

Attractors in synchronous and asynchronous genetic regulatory networks

Marco Pedicini (Roma Tre University)

in collaboration with Maria Concetta Palumbo (IAC-CNR),
Filippo Castiglione (IAC-CNR)

ITALIAN WORKSHOP ON ARTIFICIAL LIFE
AND EVOLUTIONARY COMPUTATION

2017 Edition - 19th-21st September 2017, Venice, Italy

Networks and Graphs

- For **Genetic Regulatory Networks** various presentations are possible:
 - boolean networks,
 - polynomial dynamical systems,
 - activation/inhibition graphs.

For a fixed network R , we have to associate its **dynamics**

$$T_R : S \rightarrow S;$$

- For any R , the T_R determines the **state transition graph**

$$G(T_R) = \{(s, s') \mid s' = T_R(s)\}.$$

Networks Properties

Since Stuart Kauffman studies one relevant task in systems biology consists in inferring properties of the STG starting from the network and its dynamics.

The typical question is to characterise/find stability in the space of the states:

- **point attractors**: $s \in S$ such that $T_R(s) = s$,
- **limit cycles**: $s_1, s_2, \dots, s_k \in S$ such that $s_{i+1} = T_R(s_i)$, and $s_1 = s_k$.

Boolean Expressions

Let us consider a boolean regulatory network of 3 genes, components are given as boolean expressions:

$$f_1(x_1, x_2, x_3) = \neg x_3 \wedge (x_1 \vee x_2)$$

$$f_2(x_1, x_2, x_3) = x_1 \wedge x_3$$

$$f_3(x_1, x_2, x_3) = \neg x_3 \vee (x_1 \wedge x_2)$$

So that, if (x_1, x_2, x_3) represents an element of S , then we have:

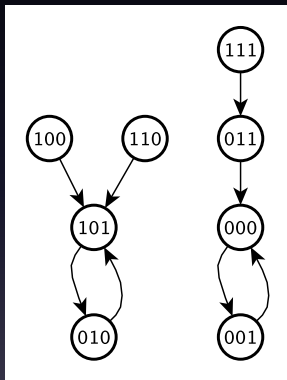
$$T_R(x_1, x_2, x_3) = (\neg x_3 \wedge (x_1 \vee x_2), x_1 \wedge x_3, \neg x_3 \vee (x_1 \wedge x_2))$$

This is called **synchronous dynamics** since expressions for all the genes are updated at the same time:

$$T_R(s) = (f_1(s), f_2(s), f_3(s))$$

Synchronous Dynamics

STG of the 3-genes boolean network:



Evolution is deterministic: for any given state there is one possible way to proceed, in the graph out-degree for any node is exactly 1.

Synchronous Dynamics

Another example with 8 genes:



Attractors with synchronous dynamics

In the previous example the entire STG of the network was explored in order to find the two cycles.

In general, since the STG of a boolean network of n genes contains 2^n nodes, we go easily beyond the current computational capacity even when $n > 50$.

But, under certain hypothesis on the **cycles structure** (**small number**, **short transients** to reach them, **short length**) it is possible to find them even in networks with **thousands of genes**:

n	$ G(T) $	# limit cycles	visited nodes	paths (# SAT calls)	max length of paths	reference	time
10	2^{10}	$1 \times 6, 1 \times 1$	25	6	16	[?]	0.260959
10	2^{10}	$1 \times 7, 1 \times 1$	29	6	16	[?]	0.206969
23	2^{23}	3×1	42	7	16	[?]	0.275958
40	2^{40}	$1 \times 6, 8 \times 1$	136	14	32	[?]	3.59245
51	2^{51}	$2 \times 1, 1 \times 3$	97	10	64	[?]	5.96909
52	2^{52}	7×1	172	13	32	[?]	7.71783
1000	2^{1000}	2×2	92	7	32	random net	444.09

