On beta-skeleton automata with memory

Ramón Alonso-Sanz[1] and Andrew Adamatzky[2]

[1] ETSI Agronomos (Estadistica, GSC), C. Universitaria 28040, Madrid, Spain
`ramon.alonso@upm.es`
[2] University of the West of England, Bristol BS16 1QY, United Kingdom
`andrew.adamatzky@uwe.ac.uk`

Abstract

A β-skeleton is a proximity undirected graph whose connectivity is
determined by the parameter β. We study β-skeleton automata where
every node is a finite state machine taking two states, and updating its
states depending on the states of adjacent automata-nodes. We allow
automata-nodes to remember their previous states. In computational ex-
periments we study how memory affects the global space-time dynamics
on β-skeleton automata.

Keywords: Beta-skeletons; automata; memory

1   Introduction

In computational geometry and geometric graph theory, a β-skeleton or beta
skeleton is an undirected proximity graph defined from a set of points in the
Euclidean plane. Two points p and q are connected by an edge whenever their
β neighborhood is empty [24]. In lune-based β-skeletons, the β-neighborhood is
defined as: the intersection of two circles of radius $d(p,q)/2\beta$ that pass through
p and q, if $\beta \in [0,1]$; the intersection of two circles of radius $\beta d(p,q)/2$ centered
at the points $(1 - \beta/2)p + (\beta/2)q$ and $(\beta/2)p + (1 - \beta/2)q$, if $\beta \geq 1$.

Figure 1 shows three β-skeletons with increasing β parameter value, based
on the same ten nodes. The figure shows also the β-neighborhoods of the node
labeled 1 (upper-right). In the transition from β=0.9 to β=1.0, the 1-node loses
its 1-6 and 1-8 links, whereas from β=1.0 to β=1.5 it loses the 1-5 link.

β-Skeletons belongs to a family of proximity graphs. which are monotonously
parameterised by parameter the β. The structure of proximity graphs repre-
sents a wide range of natural systems and is applied in many fields of science.
Few examples include geographical variational analysis [21, 27, 33], evolutionary
biology [26], simulation of epidemics [36], study of percolation [19] and magnetic
field [35], design of ad hoc wireless networks [25, 34, 31, 28, 40]. Thus developing
and analysing computational models of spatially-extended systems on proxim-
ity graphs will shed a light onto basic mechanisms of activity propagation on
natural systems.

Automata on β-skeletons were originally introduced in [5] and studied in a
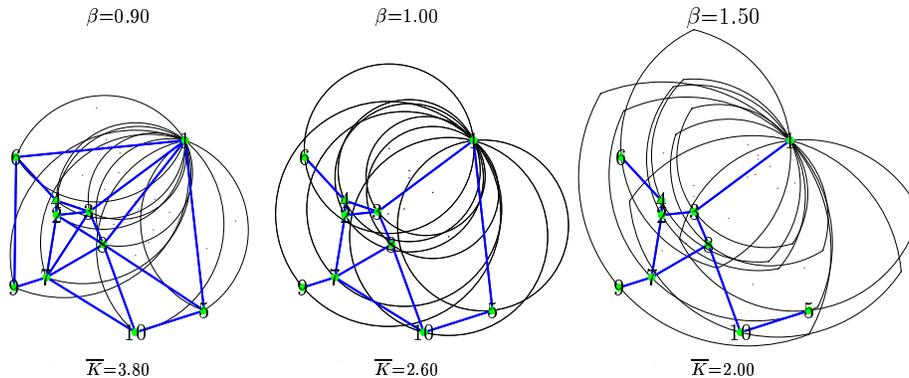context of excitation dynamics. In present paper we develop ideas of [5] along

Figure 1: Three β-skeletons on the same ten nodes.

lines of memory-enriched automata and global dynamics. The paper is structured as follows. In Sect. 2 we define automata on β-skeletons. Global dynamics on automata for $\beta \geq 1$ (planar graphs) are studied in Sects. 2.1 and 2.3, whereas the $\beta < 1$ (non-planar graphs) is studied in Sect. 2.2. Section 4 demonstrates the effects of weighted memory on the behaviour of automata networks. Automata networks with non-parity rules are briefly tackled in Sect. 5. Possible applications of our computational findings are discussed in Sect. 6.

2 Automata on beta-skeletons

In the automata on beta-skeletons studied here, each node is characterized by an internal state whose value belongs to a finite set. The updating of these states is made simultaneously (*à la* cellular automata) according to a common local transition rule involving only the neighborhood of each node [5]. Thus, if $\sigma_i^{(T)}$ is taken to denote the state value of node i at time step T, the site values evolve by iteration of the mapping : $\sigma_i^{(T+1)} = \phi\Big(\{\sigma_j^{(T)}\} \in \mathcal{N}_i\Big)$, where $\mathcal{N}_i$ is the set of nodes in the neighborhood of i and ϕ is an arbitrary function which specifies the automaton *rule*. This article deals with two possible state values at each site: $\sigma \in \{0, 1\}$, and the parity rule: $\sigma_i^{(T+1)} = \sum_{j \in \mathcal{N}_i} \sigma_j^{(T)} \mod 2$. Despite its formal simplicity, the parity rule may exhibit complex behaviour [23].

In the Markovian approach just outlined (referred as *ahistoric*), the transition function depends on the neighborhood configuration of the nodes only at the preceding time step. Explicit historic memory can be embedded in the dynamics by featuring every node by a mapping of its states in the previous time steps. Thus, what is here proposed is to maintain the transition function ϕ unaltered, but make it act on the nodes featured by a trait state obtained as a function of their previous states: $\sigma_i^{(T+1)} = \phi\Big(\{s_j^{(T)}\} \in \mathcal{N}_j\Big)$, $s_j^{(T)}$ being a state function of the series of states of the node j up to time-step T. We will consider

here the most frequent state (or *majority*) memory implementation. Thus, with unlimited trailing memory: $s_i^{(T)} = mode\left(\sigma_i^{(T)}, \sigma_i^{(T-1)}, \ldots, \sigma_i^{(1)}\right)$, whereas with memory of the last τ state values: $s_i^{(T)} = mode\left(\sigma_i^{(T)}, \sigma_i^{(T-1)}, \ldots, \sigma_i^{(\top)}\right)$, with $\top = \max(1, T - \tau + 1)$. In the case of equality in the number of time-steps that a node was 0 and 1, the last state is kept, in which case memory does not really actuate. This lack of effect of memory explains the lower effectiveness of even size τ-memories found in the results presented below.

