Analyzing the Complexity of Networks Relationships by Graph Theory and Frameworks Based on

Diana RIZESCU (AVRAM), Vasile AVRAM
Bucharest University of Economic Studies
diana.rizescu@csie.ase.ro, vasileavram@ie.ase.ro

The aim of this paper is to illustrate the application of a common conceptual "framework like" for representing relationships between elements (agents, parts, individuals) inside of complex systems and to find out indicators helping us in analysis and control of the modelled system. The representation of relationships networks realizes analogies with the real networks and randomly generated networks in a permanent effort in using the models inspired by these to model social and economic systems. The goal is to find a proof graph representation of the network relationships inside of "social groups" (does no matter if is any small group, an entire enterprise or organization or an entire social system) and to create the possibility to realize the measurements regarding its internal and external complexity, neighbouring, attractors and clusterization, communication etc. The graph framework like developed here allows to represent the attractors and their basins as neighbouring relationships that are proof using two Euler formulas, one specific to graphs and another defined by Euler to use when cutting in facets a sphere and adapted by us to check the graphs/ subgraphs completeness.
Keywords: *Complexity, Graphs, Complex Systems, Emergence, Relationships Graph*

1 Introduction

We consider that an objective reality is that in which about every item we can say that is true or not. A model of the world can be achieved using these true considered elements to mentally represent that external world and thus to define for that a formal model. Relations between elements of the world seen results from some cause. Conceptually that is deduced from the received sensory data from this and from some form of coding these signals in the formal system. The formal system thus constructed is handled by the mind through what we call inference.

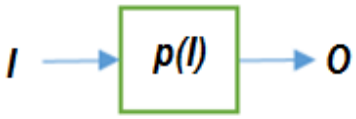

Fig. 1. The Representation of Agents as Transaction

The simplest way to abstractly represent the components of a system is realized by the input-output (I/O) diagram that describes a transition, which can be modelled mathematically as a functional $p:I \rightarrow O$, where each processing box (block) is an Agent that realizes changes on the inputs to obtain outputs (Figure 1).

The processes inside *p(I)* are combinations of perceptions, at receiving inputs, deliberating inside that processes the inputs and decide the actions to be taken, and action processes which are the result having effect on the environment and its internal status, and all those processes can consult the accumulated knowledge and maybe infer new knowledge. By abstracting the Agent with a system then $I=X=\{X_1, X_2, \ldots, X_n\}$ are the possible inputs to the system from its environment and are processed by perception mechanisms, and $O=Y=\{Y_1, Y_2, \ldots, Y_m\}$ are the possible outputs of the system to its environment as actions to be done, represented at a moment in time. By considering the time the evolution of the system will have all the results in time. The computation of all possible statuses denoted by (Σ) of the Agent in Figure 1 will be achieved by representing them as tuples of $n+m$ values $\Sigma=< X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_m >$ where X_i represents an input value $(i=\overline{1,n})$ and Y_j an output value $(j=\overline{1,m})$.

When we build an agent based system these behave as basic blocks of a software system

that uses at least two technologies and concepts used in practice, artificial intelligence (AI) and object-oriented (OO) distributed processing. The resulting software system must have the functionality of mapping inputs into outputs via pre-established processing and is reactive by that it focuses on interactions between components as reaction/ response to stimulus from external world and not as traditional applications where important is the execution flux (this dictates also the predefined reactions. In the "agents society" the agent have the properties of atomicity (it means is no decomposable, indivisible), consistency and isolation (is closed and is not affected by the environment changes) and, durability (it have a permanent effect without any possibility for "roll-back" processes). In this context we can see a system with agents as a network of such interactions and the relationship system do not represent nothing than a special case of transition of that kind. A way to understand natural and artificial systems, and also to model them, is represented by the relationships structure between its composing elements (or parts). In a relational system the outputs of on agent can be inputs for on or many other agents. The transformation denoted as *p()* is a functional component having the property it exists independently of material parts that makes it possible (of course the death/ damaging of material part can result in incapacity of perception of the observer so as scientifically demonstrated recently for the human soul). The architecture of complex systems do not necessitate a central processor or coordinator. The results of recent studies in neurobiology and cognitive psychology exploit the idea that the emergence of consciousness and self consciousness are dependent on the rate of production of "cells meta-assemblies" as neural results of self-reflection. A provision mental state is understood as a global system status that is caused by nonlinear local complex interactions of its parts, but cannot be reduced to such parties [2]. Topological measurements of these kind of complex networks showed a striking similarity to their many other types of networks, but completely different and parts of the objective reality (such as Internet, electrical circuits food chains etc.).

In the human systems the connectivity reflected by that a decision or an action of an individual (group, organization, institution, human system, etc.) will affect all other related individuals in the system. The effect do not have a uniform impact but vary directly with the state of each individual related at a time.

The connectivity applies to the interrelationship of individuals inside a system and also to the kinship human systems. The self-organization in the context of human systems is taken to mean the group of individuals gathering together to perform a specific task.

2 From Local Level to Global Level Organization

Simulation of a multi-agent system of an organization must include both centralized method to determine the path to follow by the system and local methods, individual, to establish the trajectory movement of a single individual.

The centralized methods and from top-to-bottom are those in which the overall evaluation function is used to select / decide the best state of the system in its current neighbourhood.

