Journal of Cellular Automata, Vol. 0, pp. 1–18
Reprints available directly from the publisher
Photocopying permitted by license only

Contamination and Decontamination in Majority-Based Systems

PAOLA FLOCCHINI*

School of Information Technology and Engineering, University of Ottawa, Ottawa, Ontario, K1N 6N5, Canada

Received: November 3, 2007. Accepted: February 27, 2008.

The dynamics of majority voting has always been of interest in the area of discrete dynamical systems. In recent years, there has been a growing interest on this process also in the distributed computing field, due to its links to fault-tolerance, reliability, and virus disinfection. In fact, local voting mechanisms are often employed in distributed systems and networks as a decision tool for a variety of applications.

In presence of faults, these schemes can trigger a dynamics of *contamination*: a non-faulty node will exhibit a faulty behavior if the majority of its neighbors is faulty. Some distributed and networked systems employ mechanisms to mend the faults; in these cases a *decontamination* dynamics is present and interacts with the contamination process. Depending on whether the decontamination is carried out by the majority-voting mechanism already in place or by the use of a team of mobile agents, the decontamination process is called *internal* or *external*, respectively.

In this paper we focus on the contamination and decontamination processes in majority based systems and we survey the recent results in presence of both internal and external decontamination.

Keywords: Majority voting, majority rule, dynamic monopolies, dynamos, decontamination.

1 INTRODUCTION

Discrete dynamical systems, and in particular Cellular Automata have been often employed to describe, model, analyze and investigate situations arising in a variety of application domains. This has been recently the case in the context of fault-tolerance of distributed systems and networks.

*E-mail: flocchin@site.uottawa.ca

Distributed and networked systems often employ *local majority* based rules to enhance reliability and fault-tolerance. Indeed, local voting schemes are used as a decision tool in a variety of different applications. For example, majority voting among the participants has been employed in algorithms for agreement and consensus in distributed environments. In distributed databases management, inconsistencies are commonly resolved by majority voting. Voting has also been used to enforce data consistency when updating copies of the same data by forcing changes of majority-based quorum systems. In the context of resource allocation, majority is typically employed to ensure mutual exclusion to dedicated resources. Local voting schemes are also used for key distribution in security, and reconfiguration under catastrophic faults in system level analysis. Systems employing majority-based local voting schemes have clearly a higher level of resistance e.g., to virus contamination: an un-contaminated site will avoid contamination as long as a majority of its neighbours is un-contaminated.

In spite of the higher reliability, in majority-based distributed processes like the one mentioned above, faulty elements can still induce a faulty behavior in their neighbors. This is for example the situation in distributed systems where majority voting among various copies of crucial data are performed between neighbours at each step: if the majority of its neighbors is faulty (i.e., has corrupted data), a non-faulty element will exhibit a faulty behavior (i.e., its data will become corrupted) and will therefore be indistinguishable from a faulty one.

The study of the effectiveness of using local-majority voting to achieve reliability and fault tolerance in distributed and networked systems has recently been the subject of intensive theoretical research. In this research, the system is modeled as a simple undirected connected graph $G = (V, E)$ of size n where nodes are colored black or white (i.e, have a Boolean state), and each node re-colors itself at discrete time steps on the basis of the majority of the colors held by its neighbours. Different variants of the model can be identified depending on the action to be taken in case of tie, or depending on whether the node consider itself or not when applying the majority rule. Such a process has been studied also in relation to other applications: in fact, it could model spread of information, diffusion of diseases, epidemics, influence and flow of information in various environments, such as societies, genetic processes.

The spreading of a specific color, say black, in such models is traditionally referred to as *contamination*. By assigning such a color to the initially faulty elements, the study of the contamination process provides insights on the degree of reliability and fault-tolerance of the system under examination. Some systems provide mechanisms to restore faulty elements to a normal functioning; the use of these mechanisms is traditionally referred to as *decontamination*. There are two main types of decontamination: in the first type (e.g., in case of software malfunctions) a local faulty behavior can be "mended" by the existing local-majority mechanism – with possibly different rules (*internal*

decontamination); in the second type, when the voting mechanisms alone are not effective (e.g., in case of viruses), the decontamination process is carried out by external agents, called *cleaners* (*external decontamination*). In absence of decontamination mechanisms, contamination is the only process occurring in the system; in presence of decontamination, the contamination process exists in the same space and time domain as the decontamination process, their interaction creating a dynamic of faults propagation and their mending.

The study of the dynamics of these processes is a crucial first step in the analysis of these systems, for example to determine whether the initial impact of the initial faults will be limited in scope (e.g., a bichromatic fixed point will be reached) or leads to a collapse of the entire system (i.e., a monochromatic black fixed point will be reached); to determine, in case the system provides a decontamination mechanism, whether the decontamination will be successful (i.e., a monochromatic white fixed point is reached) or not, and if not whether the system will be forever unstable.

In addition to the dynamics, the most important questions are quantitative or decisional or both. For systems without decontamination or with internal decontamination, the research in the distributed computing community has been focused almost exclusively on the patterns of initial faults that lead to a monochromatic black fixed point (i.e., lead the entire system to a faulty behavior); the focus is on the identification and characterization of these patterns, called *dynamos*, and on questions about their size. External decontamination has been investigated more in the context of cleaning a network from viral infections and the typical problem has been the determination of the smallest possible team of external agents necessary to perform a full decontamination (i.e., reaching white fixed points) and the design of the decontamination strategy.

This paper is a survey reviewing recent results on contamination and decontamination in majority based systems. More precisely, in Section 2 we introduce the basic terminology; in Section 3 we consider solely the contamination process; in Section 4 we focus on systems with an embedded local-majority mechanism of decontamination; in Section 5 we consider systems where the decontamination process is carried out by external agents. In all cases we reviewing recent results.