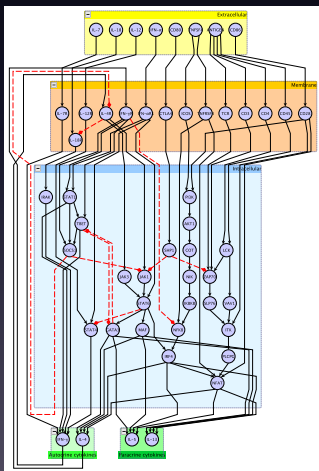
A SAT based approach

Elena Dubrova designed an algorithm to find these limit cycles in networks with synchronous dynamics:

```
1: function Cycles( $T$ )
2:   Initialise
3:    $path\_length := 1$ 
4:    $F := PathExpression[-path\_length, 0]$ 
5:   while Satisfiability[ $F$ ] do
6:      $(c_{-path\_length}, \dots, c_0) := SAT[F]$ 
7:     if TestLoops[ $(c_{-path\_length}, \dots, c_0)$ ] then
8:        $c_j$  minimal state forming the loop
9:        $Attractors(s_0) := Attractors(s_0) \wedge (s_0 \leftrightarrow c_j)$ 
10:       $F := F \wedge \neg Attractors(s_0)$ 
11:     end if
12:     if attractor_is_found then
13:       attractor_is_found := false
14:     else
15:        $F := PathExpression[-2path\_length, 0]$ 
16:        $path\_length := 2 path\_length$ 
17:     end if
18:   end while
19: end function
```


Th1/Th2 switch network

Presented in Pedicini et al. *Combining network modelling and gene expression microarray analysis to explore the dynamics of th1 and th2 cell regulation*, *PLoS computational biology*, 6(12):e1001032, 2010



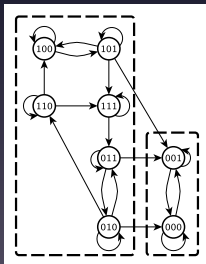
We have a $n = 51$ genes network for which we want to compute the lattice of loose attractors

Asynchronous Dynamics

For a given network specification, we can derive a non-deterministic dynamics, by taking in account the update of a single gene expression at each step:

$$T_R(x_1, x_2, x_3) = \begin{cases} (f_1(s), x_2, x_3) & \text{if we decide to update gene 1} \\ (x_1, f_2(s), x_3) & \text{if we decide to update gene 2} \\ (x_1, x_2, f_3(s)) & \text{if we decide to update gene 3} \end{cases}$$

In our first example we get the following STG:



Asynchronous Dynamics

With the non deterministic evolution the STG is more complex:



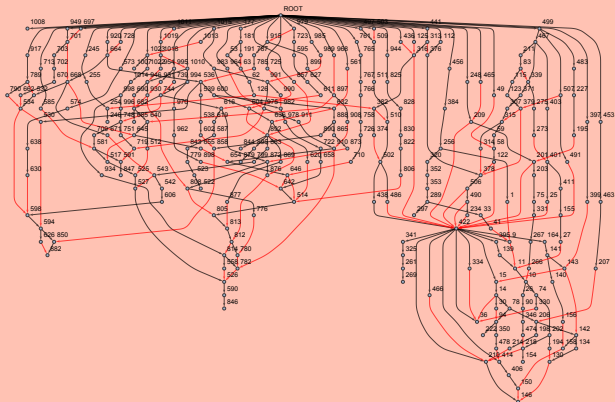
Loose Attractors Lattice

In analogy, with previous results on the synchronous case we extended Dubrova-Teslenko's approach to the asynchronous case in order to find loose attractors and relations between them.

The problem is well known on graphs, and it has an optimal solution by Tarjan's Algorithm (a variant of depth first graph visit).

The problem is that in order to visit the graph, we have to know the graph, but here, we cannot have direct access to the STG graph.

Visited Lattice of the STG



Adversarial order

We implemented an algorithm which combines Dubrova-Teslenko's algorithm with the Tarjan one: in the code, when we get a **new path from the SAT** solver we replace the test for the presence of a loop in the path with the execution of the DFS of Tarjan.