Local methods are focused on the individual level (a single agent, for example), they are decentralized and bottom-up (bottom-to-top). Through them the locally assessed function is used to determine the motion of a single agent (change of state variable assignment). The Agents are autonomous and decide their behaviour based on local assessment tools and given its own satisfaction or objective (hence all the agents at this level are labelled as selfish or greedy or short-sighted).

The local evaluation function is the way used by system to self-organize in a state that is at the same time solution without having direct high-level control, and this is nothing else but the emergence of multi-agent systems. If the whole system can achieve an overall goal, that means that all the the agents follow their own aim, we can say that emergence

occurred, and the system is self-organized for a global target because here we have no centralized control.

At the global level (full scale), the systems have complex properties and global rules of behaviour/ operation, and the approach to model their parts requires a simple decomposition from top-to-bottom. At the basic level (the scale atomic element, non-decomposable in terms of the analysis/ modelling we want to make) there are many elements that have simple structures and properties and which are governed by simple rules. These parties, when we want to recompose the whole, suffer a multitude of aggregation processes bottom-up, and the resulting system possesses functions not found in basics and these are resulting from emergence

To pass from a local coordination level to a global level of the organization, we must keep in mind that all interactions between agents inside the complex system will tend to a coherent, stable status, until agents adapts (mutually) to each other. This process accelerate generally due to the positive feed-back.

3 Global Dynamics

If we now consider the system as a whole - rather than individual agents - we can see that the system undergoes variation. Self-organization means therefore that the system found an attractor, as for example a part of the state space where it can enter but cannot leave. In this respect, the attractor is a preferred region of global dynamics: surrounding attractor states (the attractor basin) are unstable and will eventually be lost and replaced by states inside the attractor In nonlinear systems must keep in mind that the equation of evolution can provide solutions each of them being nothing but a pattern of behaviour. The attractor for such a system represents an equilibrium position that is unique and describes a time invariant (independent of) situation. The equilibrium state becomes a universal attractor point. The stability is essentially determined by the response of the system to perturbations.

If $X=\{X_1, X_2, \ldots, X_n\}$ is the multitude of system inputs then we will consider them as evolving in time $X_i(t)$ and the perturbation will be highlighted as a stable state of them to moment t to which added the correction, $X_i(t)=X_{i,s}+x_i(t)$, and the system's equation can be defined as:

$\frac{dx_i}{dt} = F_i(\{X_{i,s} + x_i\}, \lambda) - F_i(\{X_{i,s}\}, \lambda)$, by which the time dynamics of system ($\frac{dx_i}{dt}$) is defined by laws of the changing rate of the system (F_i) applied to the snapshot of the system stable state ($X_{i,s}$) to which is highlighted the perturbation (x_i) as a correction (by incrementing or decrementing) of the input and with the condition of considering the changing of the parameters (λ) by the external world to the system (control parameters) [3].

Self-organization means to search new attractors appearing when system is far away of equilibrium state. When in the same space exists more attractors each of them have its own attraction basin containing the state sets from which starting the system goes to a specific attractor.

The coexistence of multiple attractors is common to systems and shows an adapted behaviour able to achieve regulating tasks. For systems based on informational message exchanges we can define the attractor as communication of information and interaction of some kind with another agent with the goal to solve a certain type of problems (in the area of specialization of the agent).

4 The Relationship Structure

For being able to determine the regularities of such systems we must be able to define a graph G = (V, M), for the analysed system. We denote by V={ v_i }, (i=1, 2, …, N) the vertices/ nodes multitude and by L ={ (v_i, v_j) } the multitude of links/ edges, it means the oriented graph connecting the ordered vertices pairs, Λ= (V, L). The edge *{i, j}* is the line starting in the vertices *i* and ending in the vertices *j*. The directed edges are called arcs. Two vertices *i* and *j* are called adjacent if they connected by at least one edge. Each node *i* is characterized by its degree k_i, defined as the number of attached edges. Similarly, we define the input-degree, k_i^I, as the

number of input edges (links) and output-degree, k_i^O , as number of exiting edges, and with respect of formula $k_i = k_i^I + k_i^O$. The sum of degrees of all nodes is an odd number. Depending on the existence/ inexistence of directionality in the graph edges this can be directed, respectively, undirected graph. Some graph G can be used to represent the structure of a system by considering the system's elements (parts) as nodes and the edges its interactions. If N is the number of vertices (nodes) and L is the number of links of the graph then the mean degree (G_m) is $G_m=2L/N$, because each edge is attached to two nodes.

5 The Network Structure

Conventionally a network structure is modelled as a graph G which consists of a set of vertices (nodes) V and a set of edges (and/or arcs) M that we define as unordered pairs of distinct vertices. A path (way) in G from node v_o to node v_e is defined as an alternative sequence of nodes and edges (v_o, m_{-1}, v_1, ... m_e, v_e), where $m_i = \{v_{i-1}, v_i\}$ are the edges that realizes the connection to next nodes with respect that no node can be traversed two or more times. Such a path is not necessarily unique. The length of the path is given by the number of its nodes. The degree of a node x is given by the number of edges containing x, for example the number of its neighbours: $deg(x) = |\{m \in M | x \in m\}| = |\{y \in V | \{x, y\} \in M\}| = |\sigma\{x\}|$, where the notation |A| defines the cardinality (the number of elements) of the set (multitude) A. If between two nodes v_i and v_j exists an edge then the nodes are called adjacent and the adjacency relationship can be quantified by the term a_{ii}=1 and the not adjacent by the term a_{ii}=0, of the adjacency matrix $A_{N, N}$ of the graph G, denoted by *A(G)*.

