MuNeG - The Framework for Multilayer Network Generator

Adrian Popiel, Przemysław Kazienko, Tomasz Kajdanowicz
Department of Computational Intelligence, Wroclaw University of Technology
Wybrzeże Wyspiańskiego 27, 50- 370 Wrocław, Poland
Email: {adrian.popiel, kazienko, tomasz.kajdanowicz}@pwr.edu.pl

***Abstract*—It is a common problem that cost of extracting data for network analysis could be very high. Also sometimes in the Internet is it hard to find graph with desired features such as node degree or clustering level. Because of that graph generators can than be very helpful. In the past bunch of models of such generators was developed: random graphs, small worlds and scale free networks. All of these generators were developed to quickly and efficiently create networks with desired parameters. However all of this models produce single layer graphs. Domain of multiplexes or multilayer graphs has not already been so deeply analysed, also because it is hard to collect multilayer data among real datasets or there is hard to define what kind of information layers exactly should represent. Proposed MuNeG — Multilayer Network Generator can produce, based on set of input parameters, multiplex networks - networks where each node has its counterpart in each layer. The carried out experiments proved that MuNeG graphs have different network and social parameters depends on input values. This feature gives user a very handful tool to generate multiplex networks on purpose of social network or complex network analysis. Generator features, input parameters and their influence on so called graph theory measures such as: node degree, average shortest path, diameter or clustering are described in the following article.**

Keywords*—Complex Networks, Network Generation, Synthethic Dataset, Collective Classification

I. Introduction

Extracting network data can and very often is a very complex and expensive task. Additionally once extracted network has constant network features like: distribution of node degree, number of triangles or clustering coefficient. It is known that networks from different domains show different graph properties. It is known problem to have a variety of graphs to test newly designed methods in many environments, especially when method is designed to be context free. Because of that, methods of network generation gain recently more and more followers.

One of the recent most popular problem is to analyse multilayer graphs. In reality multilayer networks are represented by complex structures where data comes from different sources. In this kind of networks, layer is represented by each source. In social networks layers can represent types of social relations like: family, friends, co-workers. Special types of multilayer graph is a multiplex. Multiplex is a structure in which each node has its counterpart in each layer. Nodes are connected only within layer, connections between layers exist only between couterparts [1].

Among the existing network generators, it is far from seen once that also generates labeled nodes for collective classification task in multilayer data. One published in [2] provides flexible tool for generating heterogeneous networks with classes, but this method is still single layer.

In this paper new multiplex network generator - MuNeG has been proposed. MuNeG can generate a variety of complex networks. It can be configured by a set of parameters, what makes this method very flexible. The carried out experiments prove that generation parameters have a significance influence on network and social features of the graph, which makes MuNeG a comfortable and context free tool.

II. Related Work

This work is strongly connected to the theory of random graphs. This topic initially was analysed by Erdős and Rényi [3]. In their approach graph is generated by placing random undirected edges between nodes. This model produces a graph where node degree comes from binomial distribution and number of edges is strongly dependent on defined number of nodes.

Degree distribution as an input for random graph generation was proposed in configuration model [4]. In this approach at the beginning each node becomes set of initially not connected edges. The number of edges comes from defined degree sequence. Last step is to randomly connect nodes to "fulfil" degree requirements. This model can generate every graph from a given degree sequence with same probability.

Another group of models are small worlds. Approach proposed by Watts and Strogatz [5] puts given number of connected nodes on a ring. Each node is connected to k nearest neighbors. Than with certain probability edges are chosen to "rewired" process. First end of the edge stays unchanged, but second end(new node) is chosen uniformly. Such generated small worlds have small average shortest-path and high clustering coefficient, which make this networks similar to real-world social networks.

Last group of random graph models are scale-free networks. Such networks have a power-law degree distribution. Method proposed by Barabási und Albert [6] generates growing network. Algorithm starts with a small network, than new nodes are added to graph and they are being connected to existing nodes with probability proportional to current node degree. This kind of network tends to produce hubs—nodes with a very high degree. Degree distribution of such nodes comes from power-law and average shortest-path is growing logarithmically.

As we can see, already many flexible and complex graph models have been proposed. However there is a very limited set of generators that produce multiplex networks labeled nodes for classification task. Model proposed by Eldardiry and Neville [2] produces graph with labels but this method is still only single layer.

In this work new multiplex network generator with node labeling will be presented and described.

III. MuNeG

MuNeG—Multilayer Network Generator is a flexible tool which in one-liner generates multilayer networks with binary labels. This labeling can be then (and successfully was [7]) used in collective classification [8].