We developed a **multithread version of Tarjan's algorithm**, in order to manage multiple instances of the visit at the same time.

The idea we take from Dubrova is the interleaving the SAT solver with the DFS visit algorithm and we develop the Tarjan variant in order to manage DFS visit in adversarial order.

$n = 10$ STG

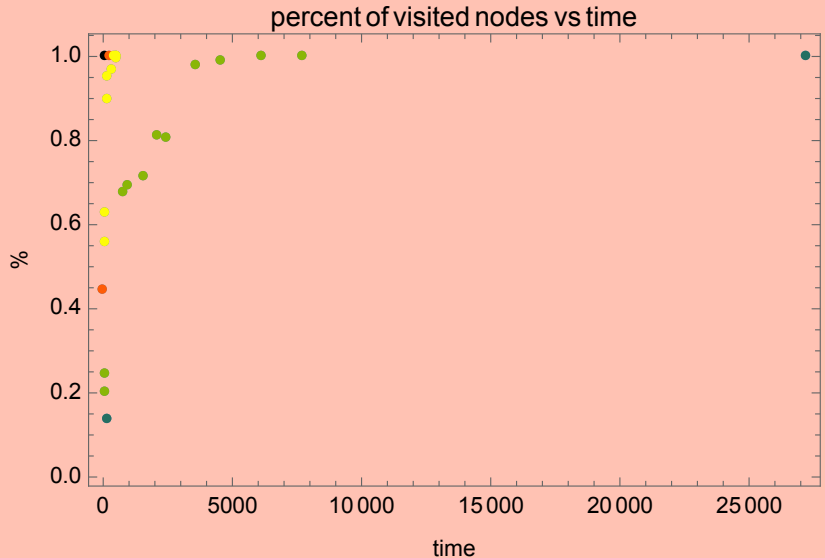


Results on Random Networks

n	visited nodes	paths (# SAT calls)	max length of paths	density (K/N)	inh/(act+inh)	time
5	32	18	4	0.28	0.142857	4.64729
7	57	12	16	0.285714	0.	4.38933
7	128	62	16	0.142857	0.142857	189.885
8	144	12	32	0.265625	0.176471	25.0422
8	162	17	32	0.15625	0.1	61.0997
8	230	41	32	0.265625	0.176471	106.767
8	244	36	32	0.265625	0.176471	138.548
8	249	61	32	0.234375	0.0666667	282.521
8	256	130	8	0.09375	0.5	408.086
8	255	124	8	0.203125	0.384615	474.364
8	256	109	16	0.203125	0.615385	481.884
8	256	125	8	0.140625	0.222222	486.152
10	210	10	64	0.2	0.1	34.9877
10	253	13	64	0.15	0.0666667	57.6532
10	693	52	64	0.29	0.0344828	738.52
10	732	117	32	0.2	0.2	1570.55
10	834	135	32	0.18	0.444444	2101.32
10	830	125	32	0.14	0.571429	2382.64
10	1002	169	32	0.24	0.25	3529.33
10	1018	182	32	0.13	0.230769	4561.46
10	1024	224	32	0.17	0.117647	6125.17
10	1024	485	16	0.23	0.478261	7703.87
11	286	17	64	0.190083	0.	100.223
11	2048	572	16	0.14876	0.111111	27188.9