A FORTRAN code working in double precision has been implemented to perform computations. The core of the code is given in Table 1. The wiring of the n nodes is encapsulated in a $n \times n$ adjacency matrix named ADJ, so that the `iadd` variable adds the states of the nodes linked to the generic node i. Thus, the parity rule is implemented by means of the `mod 2` operation on the `iadd` variable, generating `NEW(i)`, i.e., $\sigma_i^{(T+1)}$. Previously, the `MEMORY` subroutine implements the `itau`-majority memory computing. `MEMORY` provides the `MODE(i)` trait states, i.e., $s_i^{(T)}$, on which the `iadd` is calculated.

```
DO it=1,maxit
    call MEMORY(MODE,NEW,ISIG,n,it,maxit,itau)
    DO i=1,n;iadd=0
        do j=1,n;if(ADJ(i,j)==0)cycle
                iadd=iadd+MODE(j)
        enddo
        NEW(i)=mod(iadd,2)
    ENDDO
ENDDO

subroutine MEMORY(MODE,NEW,ISIG,n,it,maxit,itau)
    integer MODE(n),NEW(n),ISIG(n,maxit)
    ISIG(:,it)=NEW;MODE=NEW
    itin=max(1,it-itau+1);itaux=min(it,itau)
    do i=1,n; inn=sum(ISIG(i,itin:it))
            if(2*inn>itaux)MODE(i)=1
            if(2*inn<itaux)MODE(i)=0
    enddo
END
```

Table 1: The core of the FORTRAN progam.

2.1   The β=1 case.

Figure 2 shows the initial evolving patterns of a simulation of the parity rule on a β=1 skeleton (or Gabriel maps) with $N = 10^2$ nodes distributed at random in a unit square. Red squares denote node state values equal one, black squares denote zero state values. The effect of endowing nodes with memory of the *majority* of the last three states is shown at T=4. The encircled node exemplifies the initial effect of memory. Dynamics of perturbation spreading looks unpredictable and quasi-chaotic due to existence of long-range connections.

Figure 3 shows the evolution of the changing rate (the Hamming distance between two consecutive patterns) in eleven different β=1 simulations based in
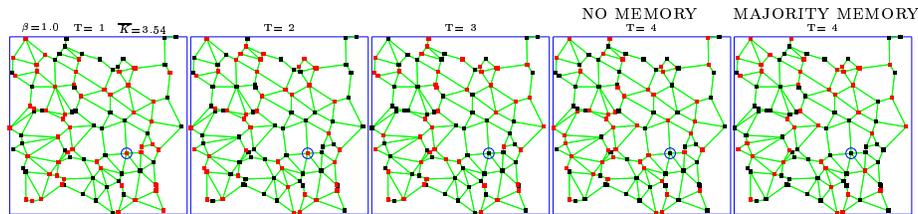
Figure 2: A simulation up to T=4 of the parity rule on a β=1 skeleton, $N = 100$. Red squares denote node state value equal one, black squares denote zero state values.

one thousand nodes distributed at random in a unit square. The red curves correspond to the ahistoric simulations, in which case the parity rule exhibits a very high level of changing rate, oscillating around 0.5. Figure 3 shows also the effect on the changing rate of endowing nodes with memory of the last τ state values (blue lines). The inertial effect of memory tends to reduce the changing rate compared to the ahistoric model, particularly when β is odd. With high memory charges, such as τ=19 in the lower left panel, the changing rate tends to vary in the long term in the $[0.1, 0.2]$ interval, after an initial *almost-oscillatory* behaviour which ceases by $T > \tau$+1=20. With unlimited trailing memory (lower right panel), this *oscillatory* pattern is never truncated, so that a rather unexpected quasi-oscillatory behaviour turns out with full memory. With no exception, the proportion of node states having one given state value (*density*), oscillates near to 0.5 regardless of the model considered.

Figure 4 shows the evolution of the damage rate, i.e., the relative Hamming distance between patterns resulting from reversing the initial state value of a single node, referred to as damage (or perturbation) spreading. The damage propagates very rapidly without memory (*butterfly effect*), so that by T=30 the red curves already oscillate around the fifty per cent of cells. When memory is introduced to the system, the spread of damage is depleted, albeit the restraining effect of memory is very low if the charge of memory is low (τ=3,4). With higher memory lengths, e.g., τ=9, the depletion in the advance of the damage becomes apparent, though by T=150 the damage rate also reaches the 0.5 level in every simulation in Fig. 4. Similar evolution of damage is found with higher limited trailing memories, such as τ=11 and τ=13 (not shown in the figure). On the contrary case, unlimited trailing memory appears very effective in the control of damage, as shown in the lower-right panel.

Figure 4 also shows the relative Hamming distance of the ahistoric patterns to the corresponding historic ones. After a very short transition period, this distance reaches a fairly permanent plateau level, oscillating around 0.5 regardless of the charge of memory endowed.
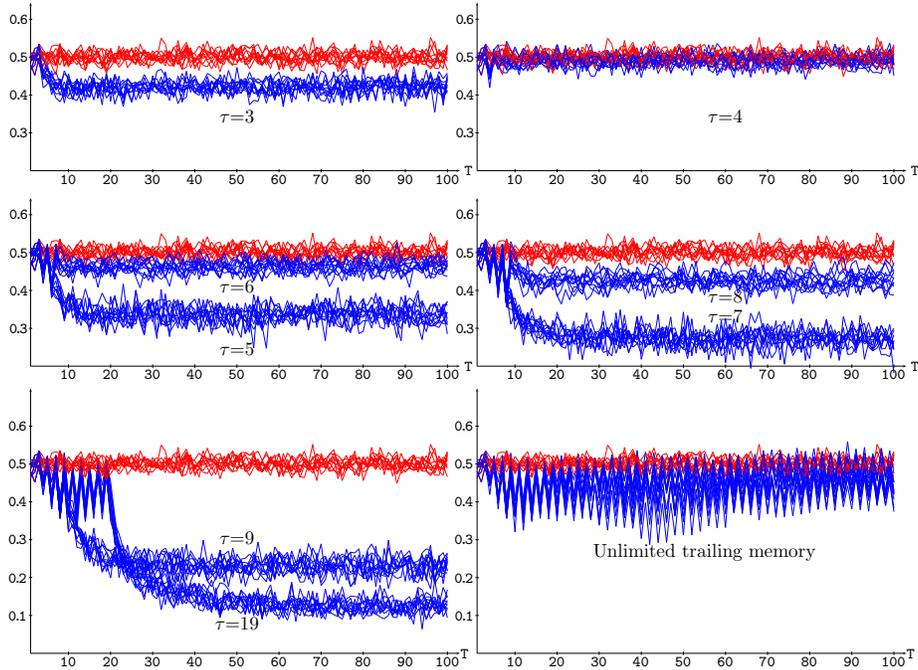
Figure 3: Changing rate in eleven simulations up to T=100 of the parity rule on β=1 skeletons, $N = 10^3$. Red plots correspond to the ahistoric simulations, blue plots correspond to τ-majority memory.
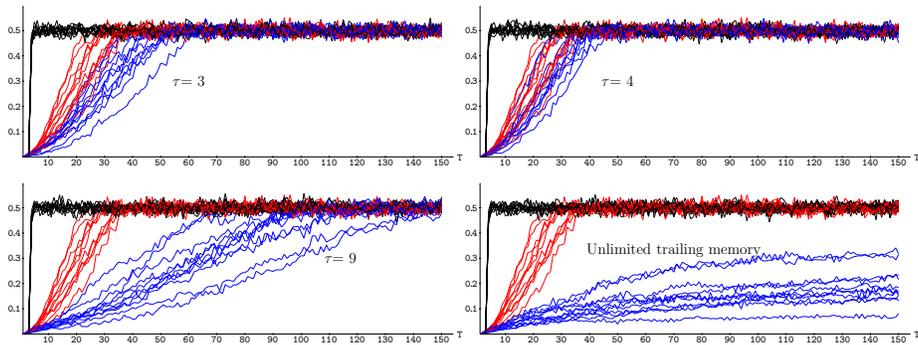

Figure 4: Damage spreading in the scenario of Fig. 3. Color code: black, distance to the ahistoric pattern; red, damage spreading with no memory; blue, damage spreading with memory.