Multiplex networks are special cases of multi-layer networks. Each node in multiplex has its counterpart in each layer [1]. Multiplexes can be described as a extension of classic graphs. Flat graph can be represented as pair $G = (V, E)$, where V is set of nodes and E is set of edges between nodes. In the directed and weighted version of such graph, an edge $\forall e_{ij} \in E : e_{ij} = (v_i, v_j, w_{ij}), v_i, v_j \in V, v_i \neq v_j$ and $w_{ij} \in \mathbb{R}$ is a weight of edge v_{ij}. All nodes $v_i \in V$ are of the same type and there exists only one edge from e_{ij} from v_i to v_j. In multiplex case network is a tuple $MG = (V, V^L, E, EL, L)$, where L is a set of distinct layers, each node $v_i \in V$ has its own representation at each layer $l \in L$ such that $v_{il} \in V^L, e_{ijl} = (v_{il}, v_{jl}, l, w_{ijl})$. Node representations v_{il} from one layer l together with their edges, in fact, form an uniplex network $G = (V, E)$. Typically, layers represent different source of relations. In social networks they can represent types of relationships between humans, e.g. one layer corresponds to friendships from Facebook whereas another to professional links between co-workers and collaborators from LinkedIn, see Fig. 1.

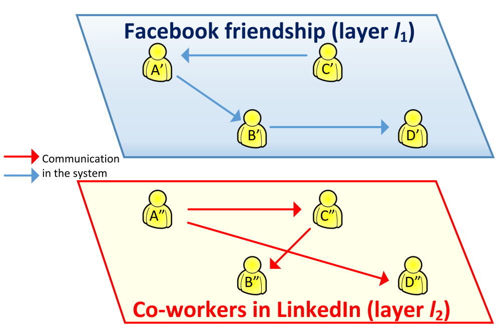

Figure 1. A multi-layer social network with two layers

MuNeG is an enhancement of Eldardiry model [2] to the space of multiplexes. Generator is controlled by six parameters:

- Number of nodes-N^V
- Number of groups-N^{Gr}
- Group homophilly - p_{Gr}
- Probability that two nodes from same group are connected - p_{in}
- Probability that two nodes from different groups are connected - p_{out}
- Number of layers - L

Before the detailed definitions of probabilities p_{Gr}, p_{in} and p_{out}, it is necessary to define some markings. Class label C for each node has value from set $\{0, 1\}$ As it was said MuNeG can generate binary classes. Nodes in generated network are collected in groups. Such collections represent constructions similar to communities in social networks. Nodes in groups are labeled alike to each other. Similarity between nodes in group is related to group homophilly(p_{Gr}) described below. Set $T = \{red, blue\}$ contains possible group names. Each of N^Gr group is either red or blue. Group color labeling Gr_i is randomly assign with same probability:

$$p(Gr_i = blue) = p(Gr_i = red) = 0,5 \tag{1}$$

Homophilly is a measure how much nodes in social groups tend to be similar to each other. In network generator probability p_{Gr} which represents group homophilly is used semantically as measure how probable is same labeling within group. It can be represented as:

$$p_{Gr} = p(C = 0|Gr = red) = p(C = 1|Gr = blue) \tag{2}$$

Probability p_{in} is a conditional likelihood of edge existence between nodes within group. Notation $E = 1$ means that edge exists. This graph parameter can be represented as:

$$p_{in}(E_{ij} = 1) = p(E = 1|Gr_i = Gr_j) \tag{3}$$

Last parameter p_{out} is a conditional probability that edge between nodes from different groups exists. It can be defined similar to previous parameters:

$$p_{out}(E_{ij} = 1) = p(E = 1|Gr_i \neq Gr_j) \tag{4}$$

With this set of parameters, MuNeG generation algorithm can be presented in following pseudo code:

Algorithm 1 MuNeG

for each group $g, 1 <= g <= N^{Gr}$ **do**
  Choose a group color from $p(Gr)$ (Eq 1)
end for
for each node $n, 1 <= n <= N^V$ **do**
  Choose uniformly group assignment Gr_n
  Choose a class C_n from p_{Gr} — $p(C|Gr_n)$ (Eq 2)
end for
for each layer l, $1 <= l <= L$ **do**
  for each node i, $1 <= i <= N^V$ **do**
    for each node j, $i <= j <= N^V$ **do**
      if $Gr_i = Gr_j$ **then**
        Choose if edge exists in layer l from p_{in} — $p(E|Gr_i = Gr_j)$ (Eq 3)
      else
        Choose if edge exists in layer l from p_{out} — $p(E|Gr_i \neq Gr_j)$ (Eq 4)
      end if
    end for
  end for
end for