2 MODELS AND TERMINOLOGY

Let $G = (V, E)$ be a simple undirected connected graph of size n where nodes $v_1, v_2, \ldots, v_n$ are colored black or white (i.e., have boolean states). Let $x(v_i) \in \{0, 1\}$ be the state of node v_i, where 0 corresponds to white and 1 corresponds to black.

A node subject to *majority voting* updates its color, assuming the colour held by the majority of its neighbours. The update is performed simultaneously at

discrete time steps by all nodes subject to majority voting; that is, the dynamics is *synchronous*. In the definition of majority voting it is very important to define what action a node should take in case of tie. Possible actions (as defined in [43]) are: prefer-white (PW), prefer-black (PB), prefer-current (PC), prefer-flip (PF). Another distinction regards whether the node making the decision is included in the computation of the majority: self-including (SI), self-not-including (SN). Depending on the combination of the various parameters, different models, often with quite different dynamics, can be defined (sometimes indicated as (PW, SI), (PB, SI), etc.). Notice that model (PB, SN) is usually called *simple majority*, while (PC, SN) is called *strong majority*.

A *contamination* rule is a local majority-based rule applied to white nodes only. In this paper we consider only system with contamination. An *internal decontamination* rule is a local majority-based rule applied only to black nodes. Hence, in systems with internal decontamination, majority rule is applied to all nodes. On the other hand, an *external decontamination* (also called *cleaning*) is defined by a different process: external entities (called agents) move on the graph from node to neighbouring node and a black node can become white only if an agent moves on it. Internal and external decontaminations are quite different processes and they are typically employed in very different context (the first in majority-based distributed systems, the second in networks infected by a virus).

A *Dynamic Monopoly* (*Dynamo* for short) for a given contamination (and decontamination) rule is an initial configuration leading to *monochromatic* black fixed point under that rule. The dynamics is said to be *monotone* if the set of black vertices at any time t is a subset of the one at time $t + 1$.

3 CONTAMINATION

We will first consider contamination in systems where no decontamination mechanism is in place. In the context of fault tolerance, this process describes the impacts that permanent faults have in distributed majority-based systems. In these systems, dynamos are also called *irreversible* because the black nodes cannot change their color. The research focus is on the determination of the smallest possible size for a dynamo and sometimes on the characterization of the dynamos patterns in particular topologies: chorded rings (a special type of one-dimensional Cellular Automata), tori (two-dimensional Cellular Automata with various boundary conditions), and some common interconnection networks.

3.1 Chorded rings

Chorded rings are a special type of one dimensional circular Cellular Automata. A *chorded ring* $C(\langle 1, 2, \ldots, p, k\rangle)$ (with $p < k$) of size n is a ring on n nodes $x_0, x_1, \ldots, x_{n-1}$ where each node x_i is connected to nodes

FIGURE 1
A portion of a chorded ring $C(\langle 1, 2, 3, 6\rangle)$ (the chords are shown only for one node).

$x_{i+1} \cdots x_{i+p}$ and x_{i+k} (all operations on the indices are modulo n). Examples of chorded rings are rings ($p = k = 1$), double-loop networks ($p = 1; k > 1$), fan networks ($p = k - 1$), and complete graphs ($p = k = \lfloor \frac{n}{2} \rfloor$). Depending on the relationship between p and k a chorded ring will be said to be *weakly* ($p < \frac{k}{2}$) or *strongly* ($p \geq \frac{k}{2}$) *chorded*. Notice that this type of topology is quite popular (e.g., in peer-to-peer systems): the p neighbours provide redundancy, while the kth neighbour is typically useful for routing purposes. A chorded ring can also be seen as a k-neighbours one dimensional circular CA where the neighbours at distance $p+1, p+2, \ldots, k-1$ do not influence the evolution rule.

Contamination in chorded rings has been studied under the simple majority rule (or, model (PB,SN)) in [13]. In particular, the focus has been on the determination of dynamos which occupy the minimum possible "window" (smallest number of consecutive nodes) and containing the minimum number of faults (minimum *size*). Such dynamos are called *optimal*. First of all, it has been shown that the smallest dynamo pattern must occupy a window of k consecutive nodes in any chorded ring $C(\langle 1, 2, \ldots, p, k\rangle)$, then the problem of determining the minimum dynamos has been studied separately for weakly and strongly chorded rings fixing the length of the window to k.

Weakly-chorded rings. ($C(\langle 1, 2, \ldots, p, k\rangle)$ with $p < \frac{k}{2}$). In [13] it has been shown that $p + 1 + \lceil \frac{(k-2p-1)}{p-1} \rceil$ is a tight bound on the size of the optimal dynamos. The general construction of optimal dynamos can be carried out by identifying an *initiating pattern* followed by an appropriate number of *filling patterns* in a window of length k. Intuitively, the initiating pattern is a pattern whose presence initiates the propagation of blacks; a filling pattern is a pattern that allows the propagation to continue until the k-window is all black. For example, Figure 2 shows an optimal dynamo for $C(\langle 1, 2, 3, 18\rangle)$ (only a portion of the chorded ring is shown and only the connections of one node): the first $2p + 1$ nodes (in this case seven) constitute the initiating pattern, which is contaminated in three steps; after that, the filling patters will be contaminated sequentially, one node at the time, until the whole window

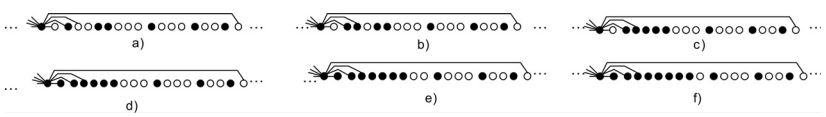

FIGURE 2
The first five steps in the evolution of an optimal dynamo for $C(\langle 1, 2, 3, 18\rangle)$.