2.2 A β <1 case.

Figure 5 shows a simulation up to T=4 of the parity rule on a β=0.9 skeleton based in the same nodes and initial states as in Fig. 2. In correspondence with

the lower β value, there are more links connecting nodes in the non-planar graph of Fig. 5 compared to those in Fig. 2 .
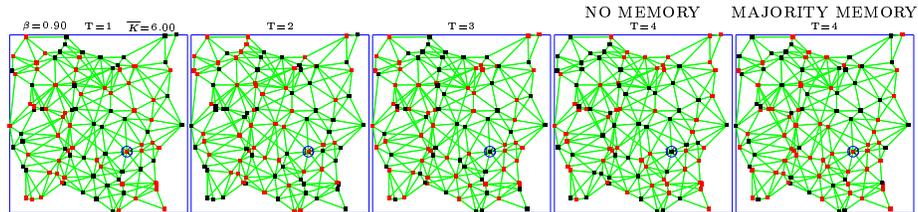

Figure 5: A simulation up to T=4 of the parity rule on a β=0.9 skeleton. The location of nodes and their initial states are those of Fig. 2

Figure 6 shows the evolution of the changing rate in eleven different β=0.9 simulations. based in the same nodes and initial states as in Fig. 3 . In this scenario, with high connectivity, low memory charges, e.g., τ=3, 4 , have not an apparent effect on the changing rate, so that there are not presented in the figure. Memory charges of τ=5 and τ=7 have a limited effect, whereas τ=9, 19 already have an apparent effect. With unlimited trailing memory, the *oscillatory* patterns get a notable amplitude, albeit below 0.5 .
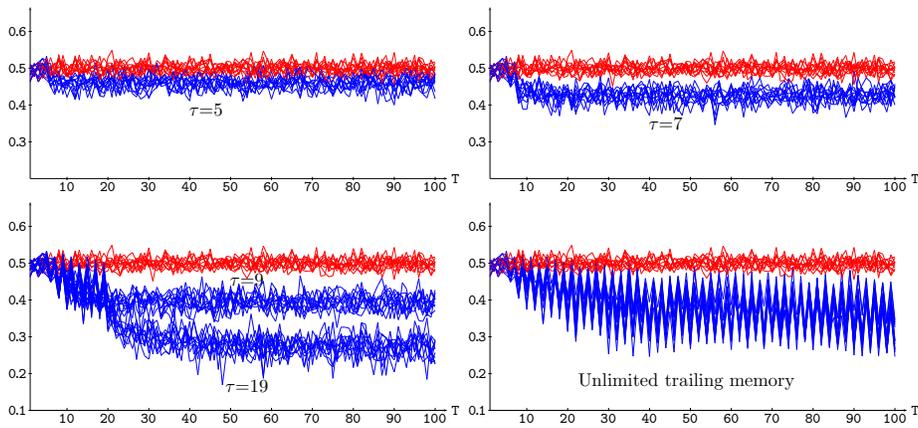

Figure 6: Changing rate in eleven simulations up to T=100 of the parity rule on β=0.9 skeletons. N=1000 . Color code as in Fig. 3 .

Figure 7 shows the damage spreading in the scenario of Fig. 6 . In this highly connected scenario, memory turns out rather ineffective in the control of damage. Even with unlimited trailing memory, the damage grows up to 0.5 fairly soon in most simulations, or is just delayed up to approximately T=100 in some others.
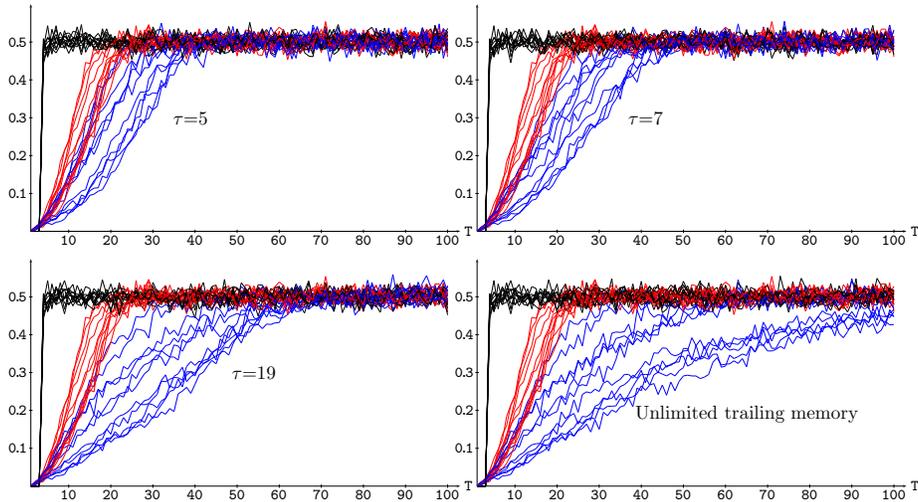
6

Figure 7: Damage spreading in the scenario of Fig. 6 . Color code as in Fig. 4 .

2.3 A $\beta > 1$ case.

Figure 8 shows a simulation up to $T=4$ of the parity rule on a lune-based $\beta=2$ skeleton (also termed relative neighborhood graphs) based in the same nodes and initial states as in Fig. 2 . In correspondence with the higher β value, there are less links connecting nodes in Fig. 8 compared to those in Fig. 2 .
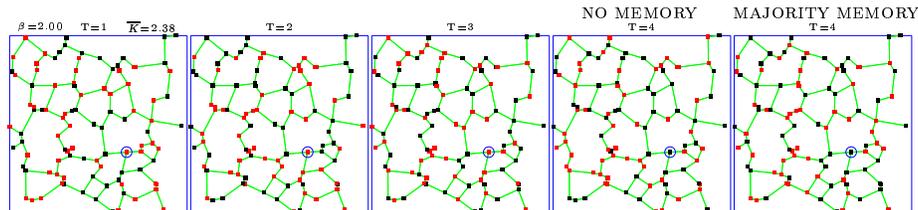

Figure 8: A simulation up to $T=4$ of the parity rule on a $\beta=2$ skeleton, $N = 10^2$. The location of nodes and their initial states are those of Fig. 2 .

