

Computational Study of Activation Dynamics on Networks of Arbitrary Structure

A. Semenov,
S. Kochemazov, D. Gorbatenko
ISDCT RAS, Irkutsk

NET-2017

General Information

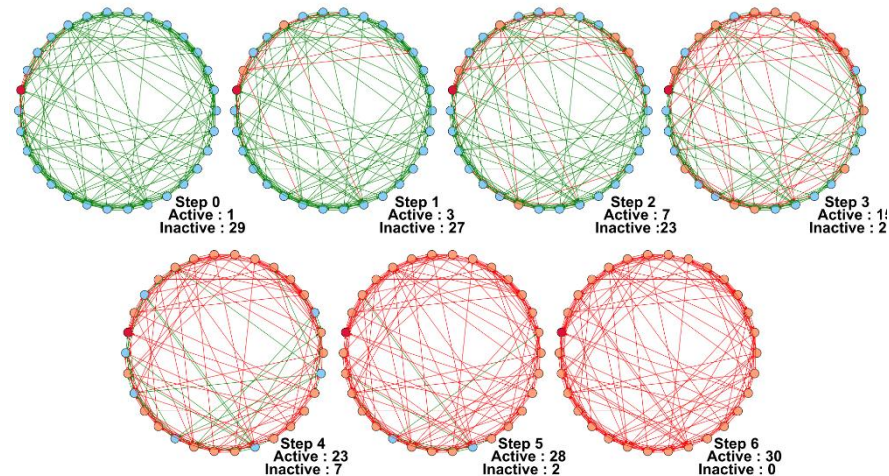
In this report we present one approach to modeling the phenomena of activation dynamics on networks of arbitrary (in general case) structure.

The main idea consists in considering the processes of network activation based on the properties of a discrete function specified by network graph. In that case we can apply state-of-the-art combinatorial algorithms to solve several naturally arising (in this context) problems. In particular, hereinafter we employ the algorithms for solving Boolean satisfiability problem (SAT).

Let us (informally) say a few words about the used models of activation dynamics. First, all the considered processes take place at discrete time. Second, at each following time moment, all agents in a network “recalculate their states” synchronously depending on the states in which they are at the present moment (synchronous dynamics). In the context of models of such kind, we are interested in how some property spreads through the network: the simplest example of such networks are the so-called Boolean networks, where the state “1” means action and “0” means inaction. The corresponding networks can be viewed as models for studying the phenomena related to conforming behavior (in spirit of “Threshold models of collective behavior” by M. Granovetter [1]).

[1] M.Granovetter (1978) “Threshold models of collective behavior” American Journal of Sociology 83 : 1420–1443.

General information



At the figure above there is presented a synchronous Boolean network on 30 vertices, where all agents except one (marked with red color) at the initial time moment are inactive. After 6 time moments all agents of this network moved to the active state. Looking forward, we would like to note that such fast activation was possible thanks to an advantageous disposition of agents-activators (marked with red).

The author of this report believes (perhaps, presumptuously) that similar activation problems can be formulated in application to various real world networks (information networks, social networks, biological networks, economical networks).

It is proposed to find a class of networks from different problem areas, which are similar in their activation principles. Since the author of this report is not a specialist in aforementioned areas, hereinafter we will mainly study the principles of activation, related combinatorial problems and methods for solving the latter.

Basic model

Consider a simple directed graph $G = (V, A)$, $V = \{v_1, \dots, v_n\}$ – a set of vertices (agents), A – a set of arcs.

For an arbitrary $v \in V$ we define its neighborhood $U_v = \{v'_1, \dots, v'_{l(v)}\}$ as a set of vertices, the arcs from which go into v (they interpret how other vertices influence v).

At each time moment $t \in \{0, 1, \dots\}$ with an arbitrary $v \in V$ we associate an element $\omega_v(t)$ from some set Ω_v (in a general case, an arbitrary symbol, such as 0 or 1).

For each $v \in V$ we define the rule f_v , in accordance with which at time moment $t + 1$ we associate with elements $\omega_{v'_1}(t), \dots, \omega_{v'_{l(v)}}(t)$ elements $\omega_v(t + 1) \in \Omega_v$:

$$\omega_v(t + 1) = f_v \left(\omega_{v'_1}(t), \dots, \omega_{v'_{l(v)}}(t) \right). \quad (i)$$

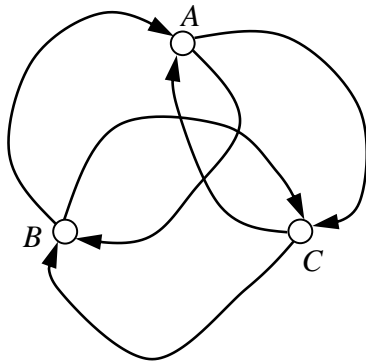
The function $f_v: \Omega_{v'_1} \times \dots \times \Omega_{v'_{l(v)}} \rightarrow \Omega_v$ is called a weight function of vertex v . By defining the weight functions in the network the following function is defined

$$F_G: \Omega_1 \times \dots \times \Omega_n \rightarrow \Omega_1 \times \dots \times \Omega_n$$

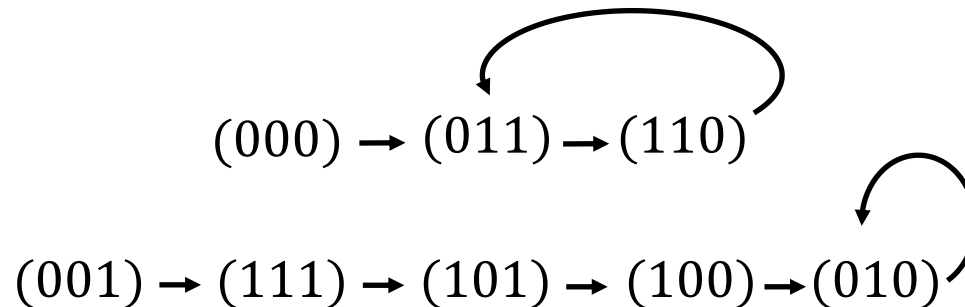
The values of F_G , considered at time moments t , are called network states (we refer to the state at time moment t as $W_G(t)$).

Example

1. If for each $v \in V$ $\Omega_v = \{0,1\}$, then the described network with weight functions specified by (i) is called a Synchronous Boolean network (SBN) or a Kauffman network [1].



$f_B(t)$	$f_C(t)$	$f_A(t+1)$
0	0	0
0	1	1
1	0	0
1	1	1



[2] Kauffman S. Metabolic stability and epigenesis in randomly constructed genetic nets // Journal of Theoretical Biology. 1969. Vol. 22, N. 3. Pp. 437–467.