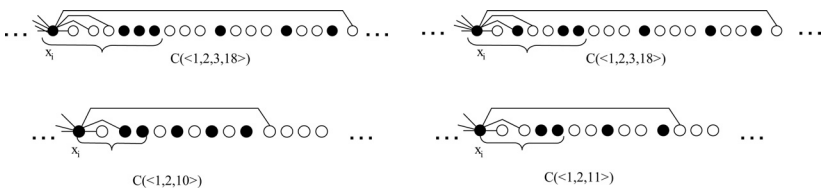

FIGURE 3
Example of optimal dynamos in various chorded rings. The portion underlined with a parenthesis is the initialing pattern.

is black. As soon as this happens, the contamination will continue for the rest of the chorded ring. Other examples of optimal dynamos in various chorded rings are shown in Figure 3.

In the particular case of *double-loops* $C(\langle 1, k\rangle)$, the minimum size is $\lceil \frac{(k+1)}{2} \rceil$. Moreover, for this case the optimal dynamos can be precisely described: to be an optimal dynamo a pattern of length k must not contain two consecutive 0s and must contain $\lceil \frac{(k+1)}{2} \rceil$ 1s. All and only such patterns are: $(10)^{\lfloor \frac{k}{2} \rfloor}(1)$ for k odd, and $(10)^a(1)(10)^b(1)$ with $a+b=(k-1)/2$, for k even. In the case of *triple-loops* $C(\langle 1, 2, k\rangle)$ the minimum size is $\lceil \frac{(k+4)}{3} \rceil$ and a complete characterization of the patterns has been provided: any pattern of length k that contains two consecutive 1s and that does not contain three consecutive 0s is a dynamo and, vice versa any minimum size dynamo of length k in $C(\langle 1, 2, k\rangle)$ must contain two consecutive 1s and must not contain three consecutive 0s. Such patterns have been enumerated.

Strongly-chorded rings. $(C(\langle 1, 2, \ldots, p, k\rangle)$ with $p \geq \frac{k}{2})$. In this case it has been shown that the size of the optimal dynamos is $p+1$; some conditions have been derived, but no complete characterization of the patterns exist. Special is the case $p = k-1$ where the resulting graph is called *fan* graph and which corresponds exactly to a one dimensional CA with neighborhood k. In such a case, in fact, there is only one optimal dynamo, which consists of k consecutive nodes with value 1.

3.2 Tori

A torus is a bi-dimensional Cellular Automata with circular connections. Dynamos in tori with different wrap-around connections have been studied in [18] as they represent one of the simplest and most natural way of connecting processors in a network. The authors have considered simple and strong majority in: (1) *toroidal mesh* (the last node of a row/column is connected to the first of the same row/column); (2) *torus cordalis* (equivalent to double-loop networks, where the last node of a row is connected to the first node of the next row while the last of a column is connet to the first of the same column); (3) *torus serpentinus* (last node of a row/column is connected to the first node of the next row/column).

Most results are based on the determination of *immune subgraph* (or *white blocks*) and *black compacts*. A white block for a certain type of majority is a pattern of white nodes whose presence forbids the evolution to a monochromatic black fixed point. A black compact is a pattern of black nodes that is instead necessary to start the propagation of black nodes so to evolve to a monochromatic black fixed point. Most lower bounds for the tori are based on combinatorial argument on the necessity of having certain black compacts while forbidding white blocks. The bounds are summarized in Tables 1, 2 and they are all are *tight* within an additive constant; in fact, dynamos almost matching the lower bounds have been constructed. Figure 4 shows examples

	Simple Majority	
	Lower Bound	*Upper Bound*
Toroidal mesh	$\lceil \frac{a+b}{2} \rceil - 1$	$\lceil \frac{a+b}{2} \rceil - 1$
Torus cordalis	$\lceil \frac{b}{2} \rceil$	$\lfloor \frac{b}{2} \rfloor + 1$
Torus serpentinus	$\lceil \frac{\min\{a,b\}}{2} \rceil$	$\lfloor \frac{\min\{a,b\}}{2} \rfloor + 1$

TABLE 1
Bounds on the size of irreversible dynamos for toroidal mesh, torus cordalis, and torus serpentinus of $a \times b$ vertices in the case of simple majority

	Strong Majority	
	Lower Bound	*Upper Bound*
Toroidal mesh	$\lceil \frac{ab+1}{3} \rceil$	$\min\{\lceil \frac{a}{3} \rceil (b+1), \lceil \frac{b}{3} \rceil (a+1)\}$
Torus cordalis	$\lceil \frac{ab+1}{3} \rceil$	$\lceil \frac{a}{3} \rceil (b+1)$
Torus serpentinus	$\lceil \frac{ab+1}{3} \rceil$	$\min\{\lceil \frac{a}{3} \rceil (b+1), \lceil \frac{b}{3} \rceil (a+1)\}$

TABLE 2
Bounds on the size of irreversible dynamos for toroidal mesh, torus cordalis, and torus serpentinus of $a \times b$ vertices in the case of strong majority

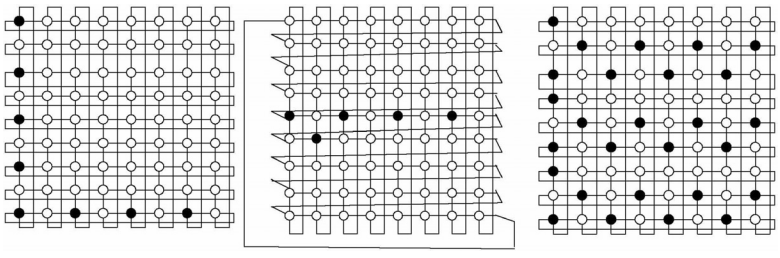

FIGURE 4
Optimal dynamos for: simple majority in (a) toroidal mesh and (b) torus cordalis; (c) strong majority in toroidal mesh.

of optimal dynamos for simple majority in a toroidal mesh and in a torus cordalis, and for strong majority in a toroidal mesh (this last construction holds also for the case of torus serpentinus).