The degree of node *i* computed as sum of all inputs in the line *i* of the matrix and the total adjacency of a graph as sum over all elements of the matrix, such: $a_i = \sum_{j=1}^{N} a_{ij}$; $A(G) = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij} = \sum_{i=1}^{N} a_i$.

Equivalent, we can define *deg(x)* as the number of edges incident with the node x and we can consider the degree of inputs and outputs. The adjacency of undirected graphs (each edge is seen for both related nodes) is greater than those of directed (because here we consider only the link in the direction indicated by arrow).

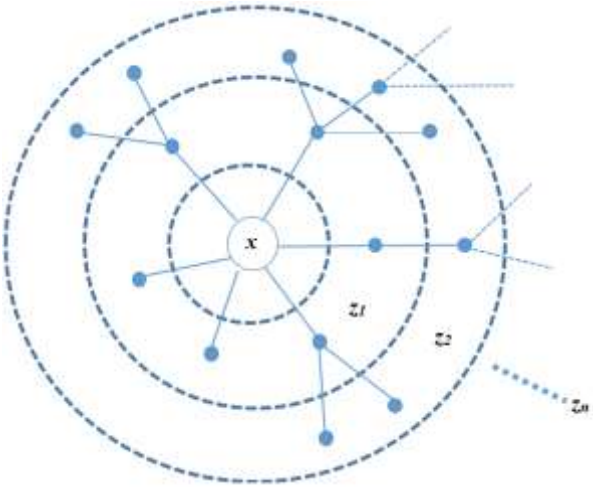

Fig. 2. Nodes Neighbouring

The mean degree of the node denoted by $<a_i>$ and the connectivity (*Conn*) are determined by formulas: $< a_i > = \frac{A(G)}{N}$; $Conn = \frac{A(G)}{N^2} = \frac{2M}{N^2}$.

The distance *d(x, y)* is the length of the shortest path in G connecting the node *x* with node *y*. If a connection path between *x* and *y* don' exist we set $d(x, y) = \infty$. Thus the graph G is connected if and only if *d(x, y)* is finite, $\forall x, y \in V$ ($x \neq y$). The distance $d(v_o, v_e)$ represents the less number of nodes that must be traversed to attain v_e from v_o. This number is an integral if we don't have weighted graphs. The mean length of a path (*l*) in a graph with N nodes is given by the formula: $l = \frac{1}{N(N-1)}\sum_{\forall i,j} d(v_i, v_j)$. In a random graph the mean length *l* rise lower the rise of the graph magnitude. The neighbours of a node can be grouped on categories depending on the number of arcs needed by the path between them, namely primary neighbours (category zone 1, z_1), secondary neighbours (category zone 2, z_2), tertiary neighbours (category zone 3, z_3), etc. The set of the neighbours of some node *x* is $\sigma\{x\} = \{y \in V | \{x, y\} \in M\}$. If we fix some node *x* then we can define its neighbours by grouping them depending on

the distance between them as primary, secondary, tertiary, … neighbours that will be included in the corresponding (suitable) neighbouring zones z_1, z_2, …, z_n (Figure 3). The concentric circles with dotted line have the role to demarcate the neighbouring zones (the levels). This representation is suitable to highlight node adjacency and their degree.

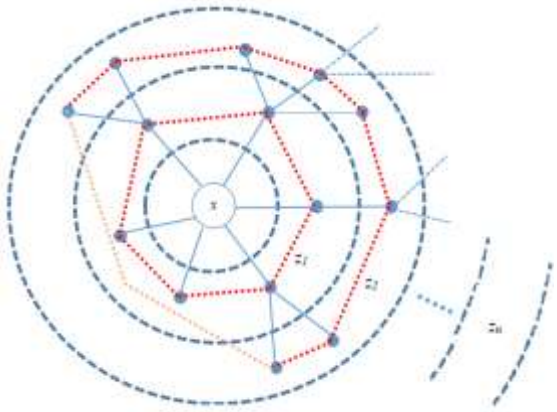

Fig. 3. The Node Neighbors, Edges and Facets

To check and verify the neighbouring completeness of a specific level we propose using the Euler's equation for spheres: if a surface of a sphere is cut into **F** facets with **E** edges and **V** nodes then we have the equality: **V-E+F=2**. This equation can be proof by realizing the subgraph of the interest neighbouring order (as for example in Figure 2) and by adding to this fictitious edges between the nodes of the same level, to form the facets, as illustrated in Figure 3 by the red lines. To level 2 is defined a dashed fictitious broken line only for understanding reasons in the flat representation.