Activation Dynamics on Networks

- Common feature of all activation processes considered below is the following.
- In the network there always are the agents, which at the initial time moment have some property (for example they are “active”).
- All the remaining agents do not possess this property at $t = 0$, however, it can be transferred to other agents at the consecutive time moments depending on this or that conditions. We refer to such property transfer as to ***network activation***.
- We call the agents that possess the considered property at the initial time moment the ***activators***. In general, any agent can be activator. We refer to agents who are not activators as to simple agents.
- Below we consider the following general problem: how to dispose relatively small number of activators within the network in such a way that the network will become activated in relatively small amount of time moments?

Activation Dynamics on Networks

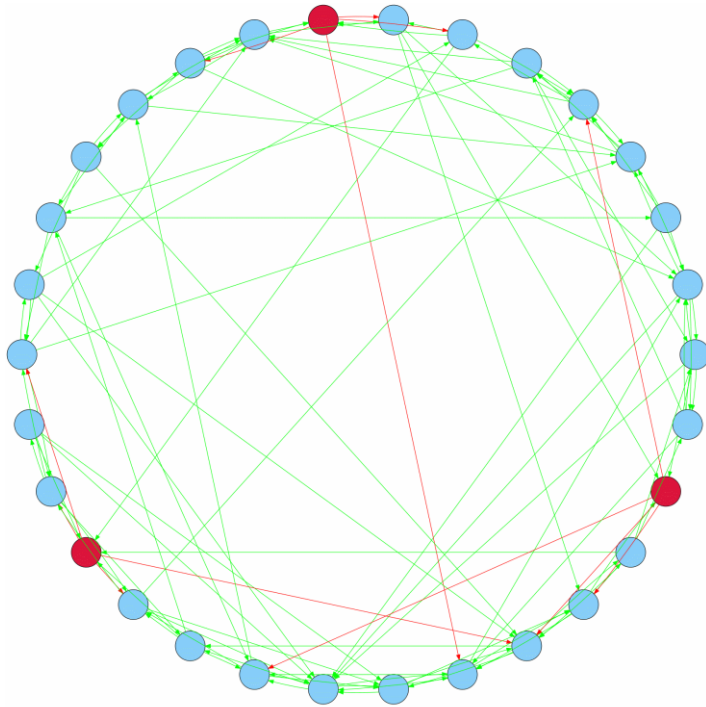
To solve the problems of described type in application to networks modeling the conforming behavior in collectives, we used in [3] the SAT approach. SAT is a Boolean satisfiability problem. It is a historically first NP-complete (NP-hard in the search variant) problem. Nevertheless, state-of-the-art algorithms for solving SAT (first of all, CDCL) show remarkably high effectiveness even on formulas of huge dimension (dozens of thousands of variables and millions of constraints – clauses). The best SAT solvers even cope with such justified hard SAT instances as cryptanalysis of a number of ciphers.

Briefly, the main principles of how the problem of finding dispositions of activators in the network, modeling the conforming behavior, is reduced to SAT, are described in [3].

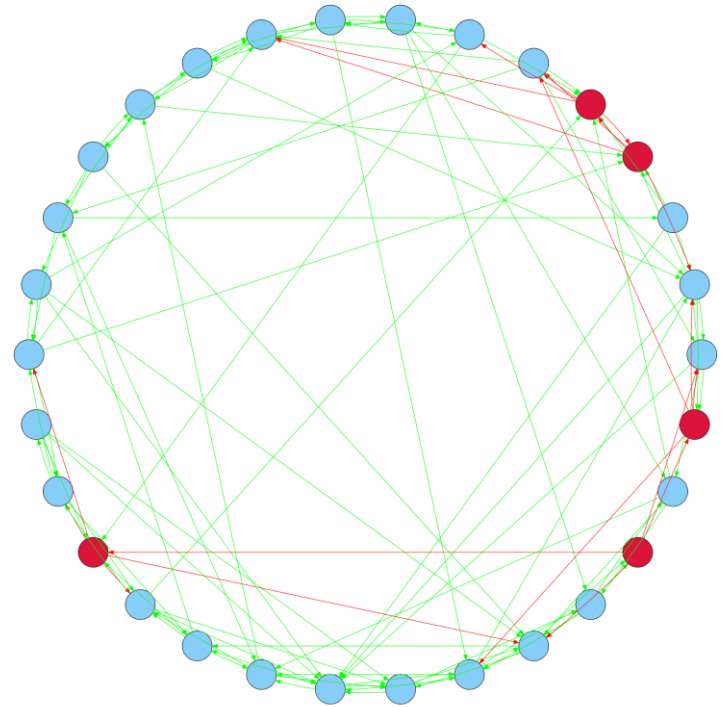
[3] Kochemazov S., Semenov A. Using Synchronous Boolean Networks to Model Several Phenomena of Collective Behavior // PLoS ONE. 2014. 9: e115156. P. 1-28.

Using the state-of-the-art SAT solvers we managed to solve corresponding problems for randomly generated networks with more than 500 vertices (using known random graph models of Erdos-Renyi, Watts-Strogatz and Barabasi-Albert). Now using new propositional encoding techniques and parameterisation methods for SAT solvers we can do it for networks with several thousand vertices.

Example: advantageous disposition of activators (instigators)