3.3 Common interconnection networks

Several typical interconnection networks have been investigated under the *simple majority* rule in [16, 35]. In [35] the authors studied Butterflies, Wrapped Butterflies, and Cube Connected Cycles. The summary of the known bounds for some interconnection networks is shown in Table 3. Let us briefly describe, for example, the results on the Butterfly. A Butterfly $BF(d)$ is composed by $d+1$ rows each containing 2^d vertices connected as in Figure 5.

In a wrapped butterfly $WBF(d)$ the first and the last row coincide. It is shown in [35] that an optimal irreversible dynamo in $BF(d)$ has size at least $2^{\lfloor \frac{(d-1)}{2} \rfloor}$ while upper bounds have been constructed with size 2^{d-2} (the construction is shown in Figure 5(a)). In the case of $WBF(d)$, an optimal irreversible dynamo

	Lower Bound	Upper Bound
$BF(d)$	2^{d-2}	2^{d-2}
$WBF(d)$	$2^{\lfloor \frac{d}{2} \rfloor}$	$2^{d-2} + 2^{d-3} + 2^{d-4}$
$CCC(d)$	$\max\{\lfloor \frac{d+1}{2} \rfloor \cdot 2^{d-2}, 2^d\}$	$n \cdot 2^{d-2} + 2^{d-3}$
$H(d)$	$\lceil \frac{2^d}{d+1} \rceil$	$O\left(\frac{2^d}{\sqrt{d}}\right)$

TABLE 3
Bounds on minimum dynamos size for some interconnection networks

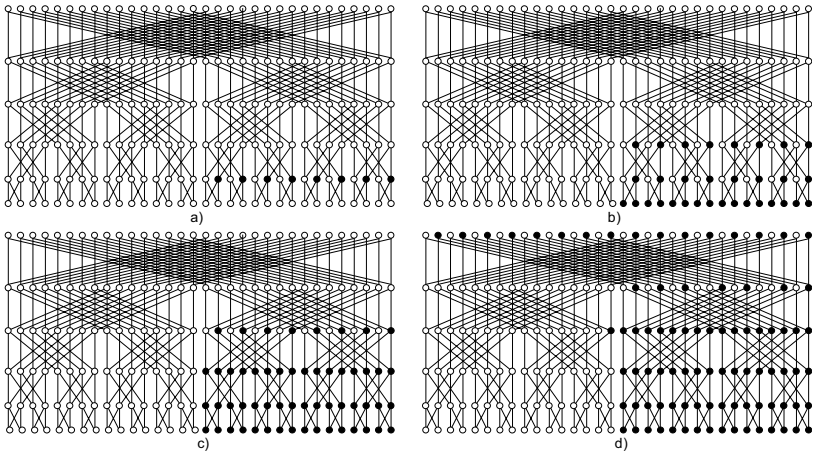

FIGURE 5
(a) Optimal dynamo in a butterfly. The first four steps in the contamination process are shown. This butterfly ($BF(5)$) is fully contaminated in 10 steps.

	Lower Bound	Upper Bound
d-regular	$\lceil \frac{n}{2t+1} \rceil$	?
Ring, Tori, *WBF*	$\lceil \frac{n}{2t+1} \rceil$	$\lceil \frac{n}{2t+1} \rceil$
Cube Connected Cycle (for $t > 5$)	$\lfloor \frac{n}{8} \rfloor$	$\lceil \frac{n}{4} \cdot (1 + \frac{3}{4(t-2)}) \rceil$
Hypercube ($n = 2^d$)	$\lceil \frac{n}{2t+1} \rceil$	$\lceil \frac{n}{(t+1)} \rceil - \lceil \frac{n}{\sqrt{\pi d/8}} \rceil$
DeBrujin	$\lceil \frac{n}{2t+1} \rceil$	$\lceil \frac{n+1}{4}(1 + \frac{1}{2^t-1}) \rceil$

TABLE 4
Bounds on minimum dynamos size as a function of completion time t in some interconnection networks of size n. The bounds hold for irreversible dynamos with simple majority

has size at least $2^{\lfloor \frac{d}{2} \rfloor}$ and the upper bound is $2^{d-2} + 2^{d-3} + 2^{d-4}$. In both cases a large gap between lower and upper bound was left. The gap has been closed in [16] for the case of $BF(d)$ by raising the lower bound to 2^{d-2}. It is still open how to close the gap for the wrapped butterfly.

Another issue that has been studied in interconnection networks is the relationship between the size of a dynamo and the time for the system to collapse [16]. In fact, depending on the topology, various trade-offs between the two measures have been derived. Let us call *t-time dynamo* a dynamo that leads the system to a monochromatic black fixed point in a time bounded by t. One of the results of [16] shows that in regular graphs with n nodes, the size of a minimal t-time irreversible dynamo is at least $\lceil \frac{n}{2t+1} \rceil$. This bound is tight for rings, tori, wrapped butterflies. Other bounds are summarized in Table 4.

3.4 Other issues

Immune subgraphs. Immune subgraphs have been introduced in [43] where the connection between immunity and expansion has been studied. In Section 3.2 we have mentioned that they can be employed to determine lower bounds on the minimum dynamos size. In [28] immune subgraphs have been further investigated focusing on the question of determining the size of the smallest immune subgraph in a given graph. The *immunity index* of a graph has been defined as the least integer c_1 such that each configuration of size c_1 is immune. The *catastrophic index* c_2 is the smallest integer such that each configuration of size c_2 is a dynamo. It is shown that determining whether a graph has an immune subgraph of a certain given size is NP-complete. Immunity and catastrophic indices are then studied systematically for various topologies (tori, hypercubes, butterflies, cube connected cycles ...). These indices have been derived also for dynamos evolving to the monochromatic configuration in a given time t deriving tight trade-offs.

Unanimity rule. Of particular relevance in this context are also the existing studies on *catastrophic fault patterns* and *deadly sets* (e.g., see [10, 41, 42]).