In figure 2 result of MuNeG algorithm is presented. This network was generated with following parameters:

- 200 nodes
- 5 groups(3 reds and 2 blues)
- 50% group homophilly
- 70% probability of edge existence within group
- 10% probability of edge existence between groups
- 1 layer

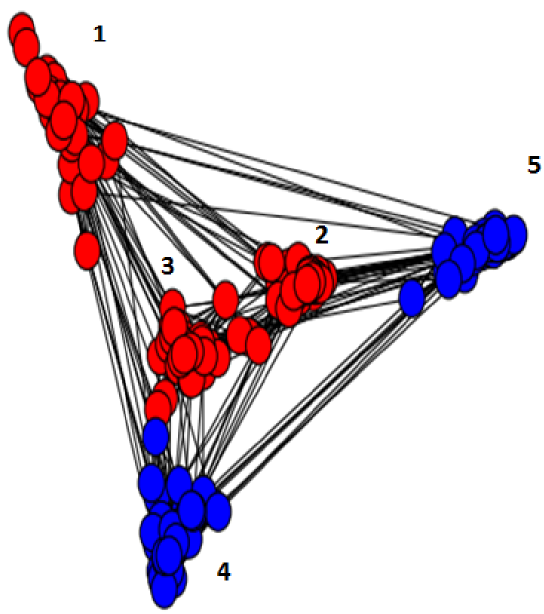

Figure 2. Graph generated by MuNeG - Groups are marked with numbers

In the figure it is clearly showed that generator parameters have strong influence on features of the generated graph. It can be seen that graph has 5 separate groups. Nodes within groups are much more "squeezed" — there are more connections inside groups, what comes from 70% probability of connections. It can also be observed that there are much less connections between groups, because probability of such connections is only on 10% level.

MuNeG is an open-source graph generator implemented in Python [9]. Because of issues, that Python has with "for" loops, code was cythonized [10], what leads to performance gain. Whole code is available in Github: https://github.com/Adek89/multiplex/tree/master/MuNeG.

In the next section parameters of the model, based on generated graphs will be analysed.

IV. Properties of the model

Depending on domain from which network is derive, graph can have different properties. Is is easy to imagine that social network can have different features that network of web pages or network of sensors. To analyse MuNeG generated graphs features, some popular measures from graph theory and social network analysis were chosen:

- Node degree
- Number of edges
- Clustering coefficient
- Number of triangles
- Average shortest path(ASP)
- Diameter

In figures 3-20 selected properties of the model were presented depends on graph generation features. Experiments shows that networks generated by the model can be used in many domains to simulate real graphs.

A. Node degree and number of edges

The node degree($deg(v)$) represents number of edges which are connected to the node(vertex). Degree has a big influence on graph and node parameters. Worthy of notice are following features [11]:

- In most graphs node degree is not equal. Each graph has maximum($\Delta(G)$) and minimum($\delta(G)$) degree.
- If in graph each node has equal degree, then graph is named regular and degree is analysed in terms of graph, not node feature.
- Isolated are these vertices, with degree 0
- Nodes with degree 1 are named leafs or end nodes. Edge connected to such vertex is named pendant
- When in each subgraph of a graph exists node with degree at most k, then such parent graph is called k-degenerated graph

Degree sequence is a non-increasing sequence of node degrees [12]. However such sequence in general do not identify real graph. It means that there are another graphs, that can have same degree sequence. Not all sequences of decreasing positive integers can be a degree sequence of a real graph. From degree sum formula [13]:

$$\sum_{v \in V} deg(v) = 2|E| \tag{5}$$

it is known that if sum is odd then sequence cannot be a degree sequence of a real graph. Converse is also not straightforward. If sum of sequence is even that from such sequence multigraph can always been constructed, but finding of simple graph from even sum is more challenging. This problem is named as graph realization problem [14], another problem of finding or estimating all graphs from degree sequence is a problem from graph enumeration domain [15].

Degree distribution is a probability distribution of nodes degrees over the analysed network. Such distribution is very important in social network analysis and is very helpful on studying real and theoretical networks. For example earlier mentioned random graphs have binomial or Poisson degree distribution [4]. However real network distributions are different. Real networks have distribution where majority of nodes have low degree, but there exist few nodes with very high value(hubs). As it can be imagined, distribution is skewed to the side of low degrees. Some social network are approximately similar to power-law represented by scale-free networks of Barabási [6].