In that way we define the facets as if they obtained by cutting a sphere in which our graph can be inscribed. For each neighbouring zone we number the vertices (nodes), and the edges and arcs and check by the formula. For given example in Figure 3 we have:

- primary neighbors: (z_1): V=1+6=7; E=6+6=12; F=7, hence V-E+F=7-12+7=2
- secondary (z_2): V=1+6+8=15; E=6+6+16=28; F=6+8+1=15, hence V-E+F=15-28+15=2

We can us this representation to analyse the attractors corroborated by measurements about the clustering. This approach can be easy adapted to weighted graphs. We define *deg(x)* as a sum of the weights of all edges containing x and we define the path length as a sum of the weight associated to all edges in the path. According to Amaral et al. (in [4]) the distribution *P(k)* of nodes grades *k = deg(x)* allows the identification of at least three types of networks structurally defined: single-scale, free networks (free) scale, and large-scale networks. The Euler's theorem on graphs relates the number of nodes (vertices) **V**, edges **E**, independent cycles **C** and components **K** of a graph as **C=E-V+K**. To model social networks a great interest represented by the high degree of clustering given by that the friends of a member tend to be friends of all other members. Clustering is low for random graphs. The clustering coefficient of node *i*, denoted by C_i, is defined as the ratio between the number of vertices V_i of primary neighbours of the node i and the maximum number of nodes of the completed subgraph formed starting with its primary neighbours, hence $V_i(max)=a_i(a_i-1)/2$, or $C_i = \frac{2E_i}{a_i(a_i-1)}$.

When C_i is the mean on the entire network, we have: $C = \frac{1}{N}\sum_{i-1}^{N} C_i = \frac{1}{N}\sum_{i=1}^{N} \frac{2E_i}{k_i(k_i-1)}$.

Similarly, we can consider the secondary neighbours and determine the clustering coefficient and so on for other neighbouring level.

6 Analysis and Simulation of Social and Economic Networks

Any kind of social groups or the social/ economic organizations can be represented as multigraph (one agent/ element in the group can have in the same time many roles in the same network and in relationship with the same other agent/ element denoted previously by nodes or vertices) and by associating to the nodes and edges of the obtained graph the probabilities corresponding to each property/ characteristic. We obtain probabilities matrices (Figure 4). The multigraph will be decomposed in simple graphs each one having

an associated probability matrix. The simple graph will be checked by using the two Euler's formula introduced previously.

By considering the time dimension and the multitude of relationships we obtain a three dimension massive composed by the probability matrix in which each panel represents a relationship/ association (suggested by label 1) for which we can compute measurements as mutual entropy, normalized mutual entropy and marginal entropy or the proposed equivalents based on informational energy when we use excerpts. Once obtained the probability matrices they can be used as inputs for the models to commensurate the internal and external complexity (as shown in [5], [6]), for example. This model will be extended and adapted to realize the modelling of the relationships in the general model used to integrate the vary brain and psyche models realize and make cooperate them.

To analyse the social and economic networks the working procedure proposed here involves the following major steps:

1) The knowledge and measurement of the network through formal relations stipulated in laws, statutes, rules of organization and functioning, organization, job descriptions etc.;
2) The knowledge and measurement of the network through informal relationships using statistical tools and other investigations methods;
3) The simulation and the generation of associated network of relations between members and finding the network parameters and mechanisms for forecasting and optimal control.

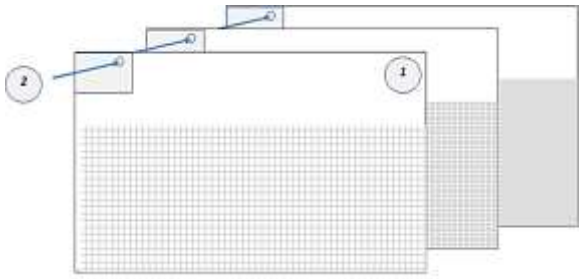

Fig. 4. The Social System Modeled as Network and the Relationships Inside as Probability Matrix [5]

Following is a GEXF (Graph Exchange XML Format, http://gexf.net/format/) description of an organizational graph using formal data retrieved at http://dice.ase.ro/?page_id=509 and informal data about the team groups relationships inside DICE organization:

- LAP, PC, SCO, BDSI, and Info grouped together in Informatics group;
- Cib_I, and Cib_II grouped together in Cybernetics group.

The way the weight was obtained is based on the analysis of relationships between the members of each group which are not included in this description (it is not subject of these paper).

```
<?xml version="1.0" encoding="utf-8" ?>
<gexf version="1.1" xmlns="http://www.gexf.net/1.1draft"
 xmlns:viz="http://www.gexf.net/1.1draft/viz"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance">
 <graph defaultedgetype="directed" mode="static">
  <nodes> <node id="0" label="DICE" />
   <node id="1" label="Informatics" ><nodes><node id="10" label="LAP"/>
       <node id="11" label="PC"/><node id="12" label="SCO"/>
       <node id="13" label="BDSI"/><node id="14" label="Info"></node></nodes>
   </node><node id="2" label="Cybernetics"><nodes><node id="20" label="Cib_I"/>
       <node id="21" label="Cib_II"/></nodes></node></nodes>
  <edges>
        <edge id="0"  source="1"  target="0"  type="directed" weight="7.86"/>
        <edge id="1"  source="2"  target="0"  type="directed" weight="2.14"/>
        <edge id="3"  source="10" target="1"  type="directed" weight="4.25"/>
        <edge id="4"  source="11" target="1"  type="directed" weight="4.15"/>
        <edge id="5"  source="12" target="1"  type="directed" weight="2.65"/>
```

```
        <edge id="6"   source="13" target="1"   type="directed" weight="3.25"/>
        <edge id="7"   source="14" target="1"   type="directed" weight="5.10"/>
        <edge id="8"   source="20" target="2"   type="directed" weight="3.85"/>
        <edge id="9"   source="21" target="2"   type="directed" weight="3.85"/>
        <edge id="10" source="10" target="11" type="mutual"   weight="-2.0"/>
        <edge id="11" source="10" target="14" type="mutual"   weight="-2.0"/>
        <edge id="12" source="11" target="14" type="mutual"   weight="2.00"/>
        <edge id="13" source="13" target="14" type="mutual"   weight="-1.0"/>
        <edge id="14" source="20" target="21" type="mutual"   weight="-3.0"/>
        <edge id="15" source="10" target="12" type="mutual"   weight="1.00"/>
        <edge id="16" source="10" target="13" type="mutual"   weight="1.00"/>
        <edge id="17" source="1"   target="2"   type="mutual"   weight="1.00"/>
        <edge id="18" source="13" target="11" type="mutual"   weight="-1.0"/>
        <edge id="19" source="13" target="12" type="mutual"   weight="-1.0"/>
        <edge id="20" source="12" target="11" type="mutual"   weight="-1.0"/>
        <edge id="21" source="12" target="14" type="mutual"   weight="-1.0"/>
</edges> </graph></gexf>
```