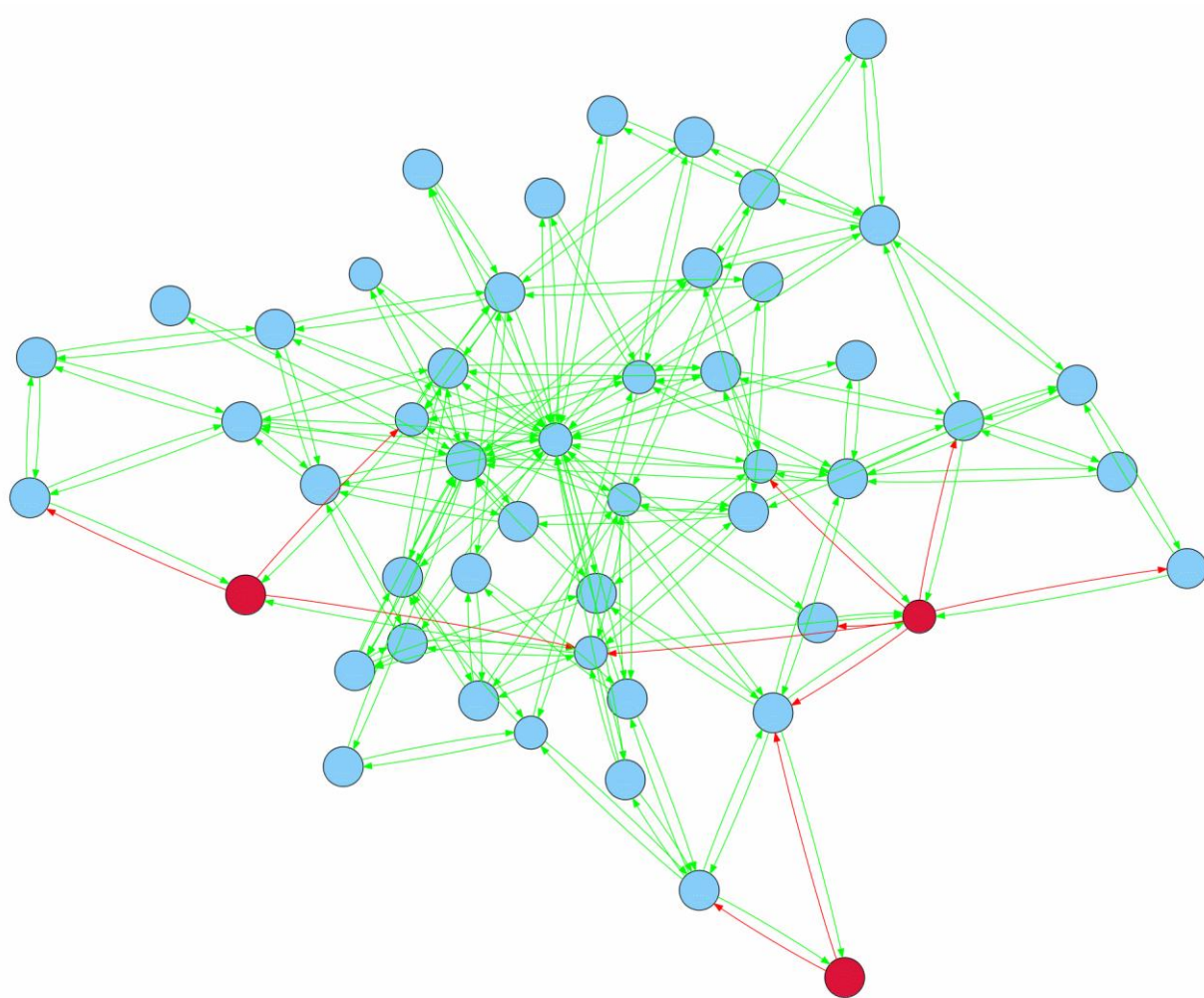


Step 0
Instigators: 3
Loyalists: 0
Active: 3
Inactive: 27



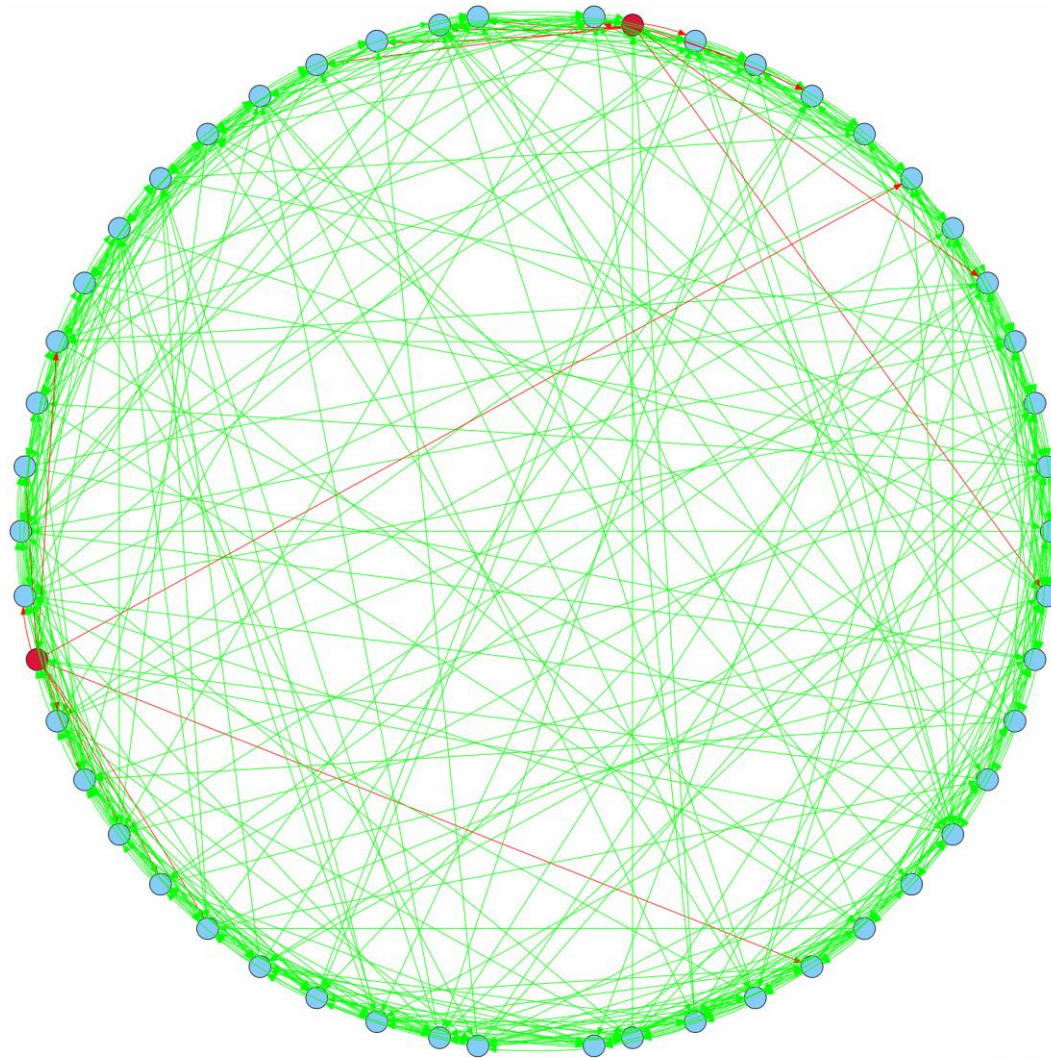
Step 0
Instigators: 5
Loyalists: 0
Active: 5
Inactive: 25