The carried out experiments produced 300000 generated graphs from MuNeG. Research showed strong influence of generator parameters on generated model. In experiments following configurations(combinations) were tested:

- $N^V = \{100, 500, 1000\}$
- $N^{Gr} = \{2, 3, 4, 5, 6, 7, 8, 9\}$
- $p_{Gr} = \{50\%, 55\%, 60\%, 65\%, 70\%, 75\%, 80\%, 85\%, 90\%, 95\%, 100\%\}$
- $p_{in} = \{50\%, 60\%, 70\%, 80\%, 90\%\}$
- $p_{out} = \{1\%, 5\%, 10\%, 20\%, 30\%, 40\%, 50\%\}$
- $L = \{2, 3, 5, 6, 8, 10, 13, 21\}$

All results presented below are averaged numbers among all experiments. In Fig. 3 it is shown that number of generated

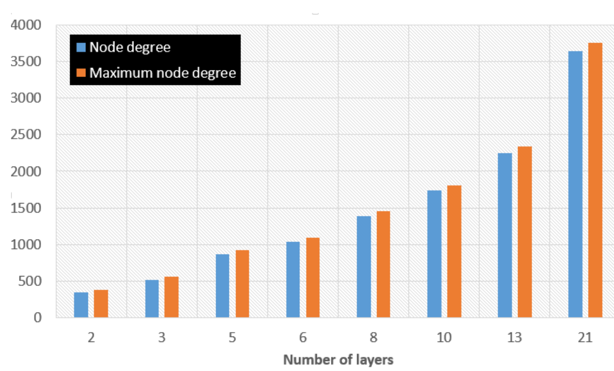

Figure 3. [**Node degree**] Number of layers

layers is proportional to node degree. This result can be simply interpreted. Each node has its counterpart in each layer and according to pseudo code each edge existence is calculated in each layer separately. That leads to conclusion that node degree in each layer is similar(it comes from same distribution). Number of edges is also layer number depending(see fig. 5). Reason for such behaviour is the same like in fig. 3

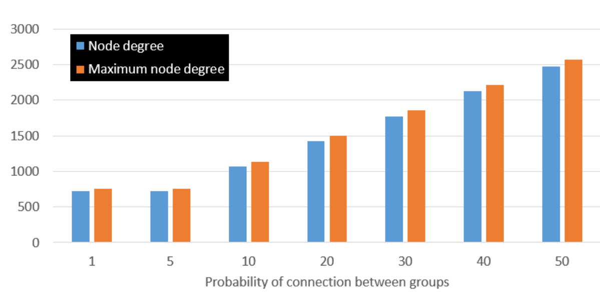

Figure 4. [**Node degree**] Probability of edge existence betweeen groups

In fig. 4 very interesting feature is shown. On horizontal axis probability of edge existence between groups is presented. As it can be observed, node degree is increasing when probability of such existence is $>= 10\%$. By $< 10\%$ node degree is stable. So to generate simple graphs with small node degree, such connection probability should be low.

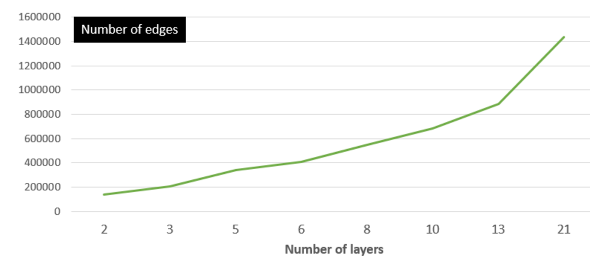

Figure 5. [**Number of edges**] Number of layers

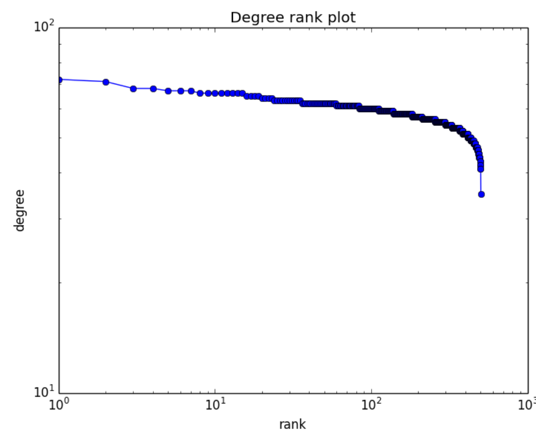

Figure 6. Degree rank plot of exampled graph

In fig. 6 degree sequence of exampled graph(500 nodes, 9 groups, 50% probability of edge existence within group, 1% probability of edge existence between groups, 2 layers) was plotted. As it can be seen degree distribution in this case is similar to normal, but with small standard deviation.