For the simulation and for generate the network of relationships between members within a social group or between social groups we make call on the similarity of such networks with a variety of real networks, such as electricity networks, the Internet, etc. and models developed for them. In what follows are introduced the main features of the Waxman model, BA model [7] and a localized model for dynamic networks [8] and [9]. The finding of the network parameters will be based on the concepts of the stochastic block model [10]. The models described here and the working procedure are intended to allow us to realize a reverse engineering like: we know the network and describe its relationships formally, considering them as obtained by generating them, in that way obtaining the parameters, associated probabilities and other measurements of the network which can be interpreted and used in other complex analysis, later on. The present approach is not expressly interested in generating such random networks, as the introduced models do, but rather in the understanding of existing social and economic networks and finding the parameters that would allow us to simulate. Once the mechanisms and its components are known we can achieve its simulations allows us to control the behaviour of the network, for example, in various situations. Because all these models based on a particular type network, the graph, we will realize first a brief introduction of graph notions and concepts.

7 Models Used to Randomly Generate Graphs

In the following introduced five models each one selected due to its meaning and to be an example for a specific action. Thus the first model (called Waxman) is capable to generate an Internet like network, it means a network with multiple paths between a pair of nodes. The second (called BA for brevity) is for self-organizing networks and the third (called The Localization Model) for localization. The fourth model (called The Stochastic Block Model) is for related groups and permit knowing the parameters to generate randomly a graph for the relationships of that groups. The fifth model, in fact two procedures called one **parameter learning** and the other one **inferring the group assignment**, allows to find out the parameters of graph governed by the Stochastic Block Model.

The **Waxman** model is a stochastic model for generating a network structure similar to Internet by stipulating that the probability of connecting two nodes i and j from i to j is given by $p(i,j) = \propto e^{-d/(\beta L)}$ where α, β are positive (sub)unit values and $0<\alpha, \beta\leq 1$, d is the Euclidean distance from vertices (node) i to vertices j, and L is the maximal distance between two vertices in the network.

BA for brevity [9] is the proposed model of Brabasi and Abert as a model responsible for

self-organization characteristics in real systems. The model considers that initial network has n_o isolated nodes and involves three major operations / actions that are performed at each time point as follows:

1. **Adding new links/ edges between existing nodes**: with probability *p*, *n* nodes, $n \leq n_o$ added to the existing ones. One end of the link of a new node is randomly chosen and the other end is chosen with the probability $p(k_i) = \frac{k_i+1}{\sum_l (k_l+1)}$ (1) where k_i is the degree of the node *i* and $l \in L$ (links).
2. **Re-linking**, by which *m* links are rebuilt with probability *q*: randomly is selected a node *i* and a link l_{ij} attached to node *i* is replaced by a new link $l_{ij'}$ connecting the node *i* with *j'*, with *j'* chosen with the probability $p(k_{j'})$.
3. **Incremental growth**: a new node is added with probability *(1-q-p)* to a node having *m* new links connected to the existing nodes in the network with probability $p(k_i)$.

The probabilities used in the model satisfies the constraints: $0 \leq p \leq 1$, $0 \leq q \leq 1$ and $0 \leq p + q \leq 1$.

The Localization Model [8] and [9] is developed for dynamic networks. The initial network have n_o nodes and l_o links and is generated by the algorithm:

1. From the existing network are selected V nodes and considered as „local-world" for the new added nodes;
2. A new node is added to the network, at each time point, and connected to the *n* existing ones, in their local world determined by (1), with the probability :$p_{local}(k_i) = \frac{N}{n_{o+t}} \frac{k_i}{\sum_{j\ local} k_j}$.

The Stochastic Block Model [10] uses the following parameters:

- g – the number of the groups;
- $\{n_u\}$ – the fraction of total nodes N which belongs to every group u, with $1 \leq u \leq g$;
- An affinity matrix with dimensions *g*x*g* containing the probabilities p_{uv}, that stipulates the existence of an edge between the group *u* and group *v*.