Example: Barabasi-Albert graph with 50 vertices



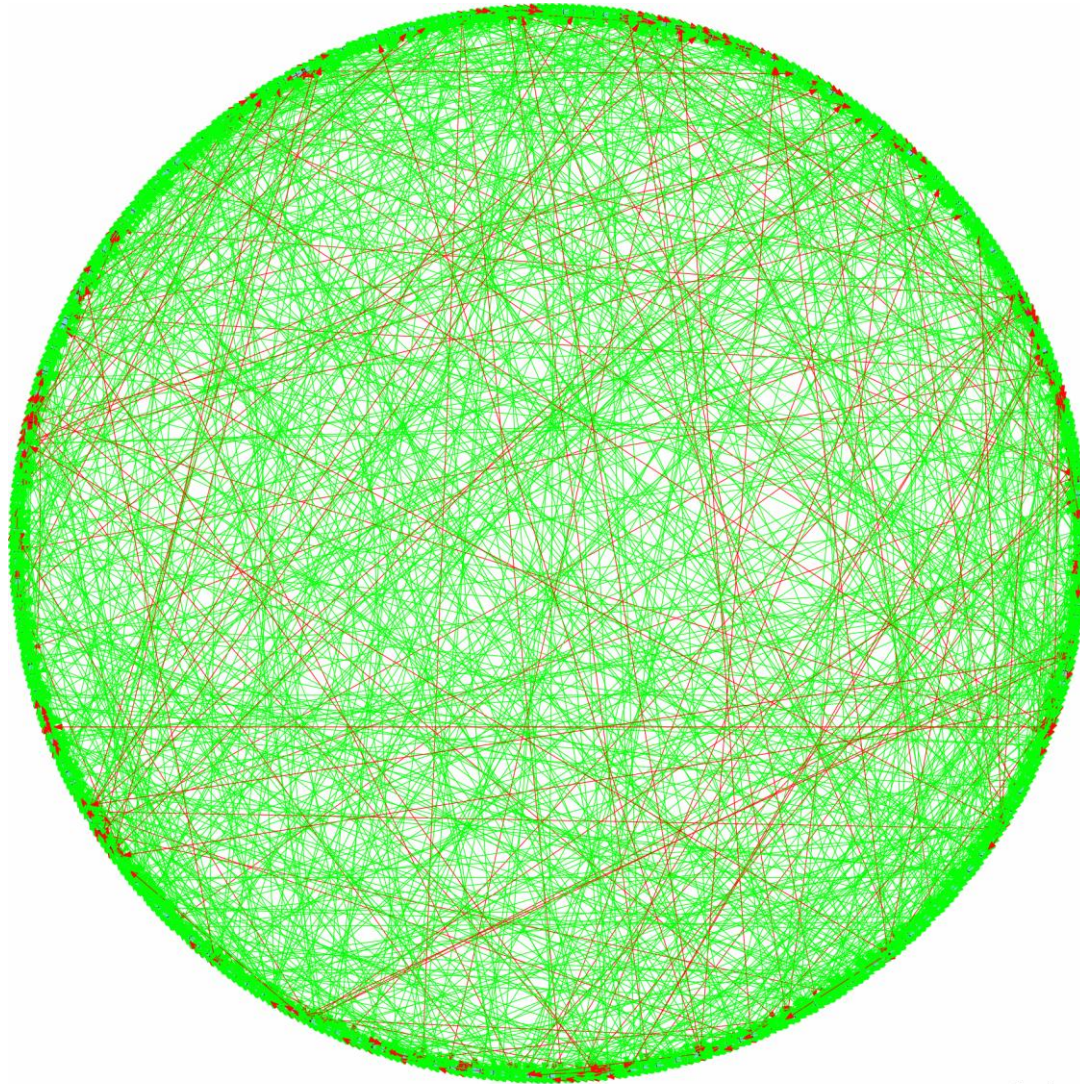
Step 0
Instigators: 3
Loyalists: 0
Active : 3
Inactive : 47

Example: Watts-Strogatz graph with 50 vertices



Step 0
Instigators: 2
Loyalists: 0
Active : 2
Inactive : 48

Example: Watts-Strogatz graph with 500 vertices



Step 0
Instigators: 50
Loyalists: 0
Active : 50
Inactive : 450

Construction of attack as the network activation dynamics problem

- Consider G as Discrete Dynamic System (DDS), which is functioning at time moments $t \in \{0, 1, \dots\}$, $t = 0$ is the initial time moment.
- $Q = \{q_1, \dots, q_l\}$ – a set of vulnerabilities of hosts
- At each t with $v \in V$ we associate the Boolean vector
$$\alpha^v(t) = (\alpha_1^v, \dots, \alpha_l^v, \alpha_{l+1}^v(t), \dots, \alpha_r^v(t)), r \geq l$$
- $(\alpha_1^v, \dots, \alpha_l^v)$ - vector of vulnerabilities. $\alpha_j^v = 1, j \in \{1, \dots, l\}$ iff there is a vulnerability q_j on host v .
- At coordinates $\alpha^v(t)$ number $l + 1, \dots, r$ there is the information that can change with time. In particular, it reflects the access rights from node v to other network nodes. We refer to the vector $(\alpha_{l+1}^v(t), \dots, \alpha_r^v(t))$ as to vector of possibilities of host v at time moment t .
- Assuming that at $t = 0$ all states are defined for all hosts, it is possible to consider the transitions of the network to consecutive states as synchronous recalculation of vectors of possibilities of all hosts at each time moment according to some fixed rules.

Transition rules

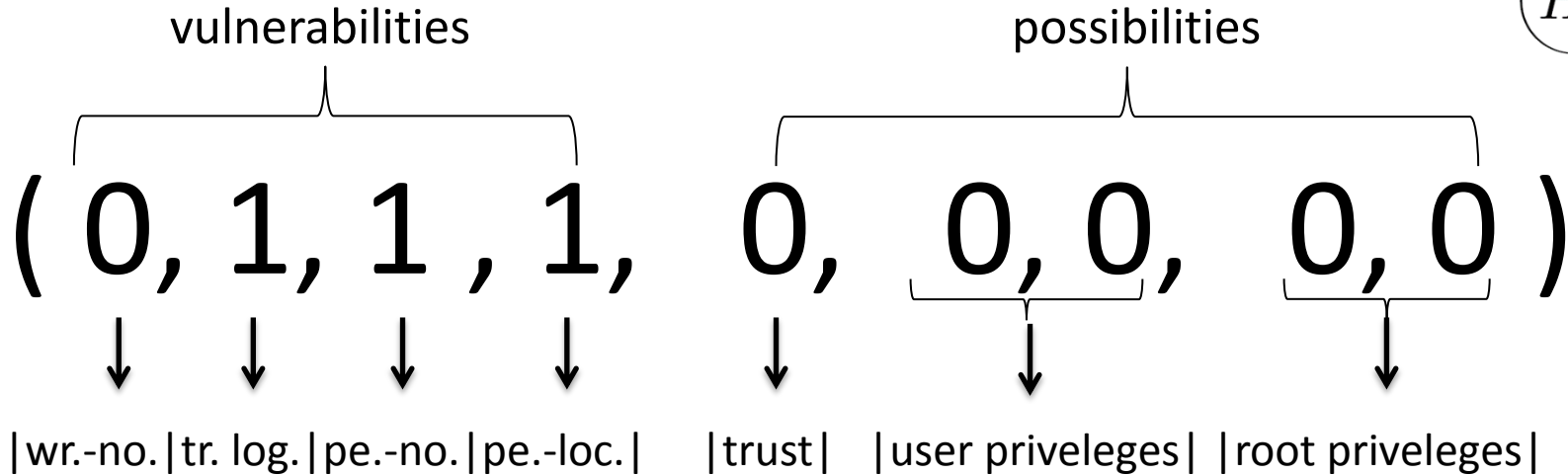
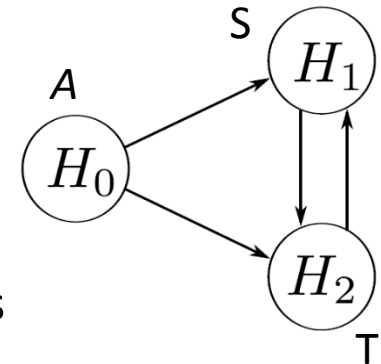
- In the present work in the role of transition rules we used pairs of the kind (precondition, postcondition), described in [5]. In particular, we considered several so-called elementary attacks.