B. *Clustering coefficient and number of triangles*

Clustering coefficient is a measure of a node tendency to cluster together with other nodes. There are two types of clustering coefficient: global and local. Global measure is a feature of the graph and is dependent on number of triangles in whole graph. To define global clustering coefficient term of triplet should be defined. Triplet is a set of three nodes connected by 2(open triplet) or 3(closed triplet) edges. Global clustering can be defined as:

$$C = \frac{3 * number\_of\_triangles}{number\_of\_connected\_triplets} \quad (6)$$

Measure is also named as transitivity of the graph [16]

In fig. 7 number of triangles depending on number of layers in generated multiplex is presented. This feature is directly connected with method of edge generation. Each edge is in layer generated independently.

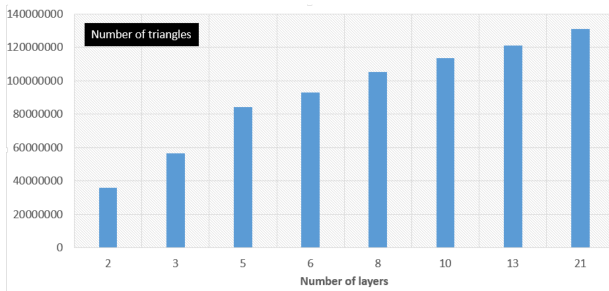

Figure 7. [**Number of triangles**] Number of layers

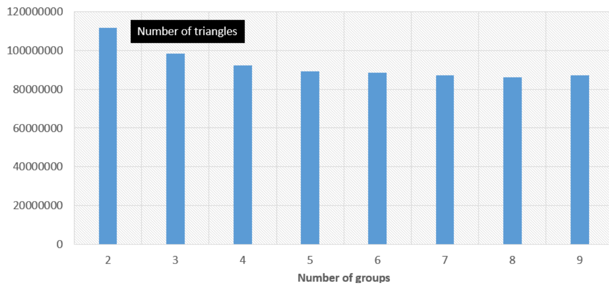

Figure 8. [**Number of triangles**] Number of groups

In fig. 8 it can be observed that increasing number of groups leads to lower number of triangles. It proves that generated groups tend to be more independent when there are lower number of nodes inside.

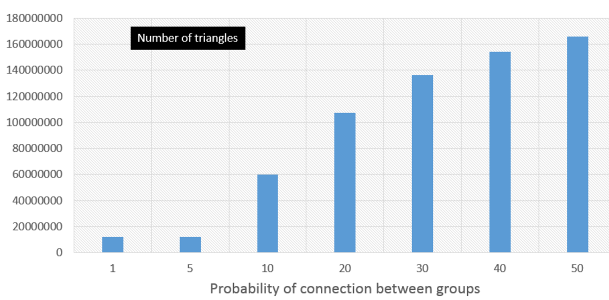

Figure 9. [**Number of triangles**] Probability of edge existence between groups

In fig. 9 probability of connection between groups was drawn on horizontal axis, while number of triangles is presented on vertical one. Results are similar to fig. 4. When probability is lower than 10% graphs are very similar. Also number of triangles is almost equal, than when probability is greater than 10% feature of number of triangles is increasing dramatically(almost 5 billions difference!).

Local clustering coefficient is a node measure, how strong its neighborhood tends to be a clique. Measure was introduce by Watts and Strogatz by checking if network is a small world [5]. It can be calculated from number of triangles through node and node degree:

$$c_v = \frac{2 * T(v)}{deg(v)(deg(v) - 1)} \tag{7}$$

, where $T(v)$ is a number of triangles through node v. As it was mentioned this local measure have a big influence of a small-world differentiation from random graph. Small-worlds have bigger clustering than random graph built from same node set, while both constructions have same average shortest path length.

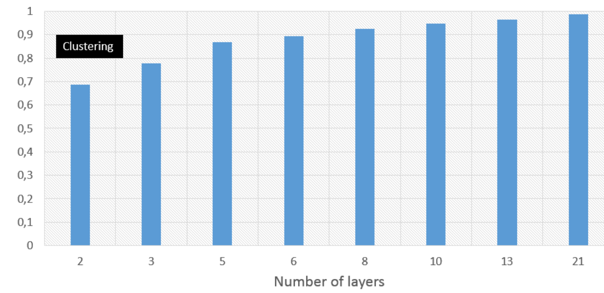

Figure 10. [**Clustering**] Number of layers

MuNeG generated graph present in average high clustering(> 0.6). But this measure is also depending on generation parameters. Similar to number of triangles, clustering is increasing by increasing number of layers(fig. 10) and decreasing by decreasing number of coloured groups(fig. 11).