In fact, these patterns are irreversible dynamos for directed graphs under *directional unanimity* (as opposed to majority rule): a node becomes black if all its in-neighbours (or out-neighbours) are black. They have been studied in the context of VLS design to analyze the limits of using link redundancy to achieve fault-tolerance in linear arrays.

4 INTERNAL DECONTAMINATIONS

When internal decontamination mechanisms are in place, a local faulty behavior can be mended by the existing local-majority mechanism: that is, in systems with internal decontamination, all nodes are subject to the majority rule. In this case, dynamos are also called *reversible*. This is the "classical" model studied in most of the literature on this subject. A comprehensive survey on this topic already exists (see [44]) so we will only briefly sketch the major results. Similarly to the irreversible case, determination of minimum size dynamos has been one of the major focus. However, in most cases, the study has been restricted to *monotone reversible* dynamos, where decontamination is ineffective. In fact, in this case decontamination is in principle possible, however the dynamos must be designed so to guarantee that a black node never happens to be in the condition to become white (i.e., decontamination has no effect).

4.1 Dynamics

The majority rule has been studied extensively in the context of discrete dynamical systems, for example, in neural networks (e.g., see [23, 25]). Let $X^t = (x^t(v_1), \ldots, x^t(v_n))$ be the global configuration at time t of the n vertices $v_1, \ldots, v_n$ of a graph, where the majority rule is applied synchronously, at discrete time steps to all vertices. It has been shown by Goles that, for finite graphs, any sequence $\{X^t\}$ reaches a period of length at most two [24]. This interesting behavior has been actually shown to hold for a more general setting (for example, when the majority function is replaced by a threshold function and when the edges have weights) [25, 47, 48]. This property has been generalized by Moran in [39] for majority rules over locally finite connected graphs, for which a sufficient condition is given in terms of the "rate of growth" of the graph. The property has been generalized further to include local periodicity [22], and also in this more general case a sufficient condition is provided.

The ring has received special attention due to its application for modeling biological processes such as the immune system, interaction between cells, drug scheduling, gene rearrangement. In such cases the interest in fixed-points is motivated by experiments in molecular biology which have shown that even very large gene expression networks have only a few stable structures. Several problems were studied in rings, among these counting the number of fixed points (e.g., see [1, 2, 26]). For example, in [2] it is shown that the number of fixed points is only an exponentially small fraction of all configuration. The

number of fixed points have been studied also in trees in [29] where results analogous to the ones for the ring are shown. Infinite sequences have been the object of extensive investigation by Moran ([37–39]) who has provided conditions for the period-2 property to hold for the particular case of the ring.

4.2 Size of dynamos

The study of reversible dynamos in distributed computing has been introduced by Peleg [45], whose concern has been mainly the determination of the smallest size for a dynamo. His work started with the study of 1-time dynamos, where the monochromatic fixed point has to be reached in a single step (e.g., see [4, 43]). A variety of results, mostly on the minimum size of 1-time dynamos both in general graphs and in specific topologies have been derived, also considering a more general case when the majority is performed on a r-neighbourhood. For an extensive and comprehensive survey of all the existing results on this subject see [44].

The choice of action in case of tie can dramatically influence the dynamics of the system. Moreover, it is very clear that requiring monotonicity in the process also strongly influence the dynamics. For example, Peleg proved in [45] a lower bound of $\Omega(\sqrt{n})$ on the size of *monotone* dynamos in most variant of the models, while Berger [5] proved that, without imposing monotonicity, for every $n \geq 1$ there exists a graph G of n or more vertices with a dynamo of size $O(1)$ in all models.

Some specific topologies have been studied also in the reversible model, but always with the monotonicity condition, that is when decontamination is ineffective. In this context, tori [18] and some interconnection networks have been studied, where trade-offs between time and size have been determined [16]. In Tables 5 and 6 the bounds for tori under reversible (but monotone) majority rule are summarized. Notice that the bounds for irreversible dynamos are smaller by a factor of two than the ones for reversible monotone dynamos (compare with Tables 1 and 2). In the case of strong majority the constant becomes $\frac{3}{2}$. This raises the intriguing question of wether it is possible to transform an irreversible dynamo into a monotone one using at most twice the number of initial black nodes ($\frac{3}{2}$ in the case of strong majority).

	Simple Majority	
	Lower Bound	*Upper Bound*
Toroidal mesh	$a + b - 2$	$a + b - 1$
Torus cordalis	$b + 1$	$b + 1$
Torus serpentinus	$\min\{a, b\} + 1$	$\min\{a, b\} + 1$

TABLE 5
Bounds on the size of reversible monotone dynamos for tori of $a \times b$ vertices with simple majority

	Strong Majority	
	Lower Bound	*Upper Bound*
Toroidal mesh	$\lceil \frac{ab+1}{2} \rceil$	$\lceil \frac{ab}{2} + \frac{\min\{a,b\}}{6} + \frac{2}{3} \rceil$
Torus cordalis	$\lceil \frac{ab+1}{2} \rceil$	$\lceil \frac{ab}{2} + \frac{\min\{a,b\}}{6} + \frac{2}{3} \rceil$
Torus serpentinus	$\lceil \frac{ab+1}{2} \rceil$	$\lceil \frac{ab}{2} + \frac{\min\{a,b\}}{6} + \frac{2}{3} \rceil$

TABLE 6
Bounds on the size of reversible monotone dynamos for tori of $a \times b$ vertices with strong majority

5 EXTERNAL DECONTAMINATION

Decontamination by mobile agents has been extensively studied in various models in the past thirty years. The underline contamination dynamics can be very different depending on the application, while the decontamination process usually works in the same way, as follows. At any time nodes can be *contaminated*, *clean*, or *guarded* (if they contain at least an agent). All nodes are initially contaminated except for one (the *homebase*) where a team of mobile agents is located. Agents can move in the network from a node to a neighboruing node and a contaminated node is transformed into clean when an agent passes by it. The goal is to reach a state when all nodes are clean (or guarded)*.