Elementary attack Trust Establishment	Description
<p><u>Preconditions:</u></p> <p>$\text{var}(T, \text{write noauth})$</p> <p>$\text{priv}_S^A \geq \text{user}$</p> <p>$\text{priv}_T^S = \text{none}$</p> <p>$\neg \text{var}(T, \text{trust})$</p>	<ul style="list-style-type: none"> Preconditions– conditions on network state, which are necessary for the attack to proceed. $\text{var}(T, v)$ – there is a vulnerability v on host T. $\text{priv}_{H_2}^{H_1}$ – reflects the access rights (privilege) which host H_1 possesses on host H_2 A – the host of adversary S – the host from which the attack is performed (source) T – the host - target of the attack.
<p><u>Postconditions:</u></p> <p>$\text{var}(T, \text{trust})$</p>	<ul style="list-style-type: none"> Postconditions– results of attack on the network

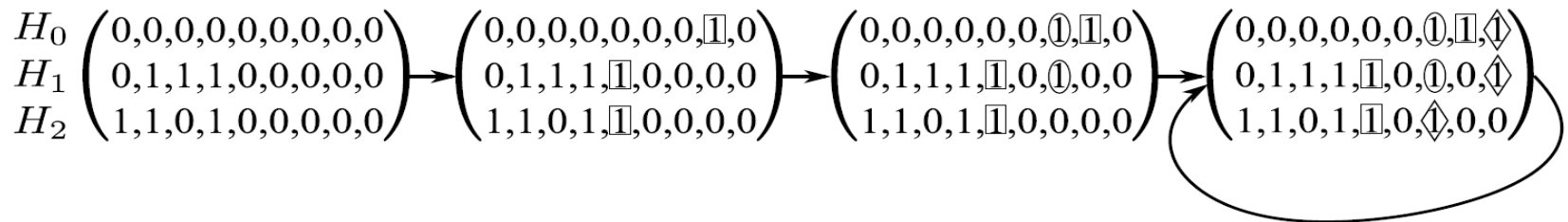
[5] Danforth M. Models for Threat Assessment in Networks. PhD Thesis, University of California, Davis. 2006.

Example

Let us consider the network of three hosts from [6] as an example. The host state at time moment $t \in \{0,1, \dots\}$ is a Boolean vector with 9 coordinates.

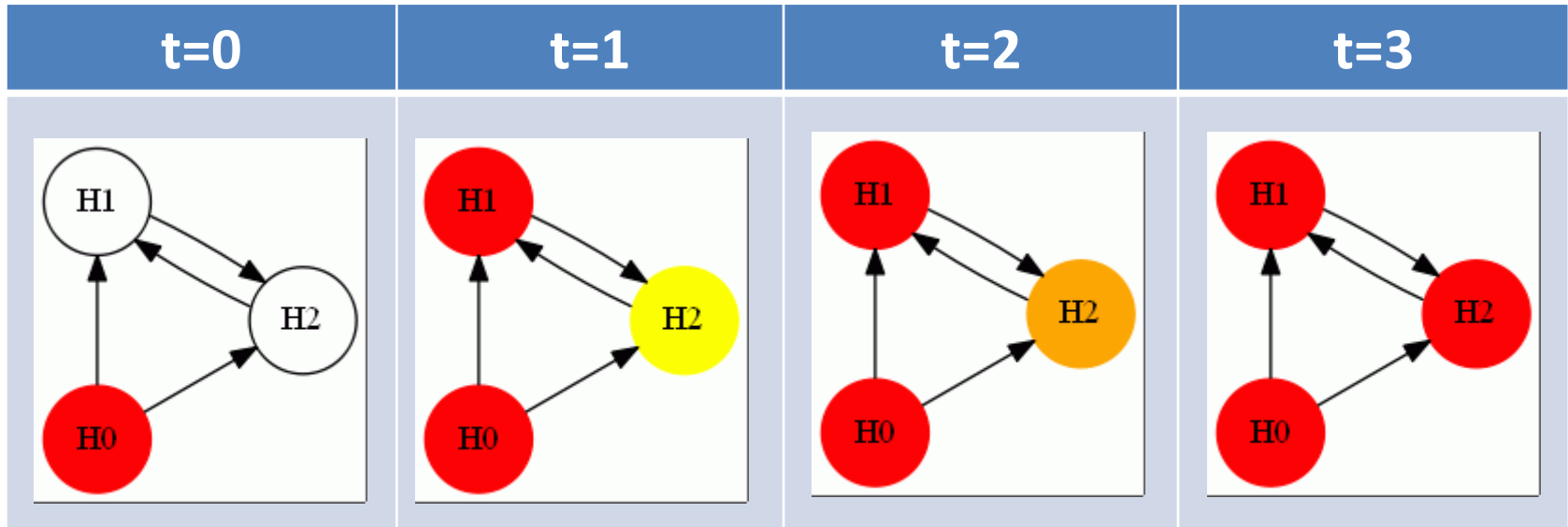


Below we present a fragment of state transition graph that shows two possible ways of attack development.



[6] Sheyner O., Haines J.W., Jha S., Lippman R., Wing J. M. Automated Generation and Analysis of Attack Graphs // Proceedings of 2002 IEEE Symposium on Security and Privacy. 2002. P. 273-284.

Example



Empty circle – the adversary have not used any vulnerabilities on this host and has no access rights on this host.

Yellow circle – the host was the target of exploited “Trust between host and all” vulnerability

Orange circle – the host was the target of exploited “trust-login” vulnerability which resulted in adversary rights at most of the User level.

Red circle – the host was the target of exploited “pe-noauth” or “pe-local” vulnerability which resulted in adversary having Superuser rights on this host.