Randomly is generated a directed graph G on N nodes with the associated adjacency matrix A with $a_{ij}=1$ if an edge exists from *i* to *j* and $a_{ij}=0$ if contrary. Each node have an associated label $e_i \in \{1, \dots, g\}$ that indicating to which group belongs to. The labels chosen independently, where for each node the probability that the label be $e_i = u$ is n_u. Between each pair of nodes <*i*, *j*> we include an edge from *i* to *j* with the probability p_{e_i,e_j}, by setting $a_{ij} = 1$ and $a_{ij} = 0$ with probability $1 - p_{e_i,e_j}$. The self loops are ignored and consequently $a_{ii} = 0$.

8 Finding the Parameters Values and Labelling For an Existing Graph

Now we suppose that the network was generated following the stochastic block model, just described in the previous paragraph, and we know the resulted graph G but we don't know the values for the parameters g, n_u, p_{uv} and for the labels e_i. To solve that problem we apply two procedures defined as [10] **parameter learning** and **inferring the group assignment**.

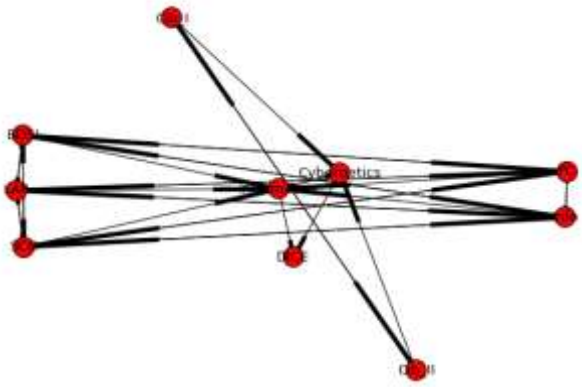

Fig. 5. The Graph Corresponding to the GEXF Description

In order to be able inferring the group assignment an agreement is defined between the original assignment $\{e_i\}$ and its estimate $\{f_i\}$. The probability that the stochastic block model generate the graph G having the adjacency matrix A, together with the group associations $\{f_i\}$, and conditioned on the parameter $\theta = \{g, \{n_u\}, \{p_{uv}\}\}$ is

$P(G,\{f_i\}|\theta) = \prod_{i\neq j}\left[p_{f_i'f_j}^{a_{ij}}\left(1 - p_{f_i,f_j}\right)^{1-a_{ij}}\right]\prod_i n_{f_i}$, where the product is on the pairs $i < j$ for the undirected case.

The following is a small script in Python using the framework Networkx [11] that reads the GEXF description of the graph which is illustratively described and analysed from different points of views using the graph theory and indicators. The graph is shown visually in Figure 5. The output of this illustrative sequence of instructions is obtained as effect of numbered print() commands.

```
import networkx as nx; import numpy as np; import scipy as xsp
import networkx.algorithms.approximation as alg;
import networkx.algorithms.components.attracting as att
import matplotlib.pyplot as plt; import sys
path="D:\__myPython\\"; ret=raw_input("Type GEXF File Name: ")
path+=ret; print "\r\n 0)- The requested file is: "+path
fis=open(path,'rb'); bl=nx.read_gexf(path)
print "\r\n 1)- List of nodes read in the external file\r\n",; print(list(bl))
print "\r\n 2)- The list of lines in the external file, line by line\r\n",
for line in bl:;     print line
V=nx.relabel_gexf_graph(bl); print "\r\n 3)- Renaming graph <bl> in V: \r\n",;print("\r\n 3.1)-
Nodes in V: \r\n"),; print(V.nodes()); print("\r\n 3.2)- Edges in V: \r\n"),; print(V.edges())
fis.close()
# Prepare canvas and drawing layout
print("\r\n 4)- Prepare canvas and drawing layout \r\n"),
nx.spectral_layout(V); nx.draw(V,with_labels=True); plt.show()
# convert directed graph to an undirected copy for analysis
UG=nx.DiGraph.to_undirected(V)
print("\r\n 5)- Attractor components: \r\n"); print(list(att.attracting_components(V)))
print("\r\n 6)- CLIQUES of different degrees \r\n")
for i in range(2,len(UG),1):; print("%s %d %s\r\n"%("Cliques of ",i," or more nodes:"))
    print(list(nx.k_clique_communities(UG, i)))
# find attracting components subgraphs
rg=att.attracting_component_subgraphs(V,copy=True)
print("\r\n 7)- List of subgraphs of attracting components: \r\n")
print list(list(rg)); for k in rg:;    print k,
print("\r\n 8)- Triangle Components: \r\n"); print(nx.triangles(UG))
print("\r\n 9)- Tranzitivity Coefficient: \r\n"); print(nx.transitivity(UG))
print("\r\n 10)- Clustering Coefficients: \r\n"); print(nx.clustering(UG))
print("\r\n 11)- Average Clustering Coefficient: \r\n"); print(nx.average_clustering(UG))
print("\r\n 12)- Square Clustering Coefficients: \r\n"); print(nx.square_clustering(UG))
print("\r\n 13)- Min weighted dominating set:
\r\n");print(alg.min_weighted_dominating_set(UG))
print("\r\n 14)- Cardinality min edge dominating set:
\r\n");print(alg.min_edge_dominating_set(UG))
print("\r\n 15)- Maximum independent set: \r\n"); print(alg.maximum_independent_set(UG))
A=nx.adjacency_matrix(UG); print("\r\n16)- Adjacency matrix: "); print(A.todense())
print("\r\n 17)- Adjacency list: ")
Al=nx.generate_adjlist(UG, delimiter=' '); for line in Al:;  print(line); print(list(Al))
```