Figure 9 shows the evolution of the changing rate in eleven different $\beta=2$ simulations based in the same nodes and initial states as in Fig. 3 . In this scenario, even low memory charges, e.g., $\tau=3, 4$, have an apparent effect on the changing rate, and higher ones, e.g., $\tau=9, 19$, led this parameter to very low values. Thus for example with $\tau=19$ the changing rate varies in the [0,0.1] interval. With unlimited trailing memory (lower right panel), the non-truncated *oscillatory* patterns soon reach a very high level near 0.5 .

Figure 10 shows the latest snapshots of the changing nodes in one of the $\beta=2$ skeleton simulations in Fig. 9 . Both in the ahistoric context, where ap-
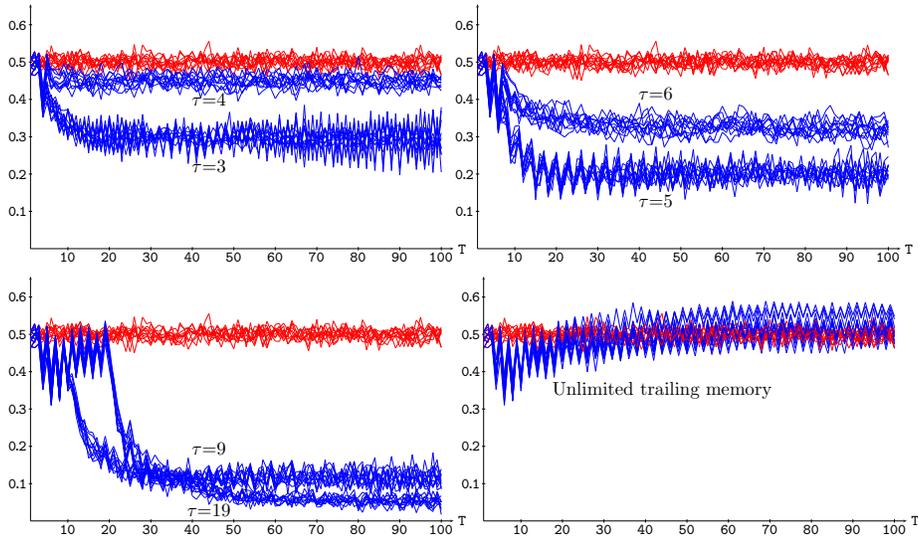
Figure 9: Changing rate in eleven simulations up to $T=100$ of the parity rule on $\beta=2$ skeletons, $N = 10^3$. Color code as in Fig. 3.

proximately half of the nodes change, and with $\tau=19$ majority memory, in which case only small sets of nodes are changing.
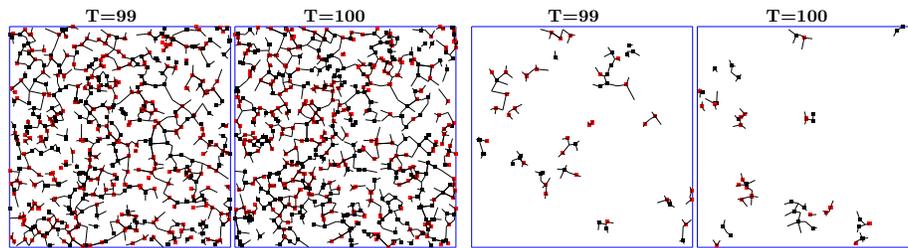

Figure 10: Latest snapshots of the changing nodes in one of the $\beta=2$ skeleton simulations in Fig. 9. Color code: Red: $0{\rightarrow}1$, Black: $1{\rightarrow}0$. Ahistoric patterns (left), and patterns with $\tau=19$ majority memory (right).

Figure 11 shows the damage spreading in the scenario of Fig. 9. In this scenario, with low connectivity, the depletion in the advance of the damage becomes apparent already with $\tau=3$ though by $T=100$ the damage rate also reaches the 0.5 level. As expected, $\tau=9$ is much more effective in the damage control, and, above all, unlimited trailing memory which virtually impedes the advance of damage.
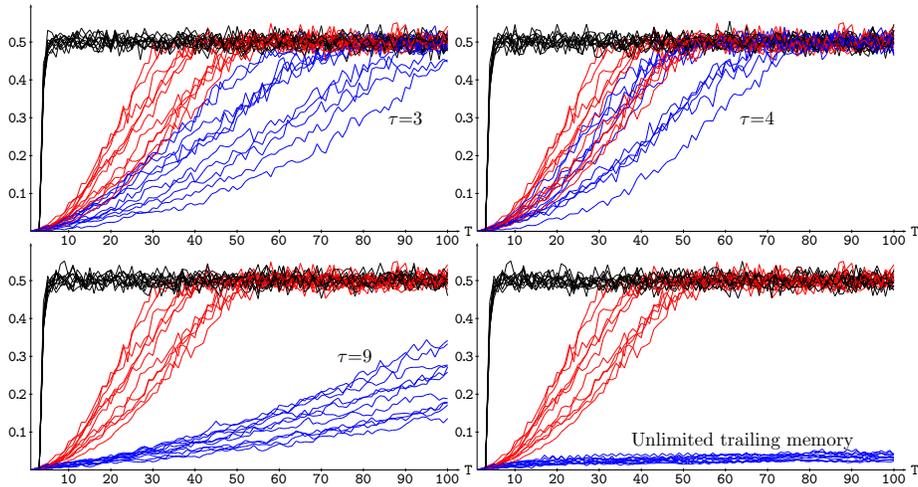
8

Figure 11: Damage spreading in the scenario of Fig. 9. Color code as in Fig. 4.

3 Asymptotic levels.

Figure 12 shows the asymptotic changing rate (left) and damage spreading (right) with increasing values of the length of memory τ in the three β-scenarios considered before. *Asymptotic* standing for the mean of then ten last values in both cases. The scenario is that of previous sections, i.e., a $N{=}1000$ network run up to $T = 100$ iterations, so that $\tau{=}100$ implies full memory. The changing rate notably declines as a result of the initial increase of the length of memory (particularly with $\beta{=}0.9$). The lowest changing rate levels are achieved around $\tau \simeq T/2{=}50$. These levels are not very much altered until a very high length of memory is reached. Namely, by $\tau \gtrsim 9T/10{=}90$, the oscillation in the changing rate is maintained most of the time-steps, so that the subsequent decay becomes increasingly difficult.
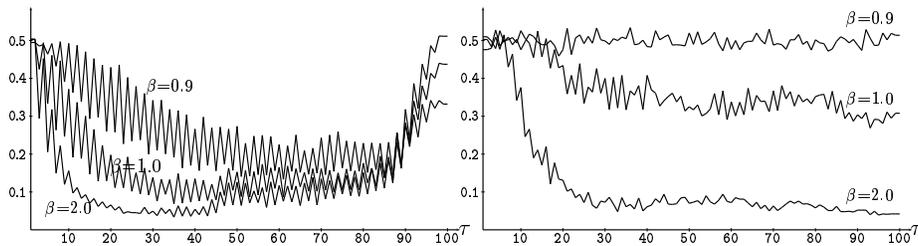

Figure 12: Asymptotic changing rate (left) and damage spreading (right) in $N{=}1000$ parity β-skeletons with τ-majority memory.