Computational problems

In the context of the present research we considered the following problems:

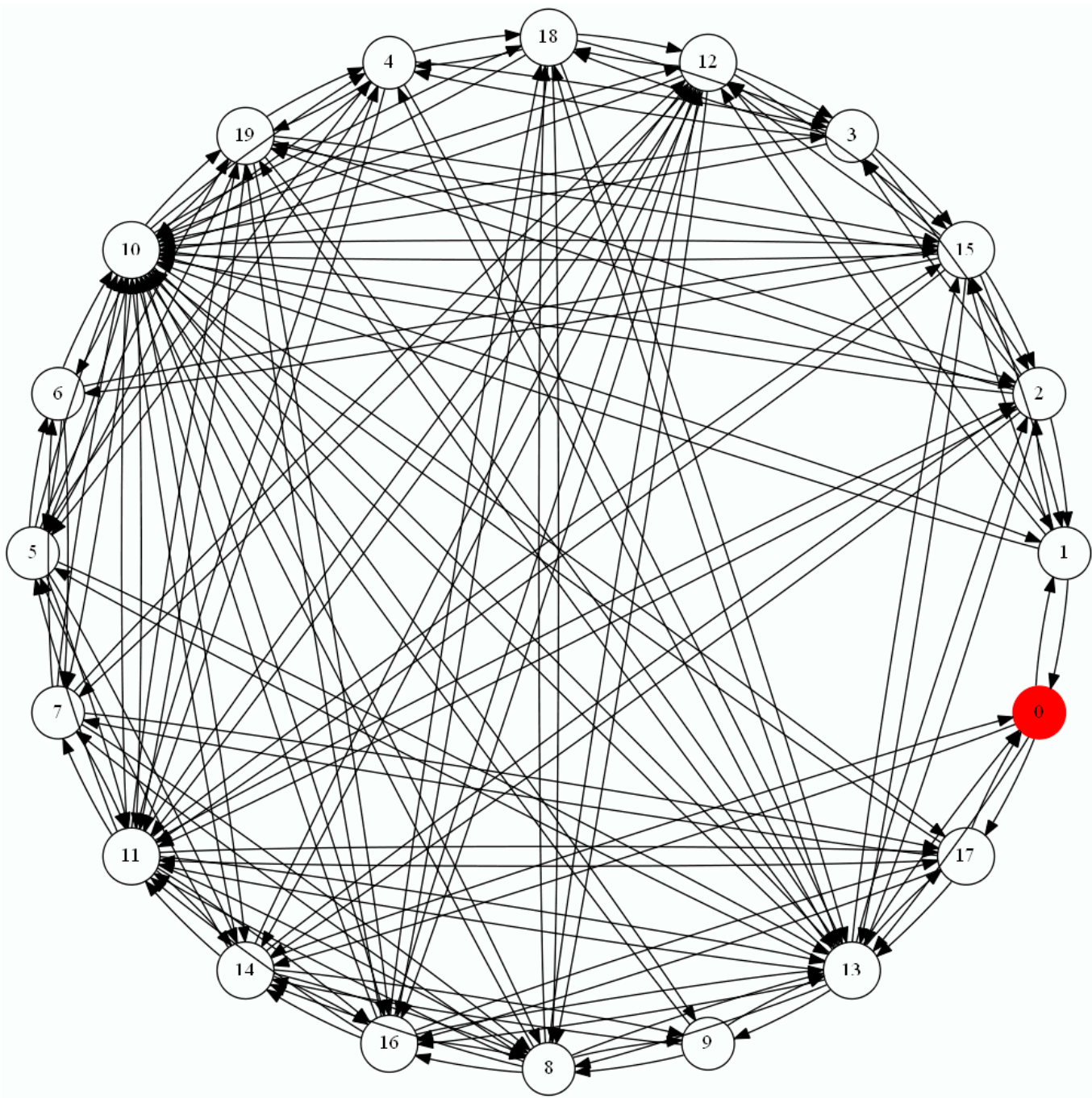
- 1. Construction of attacks.** Based on the known network graph construct all possible scenarios how the adversary might act. It is assumed that at each time moment the adversary exploits all available vulnerabilities.
- 2. Patch distribution problem.** For a known network graph with outlined target host to block a small number of vulnerabilities on several hosts in such a way that the adversary can not reach its goal, i.e. obtain the superuser rights on the target host.
- 3. Construction of attack with constraints.** To determine based on the known network graph with outlined target host if the adversary can obtain the superuser rights on the target hosts in at most n time steps, and having exploited vulnerabilities on at most m intermediate hosts.