Once more probability of edge existence between groups gives the most interesting outcome(fig. 12). By probability smaller than 10% clustering has the biggest value(near 1), then by probability exactly 10% turn up sudden decrease. From that point clustering is slowly growing up and finally by 50% is equal once more almost to 1. It can be explained in following manner. By probability lower than 10% in the network exists big clustering within the groups. Then after 10% clustering outside groups takes over the domination in the network.

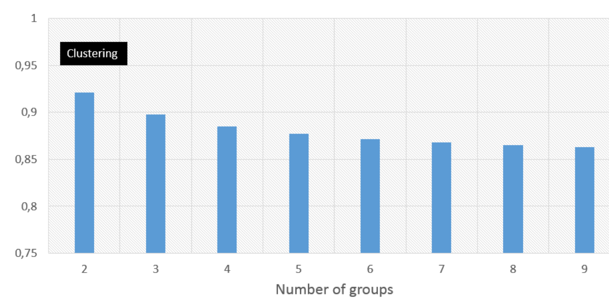

Figure 11. [**Clustering**] Number of groups

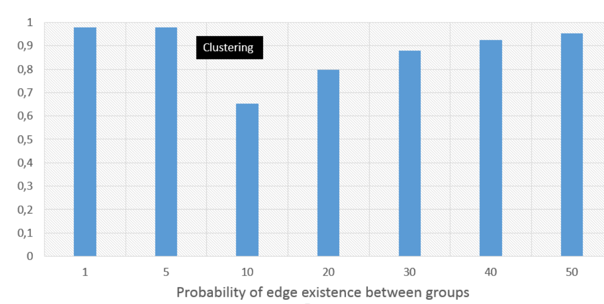

Figure 12. [**Clustering**] Probability of edge existence between groups

C. Average shortest path

Average shortest path is another measure that describes topology of the network. It represents average path length along all shortest paths between all nodes in graph. More formally it can be defined as:

$$a = \sum_{v,k \in V} \frac{d(v,k)}{n(n-1)} \tag{8}$$

, where $d(v,k)$ is a length between nodes v and k and n is number of nodes in the graph. Semantically average shortest path characterises networks that supports short and fast transfer of information like closely bind nodes in sensor network or small worlds in social network domain [17]. Because of that this parameter is one of the most important in all generators from random graph, through Watts and Strogatz model and in Barabási scale free networks. Although big influence on model, average shortest path is independent on number of nodes in the graph, what also will be presented in MuNeG charts. On fig.

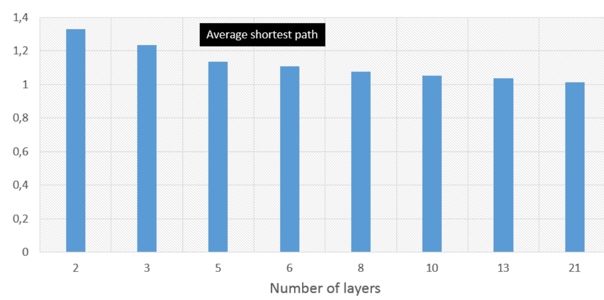

Figure 13. [**ASP**] Number of layers

13-16 results of average shortest path along graphs generated from MuNeG. As is can be observed values of ASP are small: $1.0 < ASP < 1.5$. Because of that MuNeG generated graphs are good to simulate close connected complex networks or in social domain they are more similar to small worlds.

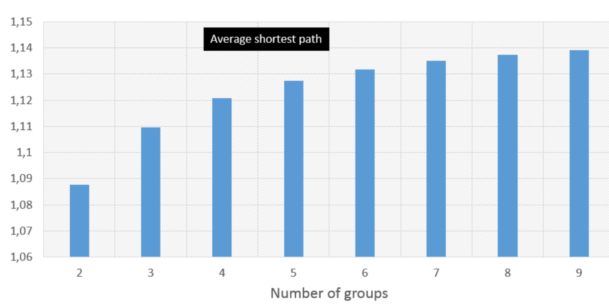

Figure 14. [**ASP**] Number of groups

In fig. 13 and 14 influence of number of layers and number of groups is analysed. As it can be seen value of ASP is rather independent from this two parameters, but it is worth mentioned that with bigger number of layers ASP is decreasing. Such inconsistency can be explained by method of calculation. Because API natively do not support calculation of ASP between layers, each layer had to be calculated independently and all ASPs where than averaged. This graph shows that this way is not correct and can lead to false

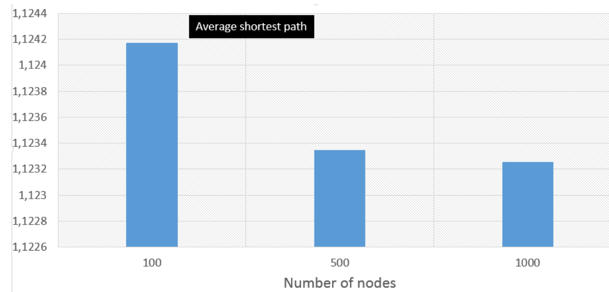

Figure 15. [**ASP**] Number of nodes

conclusions. Number of group in fig 14 shows that bigger complexity of the graph leads to longer ASP.