As opposed to the dynamics of the previous two Sections, which have been studied in synchronous settings, when dealing with external decontamination the process has been considered also (and especially) in asynchronous environments where the actions of the agents, as well as the contamination of the nodes, occur independently. Moreover, in all cases, the schedule is deterministic and the worst case is assumed.

A strategy is called *monotone* if it guarantees that once clean, a node will never be contaminated again (the set of clean nodes at time t includes the set of clean nodes at any previous time). In all models the main difficulty derives from the fact that after being cleaned, a node can get re-contaminated if some of its neighbours are contaminated. Thus, the decontamination strategy has to perform the cleaning while avoiding too much recontamination, or avoiding recontamination all together (for monotone models). Decontamination has to be executed as efficiently as possible. Efficiency is measured in terms of the size of the team of agents, traffic (i.e., the number of moves the agents have to perform), and, in case of synchronous settings, time.

*In a classical model extensively investigated in graph theory, agents can "jump" from a node to any other node in the graph giving rise to the so called *graph search* problem (e.g., see [40]), which we do not discuss here.

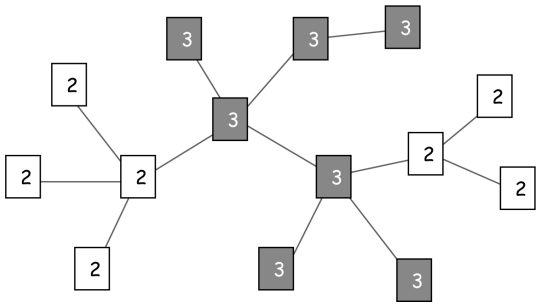

FIGURE 6
A tree with the indication of the minimum number of agents for each starting node.

Contamination by contact. As mentioned, the contamination rule depends on the application; the most commonly contamination rule employed is "contamination by contact": *a clean node with at least a contaminated neighbour becomes contaminated*. The problem of determining the optimal number of agents necessary to perform the decontamination in arbitrary topologies is NP-hard. It has been studied in some specific topologies in the case of monotone strategies, both in synchronous and asynchronous settings. All the results below are for the case of asynchronous evolution. For example, it has been shown that it can be solved in linear time in trees [6], where the location of the homebase influences the required number of agents. In the example of Figure 6 the minimum number of agents necessary and sufficient to decontaminate the tree is indicated for each possible starting location. Optimal strategies have been studied in chordal rings, tori, and meshes [14, 17], where, besides determining optimal bounds, the authors have studied the impact on efficiency of increasing power capabilities of the agents (like, for example, the possibility of "seeing" the state of their neighbours). Arbitrary topology networks have also been considered: some heuristics have been proposed in [19] and an exponential move and time solution has been described in [8] to determine an optimal strategy. Finally, several interesting properties of the decontamination process and on the relationship between various models have been investigated in [7, 20, 21]. A topology that has been studied and is perhaps of particular interest in the cellular automata community is the Sierpiński graph $SG(d)$ (see Figure 7). In [33] optimal recursive decontamination strategies for Sierpiński graphs of size $n = \frac{3^d+3}{2}$ are described, which employ $d + 1$ agents.

Contamination by majority. A more general rule that has been recently introduced is the following: *a clean node becomes contaminated if at least p neighbours are contaminated*. The contact rule described above would then be a special case for $p = 1$. This more general rule is quite reasonable to model system with a higher level of resistance to recontamination, such as aystems employing majority-based local voting schemes (in fact, in regular topologies

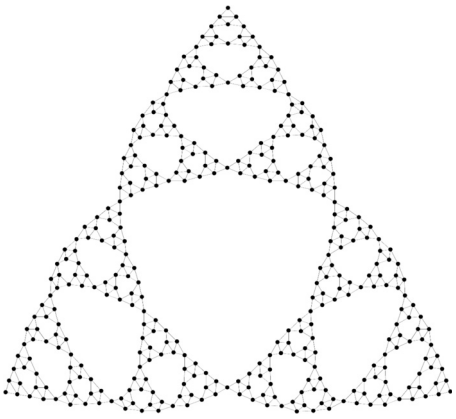

FIGURE 7
A Sierpiński graph.

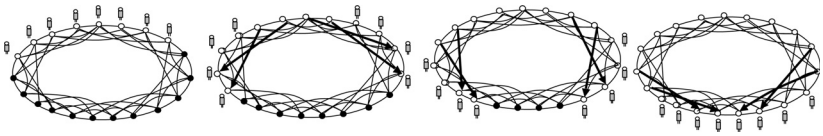

FIGURE 8
A cleaning strategy in the chordal ring when contamination is by contact. The strategy when contamination is by majority is very similar.

with degree deg, when $p = \lfloor \frac{deg}{2} \rfloor$ we have a simple majority rule, when $p = \lceil \frac{deg}{2} \rceil$ we have strong majority). This variant has been introduced in [36] for the simple majority rule, and very little is known. Monotone protocols and lower bounds on the number of cleaners and moves necessary for decontamination have been shown for k-dimensional tori and trees. These preliminary results show that, not surprisingly, the higher resistance has a strong impact of the minimal size of the team of agents required to clean a given graph. For example, a $a \times b$ torus can be decontaminated with a constant number of agents when contamination is majority-based, while $2 \cdot \min\{a, b\}$ agents are necessary when the contamination is by contact. Interestingly, it is easy to see that the change in re-contamination rule does not always have an impact. Consider, for example, the case of chordal rings. Regardless of the re-contamination rule, the number of agents needed to decontaminate the network is $\Theta(k)$, where k is the length of the longest chord [14, 34]. A strategy for decontaminating a chordal ring where contamination is by contact is shown in Figure 8. Another interesting observation concerns tree networks: with contamination by contact the worst possible tree is the binary tree where decontamination requires $\Omega(\log n)$ agents in the worst case [6]; instead, with contamination by majority, binary trees can be decontaminated by a *single* agent [36].