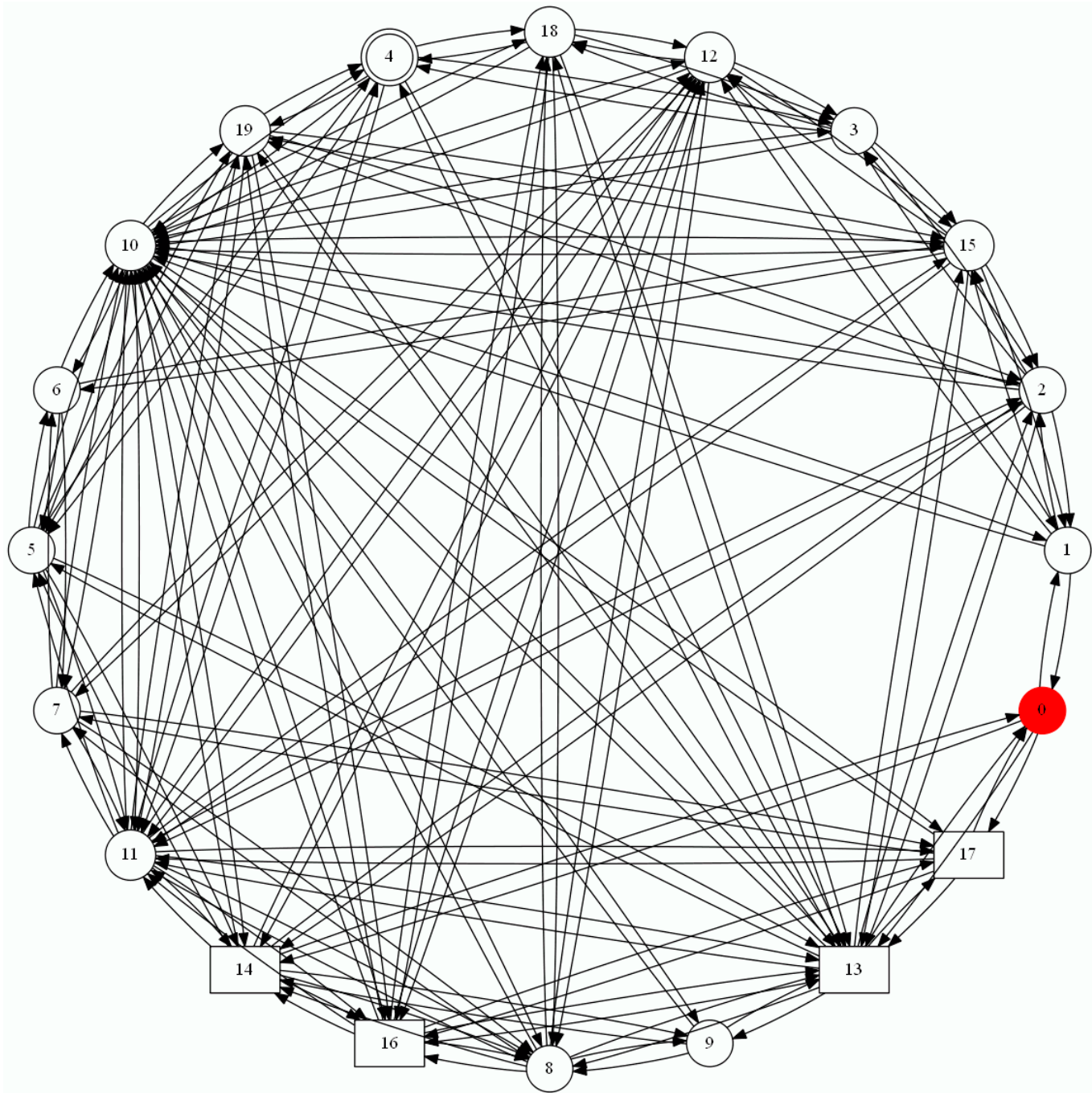
Results of experiments

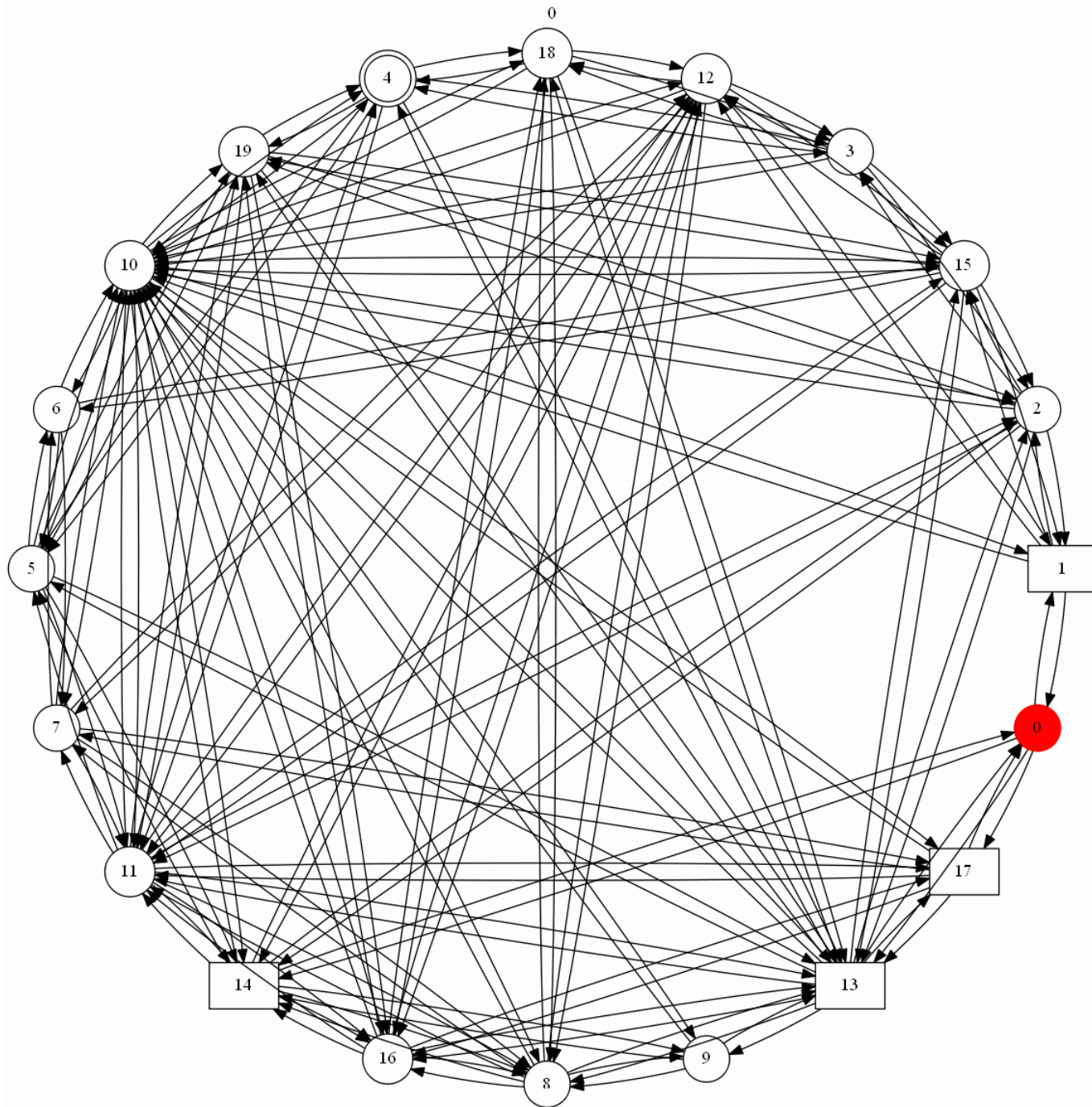
Computing platform: Core i3-2310m (8 Gb RAM, Windows 10). We used the SAT solver cryptominisat 5.0.0.