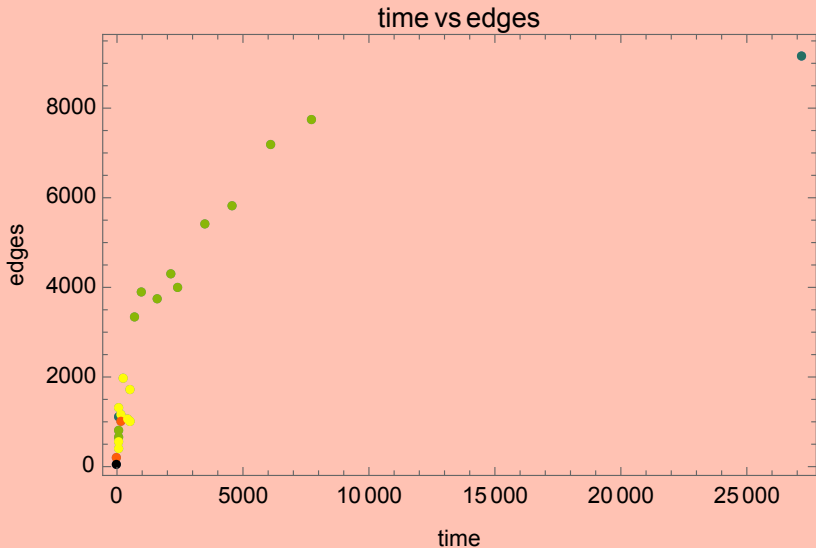
Tractable Region

of visited nodes/ total nodes vs time




Tractable Region


of generated edges vs time




Thanks !

 R Iris Bahar, Erica A Frohm, Charles M Gaona, Gary D Hachtel, Enrico Macii, Abelardo Pardo, and Fabio Somenzi.
Algebraic decision diagrams and their applications.


In Computer-Aided Design, 1993. ICCAD-93. Digest of Technical Papers., 1993 IEEE/ACM International Conference on, pages 188–191. IEEE, 1993.

 Roderick Bloem, Harold N Gabow, and Fabio Somenzi.
An algorithm for strongly connected component analysis in $n \log n$ symbolic steps.

Formal Methods in System Design, 28(1):37–56, 2006.

 Hidde De Jong.
Modeling and simulation of genetic regulatory systems: a literature review.

Journal of computational biology, 9(1):67–103, 2002.

 E. Dubrova and M. Teslenko.
A SAT-based algorithm for finding attractors in synchronous boolean networks.

Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 8(5):1393–1399, Sept 2011.



Harold N Gabow.

Path-based depth-first search for strong and biconnected components.

Information Processing Letters, 74(3-4):107–114, 2000.



Abhishek Garg, Kartik Mohanram, Alessandro Di Cara, Giovanni De Micheli, and Ioannis Xenarios.

Modeling stochasticity and robustness in gene regulatory networks.

Bioinformatics, 25(12):i101–i109, 2009.



Martin Hopfensitz, Christoph Müssel, Markus Maucher, and Hans A Kestler.

Attractors in boolean networks: a tutorial.

Computational Statistics, 28(1):19–36, 2013.



Stuart A Kauffman.

The origins of order: Self-organization and selection in evolution.

Oxford University Press, USA, 1993.



Ian Munro.

Efficient determination of the transitive closure of a directed graph.

Information Processing Letters, 1(2):56–58, 1971.



Marco Pedicini, Fredrik Barrenäs, Trevor Clancy, Filippo Castiglione, Eivind Hovig, Kartiek Kanduri, Daniele Santoni, and Mikael Benson.

Combining network modeling and gene expression microarray analysis to explore the dynamics of th1 and th2 cell regulation.

PLoS computational biology, 6(12):e1001032, 2010.



Paul Purdom.

A transitive closure algorithm.

BIT Numerical Mathematics, 10(1):76–94, 1970.



Robert Tarjan.

Depth-first search and linear graph algorithms.

SIAM J. Comput., 1(2):146–160, 1972.



Alan Veliz-Cuba and Reinhard Laubenbacher.

On the computation of fixed points in Boolean networks.

J. Appl. Math. Comput., 39(1-2):145–153, 2012.



Desheng Zheng, Guowu Yang, Xiaoyu Li, Zhicai Wang,
Feng Liu, and Lei He.

An efficient algorithm for computing attractors of
synchronous and asynchronous boolean networks.

PloS one, 8(4):e60593, 2013.