	by contact	by majority
Chordal Ring $C(\langle d_1, d_2, \ldots, d_h = k\rangle)$	$\Theta(k)$	$\Theta(k)$
Torus (size $a \times b$)	$\Theta(\min\{a, b\})$	$\Theta(1)$
Hupercube (size n)	$\Theta(\frac{n}{\sqrt{\log n}})$	unknown
Complete Binary Tree (size n)	$\Theta(\log n)$	$\Theta(1)$
Sierpiński (size n)	$\Theta(\log n)$	$\Theta(1)$

TABLE 7
Summary of bounds on the number of agents in various graphs depending on the contamination rule

6 CONCLUDING REMARKS

6.1 Open problems

Many problems and questions are open in all the three contamination and decontamination dynamics described in this survey. We will mention a few interesting research directions.

In the case of internal decontamination, reversible non-monotone dynamos have not been studied in any specific topology. Interesting would also be the study of asynchronous dynamics in these systems: notice that, in absence of decontamination mechanisms, synchronous or asynchronous updates result in the same evolution, but this is clearly not the case when there is internal decontamination. Furthermore, the case when contamination and internal decontamination follow different rules is totally open. Several computational issues are also open. For example, it is known that determining whether a graph has an immune subgraph of a certain size is NP-complete [28]. It is also known that given a graph, finding a minimum 1-time dynamo is an NP-hard problem [44]. However, these results have not been generalized for t-dynamos, and nothing is known in the case of irreversible dynamos.

In the case of external decontamination by majority almost everything is still open since only a few topologies have been investigated, all of them by employing simple majority: what happens in other classes of networks? Is the presence of majority-based contamination going to cause dramatic improvements in other networks, comparable to the ones observed in toroidal meshes and trees? And if not, why? Another, more fundamental question relates to the monotone nature of the solution protocols. If we remove monotonicity, what would be the minimum number of agents needed for the decontamination?

6.2 Related work

As mentioned in the introduction, majority rules have been studied in relation to various applications: for example, they could describe spread of information, diffusion of diseases, epidemics, influence and flow of information in

different environments, such as societies, genetic processes and distributed systems.

Particular interest in applications of the majority rule can be found in distributed computing. Arguably the largest related area concerns ways for overcoming (benign or malicious) failures by a variety of techniques for reaching agreement between the non-faulty processors of a distributed system (see [50]). In particular, a technique that makes direct usage of a dynamical process in this flavor for reaching agreement in spite of failures can be found in [27]. In distributed databases, management algorithms inconsistency resolution process are commonly resolved by majority voting (e.g., see [9]). Voting has also been used to enforce data consistency when updating copies of the same data by forcing changes of majority-based quorum systems (e.g., see [46]). In the context of resource allocation, majority voting is typically employed to ensure mutual exclusion to dedicated resources (e.g., see [49]).

Confining local failures to the vicinity of their origin and preventing them from spreading in the network has led to the related approach of local mending, which is also based on majorities [3,30,31]. In this case, a function distributed among the nodes of a network could be corrupted due to some transient failures and the goal is to perform distributed mending in time complexity which depend on the number of failed nodes, rather than on the size of the entire network.

Finally, the entire approach of self-stabilization can be viewed as dealing with the same general goal of decontamination, albeit using different techniques [11, 12].

7 ACKNOWLEDGMENTS

This work has been partially supported by NSERC Discovery Grant and by University Research Chair. The author would like to thank the anonymous referee for pointing out some of the related work.

REFERENCES

[1] Z. Agur. Fixed points of majority rule cellular automata applied to plasticity and precision of the immune response. *Complex Systems* (1991), 351–357.

[2] Z. Agur, A. S. Fraenkel and S. T. Klein. The number of fixed points of the majority rule. *Discrete Mathematics* **70** (1988), 295–302.

[3] Y. Azar, S. Kutten and B. Patt-Shamir. Distributed error confinement. *Proceedings of the 22nd Symposium on Principles of Distributed Computing (PODC)*, 2003, pp. 33–42.

[4] J-C Bermond, D. Peleg. The power of small coalitions in graphs. *Proceedings of the 2nd Int. Coll. on Structural Information and Communication Complexity (SIROCCO)*, 1995, pp. 173–184.

[5] E. Berger. Dynamic monopolies of constant size. *J. Comb. Theory Ser. B* **83** (2001), 91–200.

[6] L. Barrière, P. Flocchini, P. Fraigniaud and N. Santoro. Capture of an intruder by mobile agents, *14th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 2002, 200–209.

[7] L. Barrière, P. Fraigniaud, N. Santoro and D. M. Thilikos. Searching is not jumping, *29th Int. Workshop on Graph Theoretic Concepts in Computer Science (WG)*, 2003, 34–45.

[8] L. Blin, P. Fraigniaud, N. Nisse and S. Vial. Distributed chasing of network intruders by mobile agents, *13th Int. Coll. on Structural Information and Communication Complexity (SIROCCO)*, 2006, 70–84.

[9] S. B. Davidson, H. Garcia-Molina and D. Skeen. Consistency in partitioned networks. *ACM Comput. Surveys* **17**(3) (1985), 341–370.

[10] R. De Prisco, A. De Santis. Catastrophic faults in reconfigurable VLSI linear arrays. *Discrete Applied Mathematics* **75** (1997), 105–123.

[11] E. W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM* **17** (1974), 643–644.

[12] S. Dolev. *Self-Stabilization*, MIT Press, Cambridge, MA, 2000.

[13] P. Flocchini, F. Geurts and N. Santoro. Optimal irreversible dynamos in chordal rings. *Discrete Applied Mathematics* **113** (2001), 23–42.