The effect of increasing the memory length on the damage spreading (Fig. 12, right), is much more monotone: a seemingly plateau, below the 0.5 level, is

reached in the β=1.0 and β=2.0 cases, whereas in the highly connected β=0.9 context, memory does not qualitatively alter the maximum 0.5 level.

The effect of memory in bigger networks, does not qualitatively differ from that presented here for N=1000. The dynamics of both the changing rate and damage spreading behave in broad strokes as in the N=1000 case. Maybe in a smoother way in every single simulation, and closer when considering different simulations. But smaller networks are more problematic. Thus, Fig.13 shows the asymptotic changing rate (left) and damage spreading (right) in the same conditions of Fig. 12, except in what refers to the network size : $N = 100$ instead of $N = 1000$. The general pattern of the asymptotic changing rate frame in Fig. 13 resembles that of Fig. 12, albeit the higher variability of the changing rate in the small network is hidden by the mere presentation of the average of the last values. The structure of the damage spreading frame changes dramatically compared to that of the bigger network : memory is unable to avoid the full damage spread across small networks. Neither in the low connected scenarios, not even in that with β=2.0 .
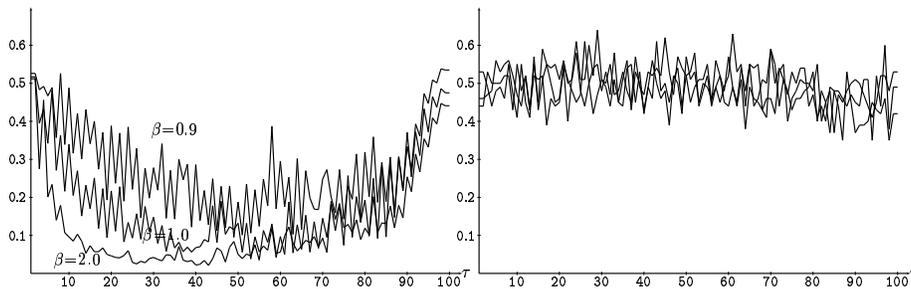

Figure 13: Asymptotic changing rate (left) and damage spreading (right) in N=100 parity β-skeletons with τ-majority memory.

4 Weighted Memory.

Majority memory of the last τ time-steps demands τ extra bits per node to store their corresponding state values. To avoid this drawback, past state values can be weighted in such a way that only the accumulated memory charge needs to be stored. Thus, for example, historic memory can be weighted by applying a geometric discounting process in which the state $\sigma_i^{(T-t)}$, obtained t time steps before the last round, is actualized to $\alpha^t \sigma_i^{(T-t)}$, α being the *memory factor* lying in the [0,1] interval. This well known mechanism fully takes into account the last round ($\alpha^0 = 1$), and tends to *forget* the older rounds.

Every node will be featured by the rounded weighted mean of all its past states, so the memory mechanism is implemented in two steps at time-step T :

(i) The unrounded weighted mean (m) of the states of every node is computed first :

$$m_i^{(T)}(\sigma_i^{(1)}, \ldots, \sigma_i^{(T)}) = \frac{\sigma_i^{(T)} + \sum_{t=1}^{T-1} \alpha^{T-t} \sigma_i^{(t)}}{1 + \sum_{t=1}^{T-1} \alpha^{T-t}} \equiv \frac{\omega_i^{(T)}}{\Omega(T)}$$

(ii) Then the trait state s is obtained by comparing the memory *charge m* to the landmark 0.5 (if $\sigma \in \{0, 1\}$), assigning the last state in case of an equality to this value, so that:

$$s_i^{(T)} = \mathcal{H}(m_i^{(T)}) = \begin{cases} 1 & if & m_i^{(T)} > 0.5 \\ \sigma_i^{(T)} & if & m_i^{(T)} = 0.5 \\ 0 & if & m_i^{(T)} < 0.5 \end{cases}.$$

The choice of the memory factor α simulates the long-term or remnant memory effect: the limit case $\alpha = 1$ corresponds to a memory with equally weighted records (*full* memory, equivalent to unlimited trailing *majority* memory), whereas $\alpha \ll 1$ intensifies the contribution of the most recent states and diminishes the contribution of the more remote states (short-term working memory). The choice $\alpha = 0$ leads to the ahistoric model. This memory implementation will be referred to as α-memory.

This geometric memory mechanism is not *holistic* but *cumulative* in its demand for knowledge of past history: the whole $\{\sigma_i^{(t)}\}$ series needs not be known to calculate the term $\omega_i^{(T)}$ of the memory *charge* $m_i^{(T)}$, while to (sequentially) calculate $\omega_i^{(T)}$ one can resort to the already calculated $\omega_i^{(T-1)}$ and compute: $\omega_i^{(T)} = \alpha \omega_i^{(T-1)} + \sigma_i^{(T)}$. Consequently, only one number per node (ω_i) needs to be stored. This positive property is accompanied by the drawback of any weighted average memory: it computes with real numbers instead of with integers as is done with majority memory.

In the most unbalanced scenario, $\sigma_i^{(1)} = \ldots = \sigma_i^{(T-1)} \neq \sigma_i^{(T)}$, it holds that [1]: $m = 0.5 \Rightarrow \alpha_T^T - 2\alpha_T + 1 = 0$ [4], where α_T holds for the critical value of α below which memory has no effect up to time-step T. When $T \to \infty$, the equation [4] becomes: $-2\alpha_\infty + 1 = 0$, thus, α-memory is not effective if $\alpha \leq 0.5$.

Figure 14 shows the changing rate with α-memory in parity β-skeletons of one thousand nodes, starting from a single active node. Without memory, the changing rate progresses very fast regardless of the β value. On the contrary, full memory, i.e., $\alpha =1.0$, dramatically restrains the changing rate. In the $\beta=2.0$ scenario, thus with low connectivity, even the smallest memory charges, e.g., $\alpha=0.6$, lead the changing rate to extinction. In the more connected networks,

