$ all right neighbours of a *disinfected* cell of column i are either *disinfected* or *disinfecting*; (iv) the distance between any two consecutive entry points in column $i+1$ (for $i < n-1$) is smaller than d_{max}.

1. Base - column 0: According to the set of rules Temporal-Moore the *disinfecting* state propagates vertically in both directions on the first column, and since the maximal distance between initially consecutive disinfecting cells is d_{max} by construction, within $\lfloor \frac{d_{max}-1}{2} \rfloor$ time units all the cells on column 0 are either *disinfecting* or *disinfected*. According to the local rules, disinfection then propagates to column 1. Since the propagation to column 1 happened for all

disinfecting cells within $\lfloor \frac{d_{max}-1}{2} \rfloor$ time units from the beginning, the distance between any two consecutive entry points in column 1 cannot be greater than d_{max}. By a similar argument as for column 0, all cells in column 1 will then become *disinfected* or *disinfecting* within other $\lfloor \frac{d_{max}-1}{2} \rfloor$ time units. Thus, within at most d_{max} time units, all the cells in column 0 and in column 1 will be either *disinfected* or *disinfecting.*

2. Induction hypothesis: At some point during the computation, assume all cells of column i $(0 < i < n-1)$ and all their right neighbours in column $i+1$ are either in *disinfecting* or *disinfected* state, their left neighbours in column $i-1$ are clean, and the entry points in column $i+1$ are at distance at most d_{max}.

3. Induction Step: Consider column $i+1$. By induction hypothesis we know that there is a time t when all cells in column $i-1$, are clean, the ones in columns i, and $i+1$ are either *disinfecting* or *disinfected* and the entry points in column $i+1$ are at maximum distance d_{max} from each other. It follows that the disinfection propagates vertically in column $i+1$ within $\lfloor \frac{d_{max}-1}{2} \rfloor$ time units from time t, thus leaving all cells in column i clean. Disinfection propagates then to column $i+2$ and within other $\lfloor \frac{d_{max}-1}{2} \rfloor$ time units all the cells in the column $i+2$ become either disinfected or disinfecting. We can conclude that, by time $t+d_{max}$ all cells in column $i+1$ together with all their right neighbours are either *disinfected* or *disinfecting* and the cells in column i are clean, thus concluding the proof.

Placing $k > 1$ disinfecting sites roughly equidistant on the first column we obtain as a corollary that:

Corollary 1. *With $k > 1$ disinfecting site per time unit, temporal disinfection can be achieved when the immunity time is at least $\lfloor \frac{2(2n-1)}{k} \rfloor$.*

For the case of $k = 1$ when the starting disinfecting cell is a corner, with an argument similar to the one of Theorem 2 we can show that the strategy is optimal:

Theorem 5. *With a single disinfecting site per time unit, temporal disinfection is possible if and only if the immunity time is is at least $2n-1$.*

5 Note on Circular CAs

In this Section we briefly discuss how the results for finite CAs can be extended to circular CAs.

In the case of *basic disinfection*, the decontamination technique can be easily extended to circular CAs with Von Neumann neighborhood with the same immunity time, doubling the number of simultaneously disinfecting sites. The n initial disinfecting sites are placed in the cells of any column and the cells obey the following simple rules: regardless of the neighbourhing sites' state, a *disinfecting* site becomes disinfected; an *unprotected* site becomes *disinfecting* at time $t+1$ if one or both its horizontal neighbours (left/right) are *disinfecting* at time t. It is easy to see that the effect of the local rules is the sequential disinfection of pairs of columns, where a disinfected site is never in contact with an unprotected one, and we have:

Theorem 6. *Basic disinfection can be achieved in a circular CA with Von Neumann neighborhood using* $2n$ *simultaneous disinfecting sites.*

Also in the case of *temporal disinfection* in CAs with Moore neighbourhood we can easily modify the strategy employed for the finite case to obtain disinfection with the same immunity time, doubling the number of simultaneously disinfecting sites. We place the initial disinfecting cells on the first column (at distance greater than 1 from each other). We design the rules in such a way that the disinfection propagates in *both* directions (up and down) on the first column until two disinfecting nodes become adjacent (or a disinfecting node has the two neighbours on the column disinfected). At this point the rules will let the disinfection propagate in both directions (right and left) to the adjacent columns. The procedure continues like for the finite case, but in both directions until two disinfected columns become adjacent or a column has both left and right column neighbours disinfected. The rules are described in Table 5.