[14] P. Flocchini, M. J. Huang and F. L. Luccio. Decontaminating chordal rings and tori using mobile agents. *Int. Journal of Foundations of Computer Science* **18**(3) (2006), 547–564.

[15] P. Flocchini, M. J. Huang and F. L. Luccio. Decontamination of hypercubes by mobile agents. *Networks*, to appear, 2008.

[16] P. Flocchini, R. Kralovic, P. Ruzicka, A. Roncato and N. Santoro. On time versus size for monotone dynamic monopolies in regular topologies. *J. Discrete Algorithms* **1**(2) (2003), 129–150.

[17] P. Flocchini, F. L. Luccio, L. Song. Size optimal strategies for capturing an intruder in mesh networks. *2005 Int. Conf. on Communications in Computing*, 2005, 200–206.

[18] P. Flocchini, E. Lodi, F. Luccio, L. Pagli and N. Santoro. Dynamic monopolies in tori. *Discrete Applied Mathematics* **137**(2) (2004), 197–212.

[19] P. Flocchini, A. Nayak and A. Schulz. Cleaning an arbitrary regular network with mobile agents. *2nd Int. Conf. on Distributed Computing and Internet Technologies*, 2005, 132–142.

[20] P. Fraigniaud, N. Nisse. Connected treewidth and connected graph searching. *7th Latin American Theoretical Informatics Symposium (LATIN)*, 2006, 479–490.

[21] P. Fraigniaud, N. Nisse. Monotony properties of connected visible graph searching. *32nd Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, 2006, 229–240.

[22] Y. Ginosar, R. Holzman. The majority action on infinite graphs: strings and puppets. *Discrete Mathematics* **215** (2000), 59–71.

[23] E. Goles, S. Martinez. *Neural and automata networks*, Kluwer Academic, Dordecht, 1990.

[24] E. Goles, J. Olivos. Periodic behavior of generalized threshold functions. *Discr. Math.*, **30** (1980), 187–189.

[25] E. Goles, F. Fogelman-Soulie and D. Pellegrin. Decreasing energy functions as a tool for studying threshold networks. *Discrete Applied Mathematics* **12** (1985), 261–277.

[26] A. Granville. On a paper by Agur, Fraenkel and Klein. *Discrete Mathematics*, **94** (1991), 147–151.

[27] Y. Hassin and D. Peleg. Distributed probabilistic polling and applications to proportionate agreement. *Information & Computation* **171** (2001), 248–268.

[28] R. Kralovic, P. Ruzicka. On Immunity and catastrophic indices of graphs. *8th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, 2001, 231–242.

[29] R. Kralovic. On majority voting games in trees. *28th Conf. on Current Trends in Theory and Practice of Informatics (SOFSEM)*, 2001, 282–291.

[30] S. Kutten and D. Peleg. Fault-local distributed mending. *Journal of Algorithms* **30** (1999), 144–165.

[31] S. Kutten, D. Peleg. Tight fault locality. *SIAM Journal on Computing* **30** (2000), 247–268.

[32] N. Linial, D. Peleg, Y. Rabinovich and M. Sachs. Sphere packing and local majority in graphs *2nd Israel Symp. on Theory of Computing and Systems*, 1993, 141–149.

[33] F. L. Luccio. Intruder capture in Sierpiński graphs, *4th Int. Conf. on Fun with Algorithms*, 2007, 249–261.

[34] F. Luccio, L. Pagli Web marshals fightling curly link farms. *4th Int. Conf. on Fun with Alg.*, 2007, 240–248.

[35] F. Luccio, L. Pagli and H. Sanossian. Irreversible dynamos in butterflies. *6th Int. Coll. on Structural Information and Communication Complexity (SIROCCO)*, 1999, 204–218.

[36] F. Luccio, L. Pagli and N. Santoro. Network decontamination in presence of local immunity. *Int. J. of Foundations of Computer Science* **18**(3) (2007), 457–474.

[37] G. Moran. Parametrization for stationary patterns of the r-majority operators on 0–1 sequences. *Discrete Mathematics* **132** (1994), 175–195.

[38] G. Moran. The r-majority vote action on 0–1 sequences. *Discrete Mathematics* **132** (1994), 145–174.

[39] G. Moran. On the period-two-property of the majority operator in infinite graphs. *Transactions of the American Mathematical Society* **347**(5) (1995), 1649–1667.

[40] N. Megiddo, S. Hakimi, M. Garey, D. Johnson and C. Papadimitriou. The complexity of searching a graph. *Journal of the ACM* **35**(1) (1988), 18–44.

[41] A. Nayak, L. Pagli and N. Santoro. On testing for catastrophic faults in reconfigurable arrays with arbitrary link redundancy *Integration: the VLSI Journal* **16** (1996), 327–342.

[42] A. Nayak, N. Santoro and R. Tan. Fault-intolerance of riconfigurable systolic arrays. *20th Int. Symp. on Fault-Tolerant Computing*, 1990, 202–209.

[43] D. Peleg. Graph immunity against local influence. Technical Report CS96-11, Mathematics & Computer Science, Weizmann Institue of Science, 1996.

[44] D. Peleg. Local majority voting, small coalitions and controlling monopolies in graphs: A review. *Theoretical Computer Science* **282** (2002), 231–257.

[45] D. Peleg. Size bounds for dynamic monopolies *Discrete Applied Mathematics* **86** (1998), 263–273.

[46] D. Peleg, A. Wool. The availability of quorum systems. *Information and Computation* **123**(2) (1995), 210–223.

[47] S. Poljak, M. Sůra. On periodical behavior in societies with symmetric influences. *Combinatorica* **1** (1983), 119–121.

[48] S. Poljak, Turzik. On an application of convexity to discrete systems. *Discr. Appl. Math.* **13** (1986), 27–32.

[49] M. Raynal. *Algorithms for mutual exclusion*, MIT Press, 1986.

[50] N. Santoro. *Design and Analysis of Distributed Algorithms*, John Wiley, 2007.