The output of the previous instructions sequence is:

Type GEXF File Name: dice.gexf
The requested file is: D:__myPython\dice.gexf
1)- List of nodes read in the external file ['11', '10', '13', '12', '20', '14', '1', '0', '2', '21']
2)- The list of lines in the external file, line by line 11 …… 21
3)- Renaming graph <bl> in V:
3.1)- Nodes in V: ['Info', 'SCO', 'DICE', 'Cib_I', 'Cib_II', 'PC', 'BDSI', 'Cybernetics', 'Informatics', 'LAP']
3.2)- Edges in V: [('Info', 'Informatics'), ('Info', 'PC'), ('Info', 'BDSI'), ('Info', 'LAP'), ('Info', 'SCO'), ('SCO', 'Informatics'), ('SCO', 'Info'), ('SCO', 'BDSI'), ('SCO', 'LAP'), ('SCO', 'PC'), ('Cib_I', 'Cib_II'), ('Cib_I', 'Cybernetics'), ('Cib_II', 'Cybernetics'), ('Cib_II', 'Cib_I'), ('PC', 'Informatics'), ('PC', 'Info'), ('PC', 'BDSI'), ('PC', 'LAP'), ('PC', 'SCO'), ('BDSI', 'Informatics'), ('BDSI', 'Info'), ('BDSI', 'SCO'), ('BDSI', 'LAP'), ('BDSI', 'PC'), ('Cybernetics', 'Informatics'), ('Cybernetics', 'DICE'), ('Informatics', 'Cybernetics'), ('Informatics', 'DICE'), ('LAP', 'Informatics'), ('LAP', 'PC'), ('LAP', 'BDSI'), ('LAP', 'Info'), ('LAP', 'SCO')]
4)- Prepare canvas and drawing layout (result in Figure 5)
5)- Attractor components: [['DICE']]
6)- CLIQUES of different degrees
Cliques of 2 or more nodes:[frozenset(['Info', 'SCO', 'DICE', 'Cib_I', 'Cib_II', 'PC', 'BDSI', 'Cybernetics', 'Informatics', 'LAP'])]
Cliques of 3 or more nodes:[frozenset(['Informatics', 'Cybernetics', 'DICE']), frozenset(['Cib_II', 'Cybernetics', 'Cib_I']), frozenset(['Info', 'PC', 'SCO', 'Informatics', 'LAP', 'BDSI'])]
Cliques of 4 or more nodes:[frozenset(['Info', 'PC', 'SCO', 'Informatics', 'LAP', 'BDSI'])]
Cliques of 5 or more nodes:[frozenset(['Info', 'PC', 'SCO', 'Informatics', 'LAP', 'BDSI'])]
Cliques of 6 or more nodes:[frozenset(['Info', 'PC', 'SCO', 'Informatics', 'LAP', 'BDSI'])]
Cliques of 7 or more nodes:[]
7)- List of subgraphs of attracting components:
[<networkx.classes.digraph.DiGraph object at 0x0000000012F90668>]
8)- Triangle Components: {'Info': 10, 'SCO': 10, 'DICE': 1, 'Cib_I': 1, 'Cib_II': 1, 'PC': 10, 'BDSI': 10, 'Cybernetics': 2, 'Informatics': 11, 'LAP': 10}
9)- Tranzitivity Coefficient: 0.825
10)- Clustering Coefficients: {'Info': 1.0, 'SCO': 1.0, 'DICE': 1.0, 'Cib_I': 1.0, 'Cib_II': 1.0, 'PC': 1.0, 'BDSI': 1.0, 'Cybernetics': 0.3333333333333333, 'Informatics': 0.5238095238095238, 'LAP': 1.0}
11)- Average Clustering Coefficient: 0.885714285714
12)- Square Clustering Coefficients: {'Info': 1.0, 'SCO': 1.0, 'DICE': 0.0, 'Cib_I': 0.0, 'Cib_II': 0.0, 'PC': 1.0, 'BDSI': 1.0, 'Cybernetics': 0.0, 'Informatics': 0.2727272727272727, 'LAP': 1.0}
13)- Min weighted dominating set: set(['Cib_II', 'BDSI', 'Cybernetics', 'Cib_I'])
14)- Cardinality min edge dominating set: set([('Cib_I', 'Cib_II'), ('DICE', 'Cybernetics'), ('Info', 'Informatics'), ('PC', 'LAP'), ('SCO', 'BDSI')])
15)- Maximum independent set: set(['LAP'])
16)- Adjacency matrix:

```
[[ 0.   -1.    0.    0.    0.    2.   -1.    0.    5.1  -2.  ]
 [-1.    0.    0.    0.    0.   -1.   -1.    0.    2.65  1.  ]
 [ 0.    0.    0.    0.    0.    0.    0.    2.14  7.86  0.  ]
 [ 0.    0.    0.    0.   -3.    0.    0.    3.85  0.    0.  ]
 [ 0.    0.    0.   -3.    0.    0.    0.    3.85  0.    0.  ]
 [ 2.   -1.    0.    0.    0.    0.   -1.    0.    4.15 -2.  ]
 [-1.   -1.    0.    0.    0.   -1.    0.    0.    3.25  1.  ]
 [ 0.    0.    2.14  3.85  3.85  0.    0.    0.    1.    0.  ]
 [ 5.1   2.65  7.86  0.    0.    4.15  3.25  1.    0.    4.25]
 [-2.    1.    0.    0.    0.   -2.    1.    0.    4.25  0.  ]]
```

