Nano Communication Networks 4 (2013) 216–228

Contents lists available at ScienceDirect

Nano Communication Networks

journal homepage: www.elsevier.com/locate/nanocomnet

On Delaunay triangulation automata with memory

Ramón Alonso-Sanz [a], Andrew Adamatzky [b,*]

[a] Universidad Politecnica de Madrid, ETSI Agronomos (Estadistica, GSC), C. Universitaria 28040, Madrid, Spain
[b] University of the West of England, Bristol BS16 1QY, United Kingdom

ARTICLE INFO

Article history:
Received 21 January 2013
Received in revised form 23 June 2013
Accepted 8 October 2013
Available online 21 October 2013

Keywords:
Delaunay triangulation
Automata
Memory

ABSTRACT

This study focuses on the analysis of the dynamics of parity automata on Delaunay triangulations (DTs): a kind of relatively highly connected proximity graphs endowed with a very active rule. The inertial effect of memory of past states is fully considered in this context, whereas other related contexts, such as cellular DT automata, are briefly tackled.

1. Introduction

The Delaunay triangulation (DT), together with its dual graph Voronoi diagram, are famous graphs, with hundreds of application in science and engineering. A Delaunay triangulation of a planar set is a triangulation of the set such that a circumcircle of any triangle does not contain a point of the set. In 1980 Toussaint demonstrated that MST $\subseteq$ RNG $\subseteq$ DT [1]. This hierarchy of proximity graphs [2] was later enriched with GG [3] as follows: NNG $\subseteq$ MST $\subseteq$ RNG $\subseteq$ GG $\subseteq$ DT, where NNG is a nearest neighborhood graph, MST is a minimum spanning tree, RNG is a relative neighborhood graph, and GG is a Gabriel graph [4]. Ascending in the hierarchy from NNG to DT is accompanied by an increase of connectivity. Thus we can consider DT as an ultimate species of proximity graphs with highest connectivity.

DTs are widely applied in nano-sciences. Main application domains are clustering of experimental data, numerical analysis and modeling [5,6], as well as applications in atom probe tomography [7]; and, representation of nano-networks by the triangulation. The latter includes nano-structure development and self-organization [8], geometry of nano-networks [9], and study of topological structure of nano-porous anodic aluminum oxide [10].

A DT is an instance of a highly connected proximity graph. Proximity graphs, particularly relative neighborhood graphs, are invaluable in simulation transport and communication networks, both man made and biological [11–15]. Thus they 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.

DTs are formal models of Belousov–Zhabotinsky (BZ) vesicle networks. Regular, or irregular but manually designed, arrangements of vesicles filled with excitable chemical Belousov–Zhabotinsky reaction mixtures bear huge computational potential [16–19]. Usually lipid vesicles filled with BZ mixture are of different sizes, they do not form a hexagonal lattice as a rule. Also the vesicles can be unstable: a coalescence transforms fine-grained networks of elementary vesicle-processors into a coarse-grained assembles of non-lattice vesicular structures. When BZ vesicles, quite possibly of different sizes, are aggregated into an assembly and tightly packed, they are represented by

* Corresponding author. Tel.: +44 1173282662.
E-mail addresses: ramon.alonso@upm.es (R. Alonso-Sanz), andrew.adamatzky@uwe.ac.uk (A. Adamatzky).

1878-7789/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.nancom.2013.10.001

a DT. A computation on a DT is implemented in the same manner as on a regular cellular automaton [20].

The paper is structured as follows. The DT and DT automata are introduced in Section 2. The effect of majority memory on automata on DT is studied in Section 3, whereas Section 4 deals with the effects of weighted memory. Both types of memory are scrutinized when starting with a single active node in Section 5. DT automaton networks with non-parity rules and other contexts are briefly tackled in Section 6.