In fig. 15 an evidence that number of nodes has no significant influence on ASP. Another interesting conclusion comes from 16. Once more before 10% marker of probability of edge existence between nodes, network shows no complexity(ASP=1), than after 10% measure reaches its peek and than slowly decreasing that at level of 50% value once more is 1. Reason here is exactly the same like for fig. 12

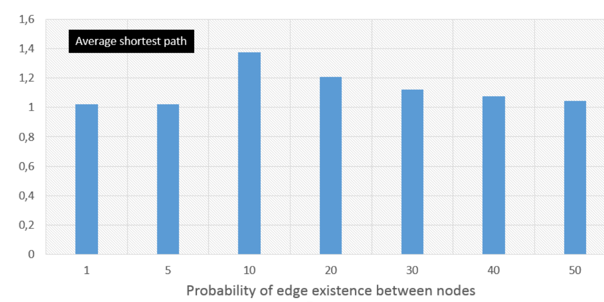

Figure 16. [**ASP**] Probability of edge existence between groups

D. Diameter

To define diameter correctly it is necessary to introduce some definitions from graph theory.

Geodesic distance [18] between nodes is a number of edges in a shortest path between them.

Eccentricity $\epsilon(v)$ of a node v is simply the biggest geodesic distance from node to any other node in the graph.

Than diameter d of the graph is maximum eccentricity in the network. Formally:

$$d = \max_{v \in V} \epsilon(v) \tag{9}$$

Diameter value has similar influence on network features like average shortest path. Very interesting is problem of diameter degree [19]. Briefly degree diameter problem is a problem of finding the largest graph with diameter k and largest node degree d. Size of such graph is upper bounden by Moore bound [20].

In fig. 17 ot can be observed that with growing network complexity parameter is decreasing. It means that if number of groups is bigger then diameter is more similar to ASP.

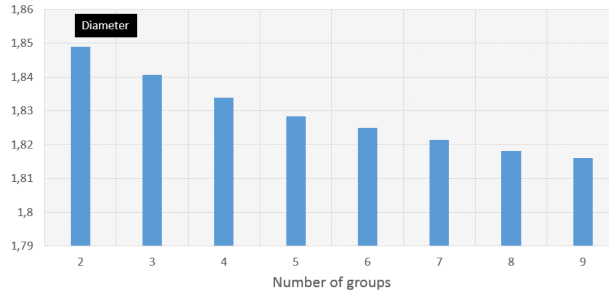

Figure 17. [**Diameter**] Number of layers

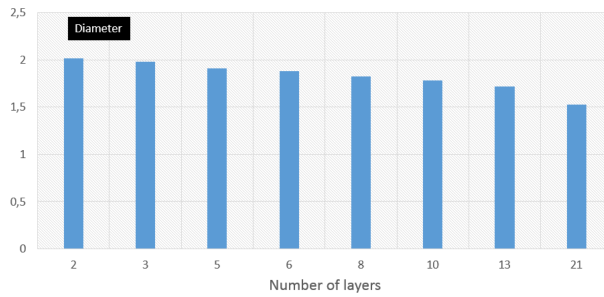

Figure 18. [**Diameter**] Number of groups

Once more number of layers has unexpected influence on diameter(fig. 18. Reason for that here is exactly the same like in fig. 13.

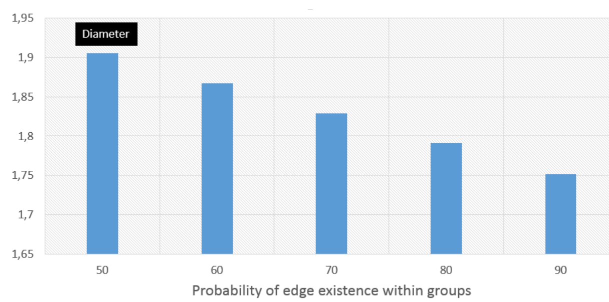

Figure 19. [**Diameter**] Probability of edge existence within group

In fig. 19 influence on diameter by probability of edge existence within group is presented. When in groups there are more edges then diameter is decreasing. It means that eccentricity of the graph is lower when groups inside are better connected.