---

[1]

$$\begin{cases} m(0, 0, \ldots, 0, 1) = \frac{1}{2} & \equiv & 1 & = \frac{1}{2}\frac{\alpha^T - 1}{\alpha - 1} \\ m(1, 1, \ldots, 1, 0) = \frac{1}{2} & \equiv & \frac{\alpha^T - 1}{\alpha - 1} = \frac{1}{2}\frac{\alpha^T - 1}{\alpha - 1} \end{cases} \Rightarrow \alpha_T^T - 2\alpha_T + 1 = 0$$

low memories such as α=0.6 and α=0.7 only delay the increase of the changing rate, that reaches 0.5 in the figure in both the β=1.0 and β=0.9 scenarios. With α=0.8 memory, the 0.5 level is not reached up to T=100 in the β=1.0 simulation, but it is reached in the β=0.9 simulation. Finally, with α=0.9 the changing rate is notably restrained in both scenarios, though it seems not stabilized as it is with full memory. Thus, in the simple context of Fig. 14 full memory exerts a, let us say, expected effect, that of maximum dynamic inhibition. The dynamic starting from a single active node may be considered as the simplest form of damage spreading, measured by the evolution of density. The density curves (not shown) mimic those of the changing rate shown in Fig. 14, including those with full memory, agreeing with the expected effect on damage spreading already shown in the majority memory scenario.
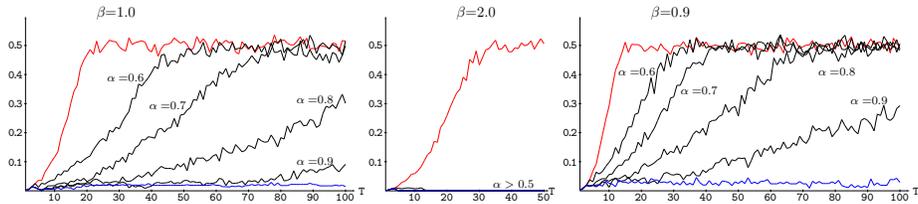

Figure 14: Changing rate with α-memory in parity β-skeletons from a single active node. N=1000. Red plots correspond to the ahistoric simulations, blue plots correspond to full majority memory.

Figure 15 shows the asymptotic changing rate (left) and damage spreading (right) in N=1000 parity β-skeletons with α-memory. The effect of α-memory is much more monotone than that of majority memory shown in Figure 12. Thus, as a rule, increasing the memory factor implies restrain in both the changing rate and the damage spreading. The changing rate in Fig. 15 sharply peaks in the full memory (α=1.0) scenario, according to the the full lengt (τ=100) majority memory changing rates shown in Fig. 12. It turns out remarkable that damage spreading is restrained only when α reaches high values: $\gtrsim$ 0.90 in the medium connected β=1.0 scenario, and $\gtrsim$ 0.85 in the low connected β=2.0 scenario.
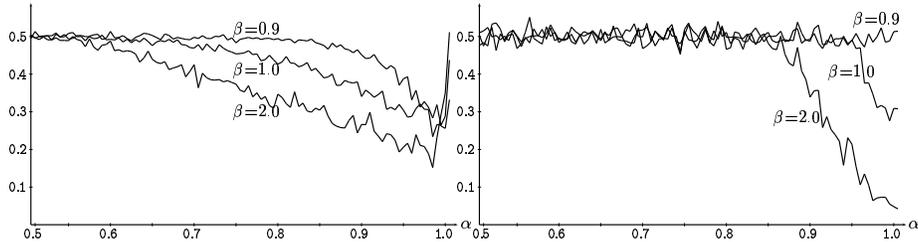

Figure 15: Asymptotic changing rate (left) and damage spreading (right) in N=1000 parity β-skeletons with α-memory.

12

5 Other rules, other contexts

The results presented up to now deal with the *parity* rule. This rule is a
particularly 'active' rule that produces very high changing rates. Neverthe-
less, most other totalistic rules, i.e., based in the sum of the neighbour states
$\sum_i = \sum_{j \in \mathcal{N}_i} \sigma_j^{(T)}$, tend to led the dynamic to low stable levels of changing rate.
This feature is particularly visible in *majority* rule: $\sigma_i^{(T+1)}{=}1$ if $2\sum_i > K_i$,
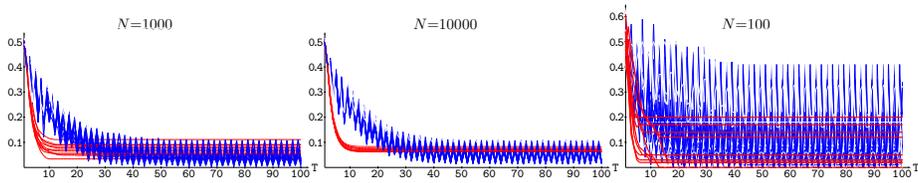$\sigma_i^{(T+1)}{=}0$, if $2\sum_i > K_i$ where K_i standing for the connectivity of node i .

Figure 16: Changing rate with the ahistoric majority rule (red), and the ma-
jority rule with nodes endowed memory of the parity of the last three states
(blue). β =1-skeletons with 10^3 nodes (left), 10^4 nodes (center), and 10^2 nodes
(right).

In the simulations presented in Fig. 16, the generic node i preserves its
last state in case of a tie: $\sigma_i^{(T+1)}{=}\sigma_i^{(T)}$ if $2\sum_i = K_i$. The dynamics is very
much sensitive at this respect, so that adopting $\sigma_i^{(T+1)}{=}1$ if $2\sum_i = K_i$ will
lead the density to 1.0, whereas with $\sigma_i^{(T+1)}{=}0$ if $2\sum_i = K_i$ the density will
tend to plummet. With the *neutral* criterion adopted here, there is no reason
for density drift to any of the extreme values, so that the red lines in Fig. 16
become stabilized after a short transition period.

In this scenario, the activity may be supported implementing memory rules
not of majority type, such as the parity rule acting as memory [18]. Thus for
example, the parity of the last three states [2], as implemented in Fig. 16. The
left panel of this figure applies to simulations with $N = 10^3$ nodes (usually for
present paper). Simulations with higher number of nodes, e.g., $N = 10^4$ in
the central panel, do not qualitatively differ from those with $N = 10^3$. They
present the same general features in their dynamics, maybe with an additional
characteristic: the evolution of density, changing rate and damage spreading
is much more similar among them. Opposite, simulations with few nodes, e.g.,
one hundred nodes in the right panel, are much more unforeseeable. Not only in
what respect to the effect of memory, even the ahistoric dynamics may converge
to very different densities and changing rates. Thus, for example, in the $N =
10^2$ simulations of the far right panel of Fig. 16, some ahistoric dynamics are
led to very low densities of $\sigma{=}1$ nodes, e.g., $\rho{=}0.2$, whereas other ones reach

---

[2] $s_i^{(1)} = \sigma_i^{(1)}$, $s_i^{(2)} = \sigma_i^{(1)} \oplus \sigma_i^{(2)}$, $s_i^{(T)} = \sigma_i^{(T-2)} \oplus \sigma_i^{(T-1)} \oplus \sigma_i^{(T)}$.

high densities, e.g., ρ=0.8, very distant from the ρ=0.5 obtained in ahistoric simulations with $N = 10^3$ or $N = 10^4$ nodes.