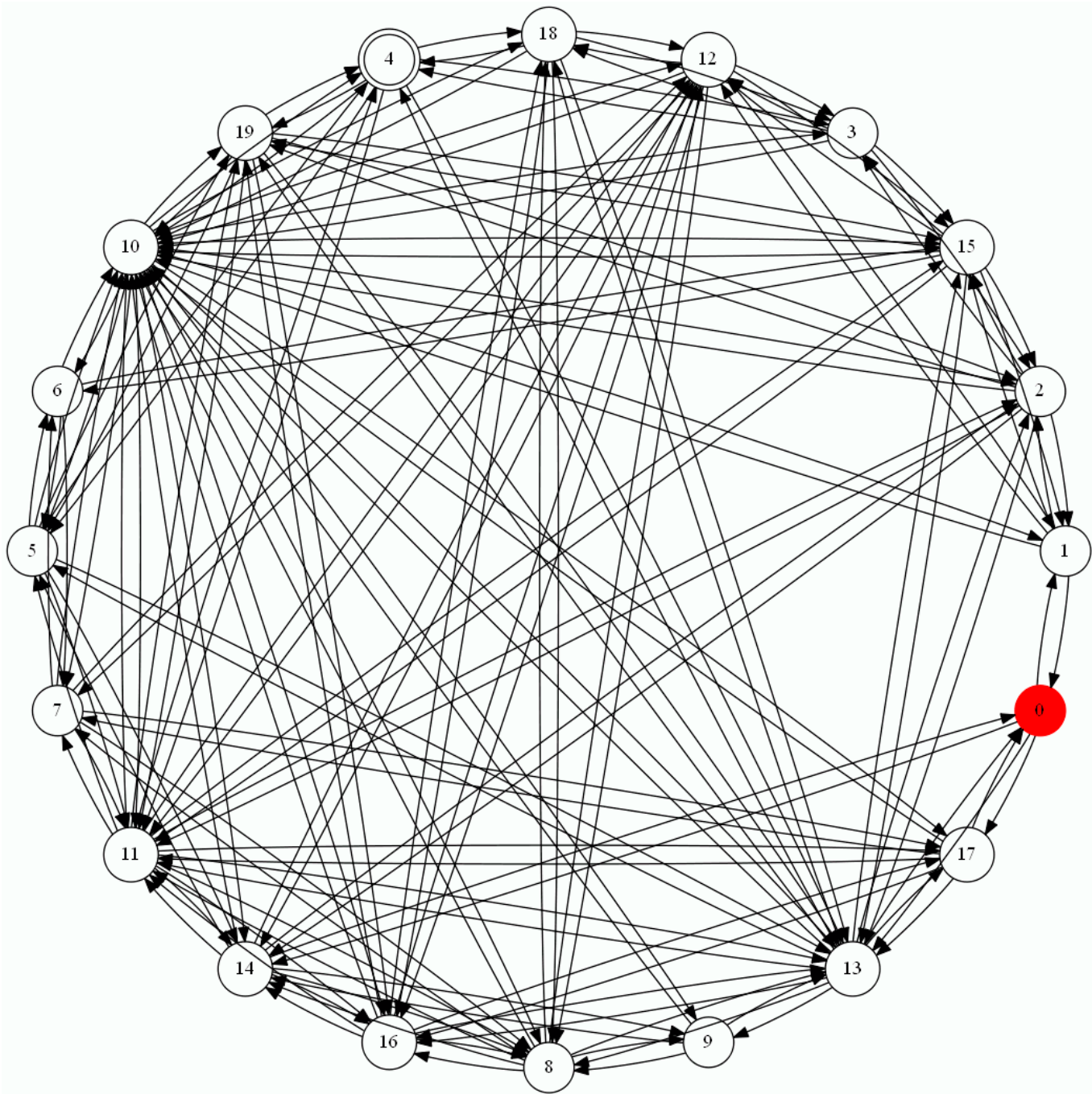
Network graphs were generated according to the Barabasi-Albert model. For each fixed network size we generated 10 tests.

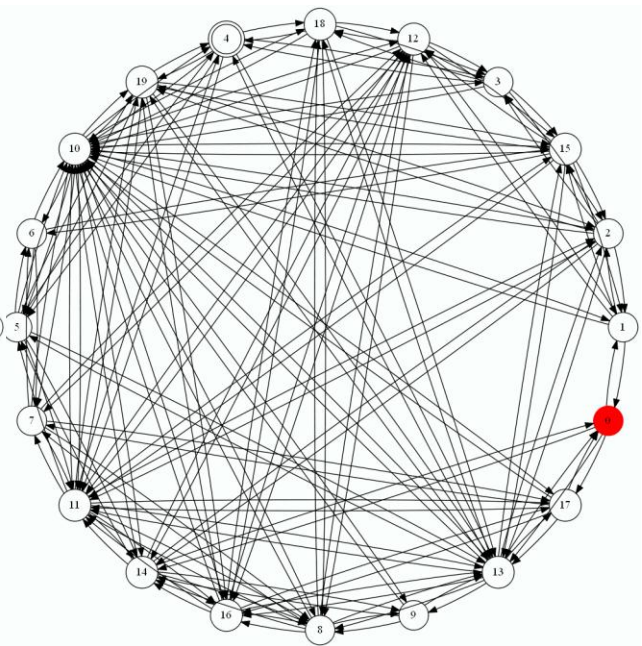
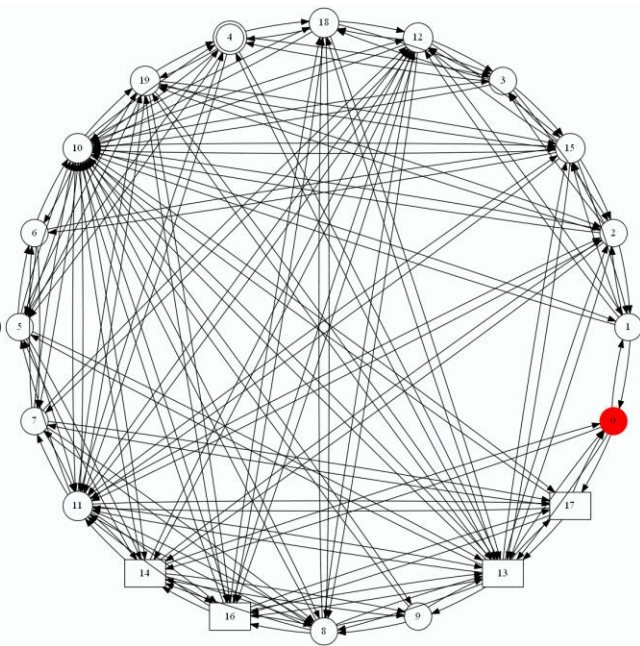
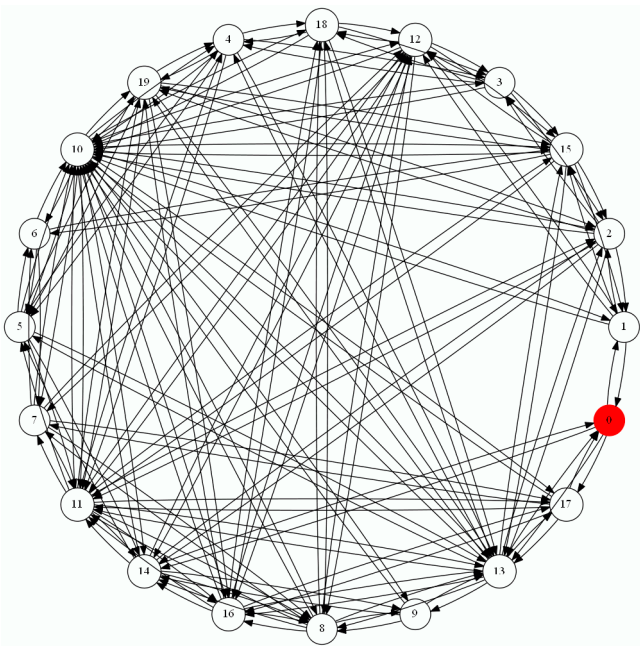
Network size	CNF size and solving time					
	Construction of attacks		Patch distribution problem		Construction of attack with constraints	
N=50	1,437 s.	13 444kb	1,757 s.	13 632kb	2,998 s.	18 588kb
N=100	6,351 s.	58 285kb	7,206 s.	58 773kb	13,789 s.	78 382kb
N=200	29,025 s.	251 699kb	31,716 s.	253 074kb	84,944 s.	339 299kb











Thank You
For Your Attention!