Table 5. Rule Table for Circular CA with temporal immunity and Moore neighborhood. All missing combinations of states for the neighbourhood of cell (i,j) do leave $x_{i,j}$ unchanged.

Temporal-Circular-Moore

Configuration	Next State
$\{\bullet, *, *, *, *, *, *, *, *\}$	0
$\{1, 1, 1, \bullet, 1, 1, 1, 1, 1\}$	$\bullet$
$\{1, 1, 1, 1, 1, 1, 1, \bullet, 1\}$	$\bullet$
$\{1, 1, 1, \bullet, 1, 1, 1, \bullet, 1\}$	$\bullet$
$\{1, \bullet, 0, 1, 1, 1, 1, 1, \bullet\}$	$\bullet$
$\{1, \bullet, 0, 1, 1, 1, 1, 1, 0\}$	$\bullet$
$\{1, 1, 1, 1, 0, \bullet, 0, 1, 1\}$	$\bullet$
$\{1, 1, 1, 1, \bullet, \bullet, 0, 1, 1\}$	$\bullet$
$\{1, 0, 0, \bullet, 1, 1, 1, 1, 0\}$	$\bullet$
$\{1, 0, 0, 1, 1, 1, 1, \bullet, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 1, 1, 1, \bullet, 0\}$	$\bullet$
$\{1, 1, 1, 1, 0, 0, 0, \bullet, 1\}$	$\bullet$
$\{1, 1, 1, \bullet, 0, 0, 0, 1, 1\}$	$\bullet$
$\{1, 1, 1, \bullet, 0, 0, 0, \bullet, 1\}$	$\bullet$
$\{1, \bullet, 0, 1, 0, 0, 0, 1, 0\}$	$\bullet$

Configuration	Next State
$\{1, 0, 0, 1, \bullet, \bullet, 0, 1, \bullet\}$	$\bullet$
$\{1, 0, 0, 1, 0, \bullet, 0, 1, \bullet\}$	$\bullet$
$\{1, 0, 0, \bullet, 1, \bullet, 0, 1, 0\}$	$\bullet$
$\{1, 0, 0, 1, 0, \bullet, 1, \bullet, 0\}$	$\bullet$
$\{1, \bullet, 1, \bullet, 0, 0, 0, 1, 0\}$	$\bullet$
$\{1, 0, 0, 1, \bullet, \bullet, 0, 1, \bullet\}$	$\bullet$
$\{1, \bullet, 0, 1, 0, 0, 0, 1, 1\}$	$\bullet$
$\{1, 0, 0, 1, \bullet, 1, 1, \bullet, 0\}$	$\bullet$
$\{1, 1, \bullet, 1, 0, 0, 0, \bullet, 1\}$	$\bullet$
$\{1, \bullet, 0, 1, 0, \bullet, 0, 1, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 0, 0, 0, 1, 0\}$	$\bullet$
$\{1, 0, 0, 1, 0, 0, 0, \bullet, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 0, 0, 0, \bullet, 0\}$	$\bullet$
$\{1, 0, 0, 0, 0, 0, 0, \bullet, 0\}$	$\bullet$
$\{1, 0, 0, \bullet, 0, 0, 0, 0, 0\}$	$\bullet$

Let d_{max} be the maximum distance between two consecutive *disinfecting* cells or between a *disinfecting* cell and a corner at time 0. Analogously to the case of finite CAs, it is easy to see that, If the initial number of disinfecting sites is at least 2, the set of rules Temporal-Circular-Moore achieves disinfection with immunity time $t = d_{max}$. It then follows that:

Theorem 7. *With $k > 2$ disinfecting sites per time unit, temporal disinfection can be achieved in a circular cellular automata when the immunity time is at least $\lfloor \frac{(2n-1)}{k} \rfloor$.*

Interestingly, for circular CAs with Von Neumann neighborhood, a straightforward extension of the technique described for the finite case cannot be proposed. The symmetry of the CA and the small neighborhood heavily limit the possibilities of disinfection. For example, we can show that:

Theorem 8. *In circular CAs with Von Neumann neighborhood temporal disinfection is not possible with a single disinfecting site per time unit, for any immunity time.*

The general case of $k > 1$ is under investigation.

6 Conclusions

In this paper, we have considered the problem of decontaminating a 2-dimensional 3-states cellular automata with and without temporal immunity. We have focused on finite cellular automata with Von Neumann and Moore neighborhoods. In each case we have described CA rules to achieve decontamination. We have measured the efficiency of our sets of rules by considering the number of simultaneous sites in *disinfecting* state and, with this metric, we have shown that most rules are optimal. We have also observed how to extend the rules to the case of circular CAs.

Some problems are still open and currently under investigation. For example, with Von Neumann neighborhood we have described rules achieving optimal disinfection which employ 1,2, and 4 simultaneously disinfecting sites, but no optimal strategy has been proposed for 3 simultaneously disinfecting sites or for any number between 4 and $n-1$. In our study we assume that the location of the initial disinfecting sites can be chosen; another interesting problem would be to devise optimal rules for a given arbitrary initial placement. Finally, we are now investigating the case of circular CAs with Von Neumann neighborhood to determine whether, regardless of the amount of temporal immunity, decontamination is possible at all with less than n agents.

References

1. Barrière, L., Flocchini, P., Fraigniaud, P., Santoro, N.: Capture of an intruder by mobile agents. In: Proc. 14th Symp. Parallel Algorithms and Architectures (SPAA 2002), pp. 200–209 (2002)
2. Bienstock, D., Seymour, P.: Monotonicity in graph searching. J. Algorithms 12, 239–245 (1991)
3. Flocchini, P.: Contamination and decontamination in majority-based systems. Journal of Cellular Automata 4(3), 183–200 (2009)
4. Flocchini, P., Huang, M.J., Luccio, F.L.: Decontamination of hypercubes by mobile agents. Networks 52(3), 167–178 (2008)

5. Flocchini, P., Luccio, F.L., Song, L.X.: Size Optimal Strategies for Capturing an Intruder in Mesh Networks. Communications in Computing, 200–206 (2005)
6. Flocchini, P., Mans, B., Santoro, N.: Tree decontamination with temporary immunity. In: Hong, S.-H., Nagamochi, H., Fukunaga, T. (eds.) ISAAC 2008. LNCS, vol. 5369, pp. 330–341. Springer, Heidelberg (2008)
7. Fomin, F., Golovach, P.: Graph searching and interval completion. SIAM J. on Discrete Mathematics 13(4), 454–464 (2000)
8. Fraigniaud, P., Nisse, N.: Connected treewidth and connected graph searching. In: Correa, J.R., Hevia, A., Kiwi, M. (eds.) LATIN 2006. LNCS, vol. 3887, pp. 479–490. Springer, Heidelberg (2006)
9. Qiu, J.: Best Effort Decontamination of Networks. Ms. Thesis, U. of Ottawa (2007)
10. Ilcinkas, D., Nisse, N., Soguet, D.: The cost of monotonicity in distributed graph searching. Distributed Computing 22(2), 117–127 (2009)
11. Luccio, F., Pagli, L., Santoro, N.: Network decontamination with local immunization. International Journal of Foundation of Computer Science 18(3), 457–474 (2007)
12. Megiddo, N., Hakimi, S., Garey, M., Johnson, D., Papadimitriou, C.: The complexity of searching a graph. Journal of the ACM 35(1), 18–44 (1988)