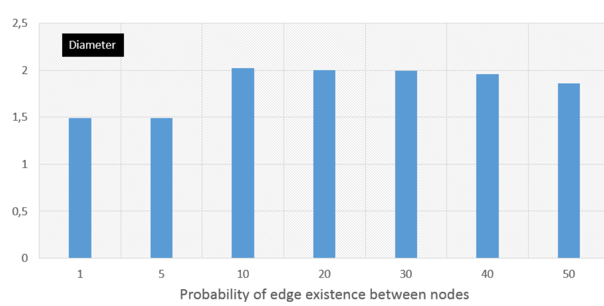

Figure 20. [**Diameter**] Probability of edge existence between groups

In fig. 20 once more probability of edge existence between groups shows that before 10% marker network is less complex than on the right side of this point. However diameter by probability $> 10\%$ is similar and near value of 2.

V. Conclusions and Future Work

Parameter of graph homophilly has no influence on analysed parameters of the network. It was proved that this parameter has only influence in task of collective classification on such generated graphs, but this task is out of scope of this article. However feature of generating labeled data is very rare among all available network generators.

As it can be observed, only small set of possible MuNeG parameters was checked in experiments. Especially number of generated nodes was not so much tested. Tests of very complex networks with bigger amount of nodes are still an open tasks.

Another issue are social measures like clustering or average shortest path. After experiments show upproblems, that MuNeG can generate only dense connected networks with short path and diameter. Clustering also showed that usually graph generated by MuNeG tend to has high value. It is hard to prepare input parameters to generate network with clustering < 0.5.

However MuNeG generator is a flexible tool which makes possible to generate:

- Social networks similar to small worlds
- Complex and densely connected networks
- Graph with different node degree
- Labeled network data

and possibly many more, because not all parameters values was tested.

All this networks are multiplexes. Thanks MuNeG already tested phenomenas in single layer domain, can be easily proved in multilayer graphs. Because it is often hard to collect such well defined multiplexes, MuNeG tool can be very helpful in researching this already not so good checked domain. Tool is completely open source and everybody is invited to developed it and to test in more complex and demanding tasks.

Acknowledgment

Calculations have been carried out using resources provided by Wroclaw Centre for Networking and Supercomputing (http://wcss.pl), grant No. 344

References

[1] S. Boccaletti, G. Bianconi, R. Criado, C. Del Genio, J. Gómez-Gardeñes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin, "The structure and dynamics of multilayer networks," *Physics Reports*, vol. 544, no. 1, pp. 1–122, 2014.

[2] H. Eldardiry and J. Neville, "Multi-network fusion for collective inference," in *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*. ACM, 2010, pp. 46–54.

[3] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[4] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003.

[5] D. J. Watts and S. H. Strogatz, “Collective dynamics of small-world networks,” *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[6] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[7] T. Kajdanowicz, P. Kazienko, A. Popiel, and J. Alvarez-Rodriguez, “Layer reduction and decision fusion in relational classification for multi-layer social networks,” in *WISE 2014 - The 15th International Conference on Web Information System Engineering*, 2014.

[8] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” *AI magazine*, vol. 29, no. 3, p. 93, 2008.

[9] *Python Software Foundation. Python Language Reference, version 2.7. Available at http://www.python.org*.

[10] R. Bradshaw, C. Citro, and D. Seljebotn, “CYTHON: THE BEST OF BOTH WORLDS,” *CiSE 2011 Special Python Issue*, p. 25, 2010.

[11] R. Diestel, “Graduate texts in mathematics: Graph theory,” 2000.

[12] P. Erdős and T. Gallai, “Graphs with points of prescribed degrees [in hungarian],” *Matematikai Lapok*, pp. 264–274, 1950.

[13] K. Cameron and J. Edmonds, “Some graphic uses of an even number of odd nodes,” in *Annales de l’institut Fourier*, vol. 49, no. 3. Chartres: L’Institut, 1950-, 1999, pp. 815–828.

[14] V. Havel, “A remark on the existence of finite graphs,” *Casopis Pest. Mat*, vol. 80, no. 477-480, p. 1253, 1955.

[15] F. Harary and E. M. Palmer, *Graphical enumeration*. Elsevier, 2014.

[16] S. Wasserman, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[17] J. Travers and S. Milgram, “An experimental study of the small world problem,” *Sociometry*, pp. 425–443, 1969.

[18] J. Bouttier, P. Di Francesco, and E. Guitter, “Geodesic distance in planar graphs,” *Nuclear Physics B*, vol. 663, no. 3, pp. 535–567, 2003.

[19] E. Bannai and T. Ito, “On finite moore graphs,” 1973.

[20] R. Damerell, “on moore graphs,” in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 74, no. 02. Cambridge Univ Press, 1973, pp. 227–236.