17)- Adjacency list: Info Informatics PC BDSI LAP SCO … []

9 Conclusions

The paper is the result of many efforts (partly described in [5], [6], [12], [13]) in finding a pragmatic way to measure the external and internal complexity of a system, efforts and findings that are subjects to other articles of the authors, as referenced. Here we found several algorithms that allows to realize a reverse-engineering like operations on existing real or artificial networks and to determine the probabilities associated to different elements. The networks are studied and investigated using existing methodologies and tools (such as the ones used in the analysis stage for information systems) and later described in terms of nodes and edges using an existing framework such as NetworkX, exemplified here, and then obtaining the figures for the measurements of interest. For large networks, where the usage of excerpts is inherent, the measurements for internal and external complexity can use the versions of formulas based on informational energy.

References

[1] K. Mainzer, (2007) Thinking in Complexity - The Computational Dynamics of Matter, Mind, and Mankind, ISBN 978-3-540-72228-1, Springer.

[2] H. Lin, (2007) Architectural Design of Multi-Agent Systems: Technologies and Techniques, IGI Global.

[3] J. Moffat, (2003) Complexity Theory and Network Centric Warefare, CCRP Publication Series.

[4] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley, (2003), "Classes of small-world networks", Applied Physical Sciences, Proceedings of the National Academy of Sciences USA, doi: 10.1073/pnas.200327197, pp. 11149–11152.

[5] D. Rizescu and V. Avram, (2013). Using Onicescu's Informational Energy to Approximate Social Entropy. Procedia - Social and Behavioral Sciences, Volume 114, 21 February 2014, ISSN 1877-0428, pp. 377-381.

[6] V. Avram and D. Rizescu, (2014) Measuring External Complexity of Complex Adaptive Systems Using Onicescu's Informational Energy, Mediterranean Journal of Social Sciences, Vol. 5 No.22, August 2014, Rome-Italy, ISSN 2039-2117, DOI:10.5901/mjss.2014.v5n22p407, pp. 408-417.

[7] X. Li, and G. Chen (2003) A local-world evolving network model, Physica A: Statistical Mechanics and its Applications, Volume 328, Issues 1–2, 1 October 2003, Pages 274–286, Elsevier.

[8] G. Chen, Z. Fan, and X. Li, 2005. Modelling the Complex Internet Topology in L. Kocarev and G. Vattay (Eds.): Compl. Dyn. in Com. Networks, UCS 5, pp. 213–234, Springer-Verlag Berlin Heidelberg 2005.

[9] AL. Barabási, R. Albert and H. Jeong, 1999. Mean-field theory for scale-free random networks, Physica A: Statistical Mechanics and its Applications 272 (1), 173-187.

[10] A. Decelle, F. Krzakala, C. Moore and L. Zdeborová, 2011. Asymptotic Analysis of the Stochastic Block Model for Modular Networks and its Algorithmic Applications, Santa Fe Institute, SFI WORKING PAPER: 2011-09-048, http://www.santafe.edu.

[11] A. Hagberg, D. Schult and P. Swart. 2012. NetworkX Reference Release 1.7, https://pypi.python.org/pypi/networkx/.

[12] D. Rizescu and V. Avram. 2013. Using Onicescu's Informational Energy to Approximate Social Entropy. Procedia - Social and Behavioral Sciences, Volume 114, 21 February 2014, ISSN 1877-0428, pp 377-381.

[13] V. Avram and D. Rizescu. 2014. Measuring External Complexity of Complex Adaptive Systems Using Onicescu's Informational Energy, Mediterranean Journal of Social Sciences, Vol. 5 No.22, August 2014, Rome-Italy, ISSN 2039-2117, DOI:10.5901/mjss.2014.v5n22p407, pp. 408-417.

[14] J. Jost. 2004. External and internal complexity of complex adaptive systems. Theory in Biosciences Volume 123, Issue 1, 25 June 2004, 69–88.

[15] O. Onicescu and V. Stefanescu. 1979. Elemente de statistica informationala cu aplicatii/ Elements of informational statistics with applications. Editura Tehnica, 11-31.

Vasile AVRAM PhD has graduated the Faculty of Economic Computation and Economic Cybernetics in 1976; PhD - 1995. Is professor of Internet Technologies for e-Business and Informatics for Business Administration at Faculty of Business Administration from the Bucharest University of Economic Studies, Romania. His research interests mainly include Internet Technologies, Technologies for e-Business, Database Management Systems, Computer Programming, and Knowledge Management. He is member of the professional associations IEEE, ACM, INFOREC, and IACSIT.

Diana RIZESCU (AVRAM) PhD has graduated the Faculty of Mathematics of the Bucharest University in 1995, the SIMPRE master program of the Faculty of Cybernetics Statistics and Informatics in Economy of Bucharest University of Economic Studies in 2012, and the faculty of Psychology from "Titu Maiorescu" University in 2015. Is Analyst at SIGMA Publishing Co Romania. His research interests mainly include Internet Technologies, Mathematical Modelling, Knowledge Management, and Psychology.