We have studied the effect of memory in several cellular automata and Boolean network scenarios [13, 14, 15]. The mean connectivity across the eleven simulations run here is $\overline{K}$=3.85, $\overline{K}$=2.50, and $\overline{K}$=6.64, for β=1.0, β=2.0, and β=0.9 respectively. The dynamics studied in this article may be somehow compared to those simulations in the referenced articles with mean connectivities approaching these values. Thus, for example, the way in which a single active node propagates activity (a kind of damage spreading), as shown in the patterns at T=100 in Fig. 17: no restrain in the ahistoric simulation, some dispersion with α=0.9, and activity concentrated near the initial active node with full memory.
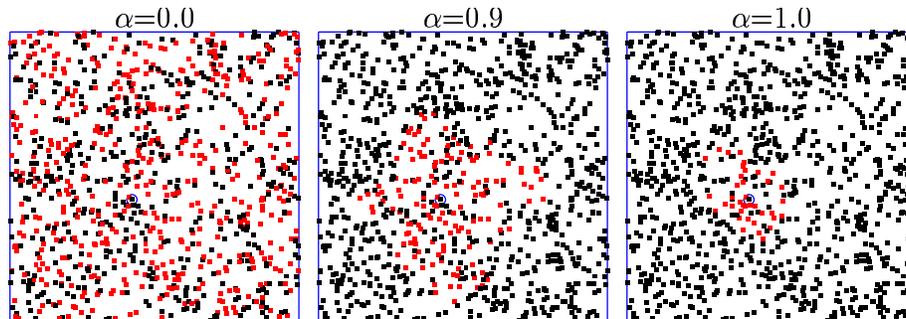

Figure 17: Patterns at T=100 with α-memory in the parity β=1-skeleton starting from a single active node considered in Fig. 14.

In previous studies we have analyzed the effect of memory in some spatialized games. Thus, the battle of the sexes [10] and the prisoner's dilemma (PD) [16, 17]. It has been concluded in those studies that the consideration of previous choices and payoffs in the updating rule, notably boosts cooperation. Although the effect of memory on games with players located in β-skeleton nodes is planned for further work, let us point here that the just outlined conclusion, reached in other spatial contexts, may likely also be found when dealing with the PD β-skeletons. Thus for example, starting with a single defector in a sea of cooperators, memory will restrain the defection progress, much in the way of the restrain of the advance of the active nodes in Fig. 17. Memory will likely have even a stronger effect on the PD dynamics in β-skeletons with low connectivities, i.e., large β. On the contrary, in highly connected β-skeletons, i.e., small β, it is likely to be found that memory will be unable to avoid the spread of defection propelled via action at distance.

6   Potential applications

What are potential outcomes of our findings?. Proximity graphs, particularly relative neighbourhood graphs (β-skeletons for $\beta = 2$) are invaluable in simulation of human-made road networks, see for example the study of Tsukuba central district [41, 42]. The graphs provide a good formal representation of biological transport networks, particularly foraging trails of ants [1] and protoplasmic networks of slime mold *Physarum polycephalum* [3, 4]. Our computational approaches could be applied toward studies of damage propagation in artificial and natural networks, and also used to develop novel techniques for controlling space-time dynamics on the networks.

Majority of the results in the paper deal with parity rule. This rule is important because parity rule cellular automata are considered to be low level models of self-replication, see e.g. [32, 23], and also, similar rules are used for fast simulation of soliton cellular automata, see e.g. [30]. Ultimately the parity rule can be considered as a good discrete approximation of propagation of perturbations in *active* non-linear discrete media, where a localised traveling perturbation (analog of excitation in excitable media) generates other perturbations. Thus, results of the paper could be, in principle, applied to studying of memory-based control of spatio-temporal dynamics of non-linear active media.

With regards to smile mold, indeed its protoplasmic network is always changing, new tubes are formed, some old tubes become abandoned. Even in this case — 'a la structure dans le flux' — β-skeleton is an excellent approximation of protoplasmic networks for over 80% of plasmodium foraging span [8]. The parity simulated in the paper generates non-trivial patterns of activity, which can be interpreted in terms of oscillatorty dynamics of electrical potential or contractility of the plasmodium of *Physarum polycephalum* [39]. By tuning capacity of a node memory of β-skeleton we could shape periodical activity of the graph and thus imitate interactions between electrical and oscillatory activity of the slime mould, and also incorporate heterogeneous network of discrete bio-chemical oscillators in the discrete models of the plasmodium [7].

Yet another, rather more specific domain where our results on β-skeleton with memory can be applied is a computation in assembles of lipid vesicles filled with Belousov-Zhabotinsky (BZ) mixture [22, 29]. When implementing computation in excitable chemical systems [2] we usually encounter an a troublesome problem of wave-fragments' instability. A traveling wave-fragment (which represents a value of logical variable) rarely preserves its shape for a long time, it rather collapses or expands. This problem can be overcome by subdividing computing substrate into interconnected compartments — BZ-vesicles — and allowing waves to collide one with another only inside the compartments, see details in [9, 20]. An ensemble of BZ-vesicles forms a microscopic analog of a massively-parallel processor. In computer simulations [9, 20] BZ-vesicles are arranged in regular arrays. It is very unlikely that in natural conditions ensembles of BZ-vesicles will be regular. When BZ-vesicles, quite possible of different sizes, are aggregated into an assemble and tightly packed, they are represented by Delaunay triangulation. A computation on Delaunay triangulation is implemented

in the same manner as on a regular cellular automata [6]. However, when some
BZ-vesicles start to degrade, and other increase in size average connectivity of
the vesicle conglomerate decreases. What kind of proximity graph represents
then the ensemble of BZ-vesicles?

In 1980 Toussaint demonstrated that MST⊆RNG⊆DT [37]. This hierarchy
of graphs was later enriched with GG [27] as follows:

$$\text{NNG} \subseteq \text{MST} \subseteq \text{RNG} \subseteq \text{GG} \subseteq \text{DT}$$

That is when BZ-vesicles bursting the graph representation of the vesicles'
ensemble will travel downwards in the Toussaint hierarchy: the diagram becomes
a Gabriel graph GG (β-skeleton for $\beta = 1$) then relative neighbourhood graph
RNG (β-skeleton for $\beta = 2$) then spanning tree MST and then disconnected
nearest-neighbour graph. Response of each BZ-vesicle to the excitation in the
neighbouring vesicles depends on the diffusion speed, size of pores between the
vesicles, concentration of reactants inside the vesicle and overall age of the
system. Usually, vesicle's reaction in any particular time step is determined
not only by current dynamics of excitation in the vesicle's neighbourhood but
also by history of chemical reactions in the vesicle itself. Thus, in computational
experiments with β-skeletons with memory we discover, at least at the level of
discrete yet fine-grained models, how activity can propagate in a ensemble of
'ageing' BZ-vesicles and what methods of the activity control can be employed.

7   Conclusion

As a rule, endowing nodes in beta-skeleton parity automata with majority mem-
ory induces a moderation in the rate of changing nodes and in the damage
spreading, albeit in the latter case memory turns out ineffective in the control
of the damage with low memory charge and in networks with high connectivity,
i.e., with low β. In a simulation up to T time-steps, the maximum depleting
effect of memory on the changing rate is achived with a memory length $\tau \simeq T/2$.
With very high memory length, the oscillatory-like effect induced by memory
in the changing rate is maintained too long to allow a effective depletion in the
changing rate. Thats is so even in low-connected networks, i.e., with large β.