2. Automata on DTs

A DT for a set **P** of $\mathbb{N}$ points in the plane is a triangulation DT(**P**) such that no point in **P** is inside the circumcircle of any triangle in DT(**P**). The Delaunay triangulation DT(**P**) corresponds to the dual graph of the Voronoi diagram for **P**, thus the latter is obtained by joining the centers of the circumcircles in DT(**P**) defining neighboring triangles [21]. As an example, Table 1 shows a simple DT on four points and its corresponding Voronoi diagram. The left panel of Table 1 shows the $C_4^3 = 4$ circumcircles of four points chosen to be the vertices of an equilateral triangle and its barycenter, so that the *central* circle (the circumcircle of the *vertices*, which has the barycenter inside) does not generate any link. In large networks, each vertex of a DT has on average six surrounding triangles, which implies a mean connectivity $K = 6$.

Automata on DTs were originally introduced in [20] and studied in a context of excitation dynamics. In the present paper we develop ideas of [20] along the lines of memory-enriched automata and global dynamics. In the automata on DTs 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 [20]. 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: $\sigma_i^{(T+1)} = \phi\left(\{\sigma_j^{(T)}\} \in \mathcal{N}_i\right)$, 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 behavior [22].

In the Markovian approach just outlined (referred to 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: $\sigma_i^{(T+1)} = \phi\left(\{s_j^{(T)}\} \in \mathcal{N}_j\right)$, $s_j^{(T)}$ being a state function of the series of states of the node j up to time-step T.

3. Majority memory

We will consider first the most frequent state (or *majority*) memory implementation. Thus, with memory limited to the last τ time-step state values:

$$s_i^{(T)} = \text{mode}\left(\sigma_i^{(T)}, \sigma_i^{(T-1)}, \ldots, \sigma_i^{(\top)}\right),$$

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 induces a lower effectiveness of even size τ-memories. That is why only odd size memory length will be implemented in the evolving figures of this section.

With unlimited trailing memory it is:

$$s_i^{(T)} = \text{mode}\left(\sigma_i^{(T)}, \sigma_i^{(T-1)}, \ldots, \sigma_i^{(1)}\right).$$

Table 2 shows the initial evolving patterns of a simulation of the parity rule on a DT automaton with $N = 11$ nodes. Red nodes denote node state values equal to one, black nodes 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 nodes exemplify the initial effect of memory: two of them are red at $T = 3$, but black as the most frequent state, whereas the other encircled node behaves in the opposite manner, thus black at $T = 3$, but red as the most frequent state.

Fig. 1 shows the evolution of the changing rate (the Hamming distance between two consecutive patterns) in ten different DTs based in one thousand nodes differently distributed at random in the unit circle. The actual mean connectivity obtained in the ten networks is $\overline{K} \simeq 5.92$. 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. Fig. 1 shows also the effect on the changing rate of endowing nodes with memory of the last τ state values (blue lines). As a general rule, the inertial effect of memory tends to reduce the changing rate compared to the ahistoric model. But in this scenario, with high connectivity, very low memory charges, e.g., $\tau = 3$ in Fig. 1 or the even size (so less effective) $\tau = 4$ do not have a significant effect on the changing rate. Memory charges of $\tau = 5$ (or $\tau = 7$) length have a limited effect, whereas $\tau = 9$, already has an apparent effect. With higher memory charges, such as $\tau = 13$ or $\tau = 29$ shown in the lower left panel of Fig. 1, memory reveals its effectiveness in moderating the changing rate, which tends to vary in the long term by nearly 0.2 and 0.3 respectively, after an initial *almost-oscillatory* behavior which ceases by $T > \tau + 1$, thus $T = 14$ and $T = 30$ respectively. With unlimited trailing memory (lower right panel), this *oscillatory* pattern is never truncated, so that a rather unexpected quasi-oscillatory behavior results with full memory. These *oscillatory* patterns get a notable amplitude, albeit below 0.5. With no exception, the proportion of node states having one given state value (*density*), oscillates near to 0.5 regardless of the model considered.

Fig. 2 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. Fig. 2 operates with the same initial simulations as Fig. 1, thus a highly connected scenario in which damage propagates very rapidly without memory (*butterfly effect*), so that by $T = 25$ the red curves already oscillate around 50 percent