Acknowledgment

References

[1] Adamatzky A. and Holland O.(2002) Reaction-diffusion and ant-based load balancing of communication networks. *Kybernetes*, 31, 667–681.

[2] Adamatzky A., De Lacy Costello B., Asai T. Reaction Diffusion Computers. Elsevier, 2005.

[3] Adamatzky,A.(2008). Developing proximity graphs by Physarum Polycephalum: Does the plasmodium follow Toussaint hierarchy?. *Parallel Processing Letters*, 19, 105–127.

[4] Adamatzky A. and Jones J.(2009). Road planning with slime mould: If Physarum built motorways it would route M6/M74 through Newcastle. *Int J Bifurcation and Chaos*, (in press). See also `arXiv:0912.3967v1`.

[5] Adamatzky,A.(2010). On excitable beta-skeletons. *Journal of Computational Science*, 1, 175–186.

[6] Adamatzky A. On excitable Delaunay triangulations. Kybernetes (2010), in press.

[7] Adamatzky A. Physarum machines: encapsulating reaction-diffusion to compute spanning tree. Naturwisseschaften 94 (2007) 975–980.

[8] Adamatzky A. Physarum Machines. World Scientific, 2010.

[9] Adamatzky A., Holley J., Bull L., De Lacy Costello B. On computing in fine-grained compartmentalised Belousov-Zhabotinsky medium (2010). arXiv:1006.1900v1 [nlin.PS]

[10] Alonso-Sanz,R.(2011). Self-organization in the battle of the sexes. *Int. J. Modern Phyisics C*, (in press).

[11] Alonso-Sanz,R.(2009). Spatial order prevails over memory in boosting cooperation in the iterated prisoner's dilemma. *Chaos*, 19,2, 023102.

[12] Alonso-sanz,R.(2009). Memory versus spatial order in the support of cooperation. *Biosystems*, 97,90-102.

[13] Alonso-Sanz,R.,Bull,L.(2008). Boolean networks with memory. *Int. J. Bifurcation and Chaos*, 18,2,3799-3814.

[14] Alonso-Sanz,R. & Cárdenas,J.P.(2007). On the effect of memory in disordered Boolean Networks: the $K = 4$ case, *Int.J. Modern Physics C*, 18,8,1313-1327 .

[15] Alonso-Sanz,R. & Cárdenas,J.P.(2007). The effect of Memory in Cellular Automata on scale-free Networks: the $\overline{K}$=4 case. *Int.J. Bifurcation & Chaos.*, in press

[16] Alonso-Sanz,R.(2009). Spatial order prevails over memory in boosting cooperation in the iterated prisoner's dilemma. *Chaos*, 19,2, 023102.

[17] Alonso-sanz,R.(2009). Memory versus spatial order in the support of cooperation. *Biosystems*, 97,90-102.

[18] Alonso-Sanz,R.,Martin,M.(2006). Elementary cellular automata with elementary memory rules in cells : the case of linear rules. *J. of Cellular Automata*, 1,1, 70-86.

[19] Billiot J. M., Corset F., Fontenas E. (2010). Continuum percolation in the relative neighborhood graph. `arXiv:1004.5292`

[20] Holley J., Adamatzky A., Bull L., De Lacy Costello B., Jahan I. Computational modalities of Belousov-Zhabotinsky encapsulated vesicles. (2010). arXiv:1009.2044v1 [nlin.CG]

[21] Gabriel K. R. and Sokal R. R. (1969). A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18,259,259-270.

[22] Gorecki J. Private communication (2010).

[23] Juliam,P.,Chua,L.O.(2002). Replication properties of parity cellular automata. *Int. J. Bifurcation and Chaos*, 12,3,479-494.

[24] Kirkpatrick,D.G,Radke,J.D. A framework for computational morphology. In G. Toussaint (ed.), Computational geometry, 217,248. North-Holland, 1985.

[25] Li X.-Y. (2004). Application of computation geometry in wireless networks. In: Cheng X., Huang X., Du D.-Z. (Eds.) Ad Hoc Wireless Networking (Kluwer Academic Publishers), 197–264.

[26] Magwene P. W. (2008). Using correlation proximity graphs to study phenotypic integration. *Evolutionary Biology*, 35, 191–198.

[27] Matula D. W. and Sokal R. R. (1980). Properties of Gabriel graphs relevant to geographic variation research and clustering of points in the plane. *Geogr. Anal.*, 12, 205-222.

[28] Muhammad R. B.A. (2007). distributed graph algorithm for geometric routing in ad hoc wireless networks. *J Networks 2*, 49–57.

[29] NeuNeu: Artificial Wet Neuronal Networks from Compartmentalised Excitable Chemical Media. (2010) `http://neu-n.eu/`

[30] Park J. K., Steiglitz K., Thurston W. P. Soliton-like behavior in automata. Physica D 19 (1986) 423–432.

[31] Santi P. (2005). Topology Control in Wireless Ad Hoc and Sensor Networks (Wiley).

[32] Sipper M. Fifty years of research on self-replication: An overview. Artificial
Life. 4 (1998) 237–257.

[33] Sokal R. R. and Oden N. L. (2008). Spatial autocorrelation in biology 1.
Methodology. *Biological Journal of the Linnean Society*, 10,199–228.

[34] Song W.-Z., Wang Y., Li X.-Y. (2004). Localized algorithms for energy
efficient topology in wireless ad hoc networks. In: *Proc. MobiHoc*, (May 24-
26, 2004, Roppongi, Japan).

[35] Sridharan M. and Ramasamy A. M. S. (2010). Gabriel graph of geomagnetic
Sq variations. *Acta Geophysica*, `10.2478/s11600-010-0004-y`

[36] Toroczkai Z. and Guclu H. (2007). Proximity networks and epidemics. *Physica A*, 378, 68. `arXiv:physics/0701255v1`

[37] Toussaint G. T., The relative neighbourhood graph of a finite planar set,
Pattern Recognition 12 (1980) 261–268.

[38] Toussaint G. T., Some unsolved problems on proximity graphs, 1st Workshop on Proximity Graphs, Las Cruces, New Mexico, December, 1989.
`http://cgm.cs.mcgill.ca/~godfried/publications/openprox.pdf`

[39] Tsuda S., Jones J. The emergence of synchronization behavior in *Physarum
polycephalum* and its particle approximation. Biosystems (2010), in press.

[40] Wan P.-J., Yi C.-W. (2007). On the longest edge of Gabriel Graphs in
wireless ad hoc networks. *IEEE Trans. on Parallel and Distributed Systems*,
18, 111–125.

[41] Watanabe D. (2005). A study on analyzing the road network pattern using
proximity graphs. *J. of the City Planning Institute of Japan*, 40,133–138.

[42] Watanabe D. (2008). Evaluating the configuration and the travel efficiency
on proximity graphs as transportation networks. *Forma*, 23,81-87.