International Journal
Information Technology Applications (ITA)
Volume 10, Number 1, September 2021

STATISTICAL MECHANICS OF COMPLEX GRAPHS

David Šablatúra

Abstract:

Modelling of graphs as abstract mathematical structures is often utilised in myriad of studies across the whole spectrum of scientific fields. This paper aims to investigate some of the graph theory characteristics of complex systems. Such investigation is applicable for real-world phenomena studies and optimisation simulations of models representable by graph theory structures. A random model generation algorithm was developed to build random graphs that were further perturbed by adding edges according to a custom preferential edge attachment algorithm. The edge attachment algorithm forces nodes in the model to coalesce into large but fewer components. Analysis of the analytically validated model graph structures by means of node degree histograms supports the proposed behaviour of graphs upon new edge addition.

Keywords:

Complex graphs, node degree, graph theory, graph and network algorithms, MATLAB.

ACM Computing Classification System:

Mathematics of computing, discrete mathematics, graph theory, random graphs.

Introduction

Graph mechanics, as both a theoretical and a practical background, constitute to various phenomena in physics, medicine, chemistry, biology, social sciences, transport systems etc. (Palko, 2017). Economics, pollution, epidemics are only a tip of the iceberg areas where graphs can be utilized to study the relationships and dynamics in complex systems. Substantial amount of research was done around small world networks, theory of which postulates that even in apparently complex network structures, the average distance between nodes remains unexpectedly small as that is analytically proportional to the logarithm of the number of nodes (Pelánek, 2011. Štefanovič, 2021). Barabási summarised networks with various average path lengths (Réka et Barabási, 2002). Existing models of graph manipulation were studied. Evolving models for preferential and random evolution, both for edge and node addition are well described in literature (Blum et al. 2020. Réka et Barabási, 2002).

A simple random graph generator algorithm was developed, implemented, and validated against theoretical concepts from literature. Preferential node attachment algorithm was developed, which manifested in nodes to components coalescence. Such pseudo-random graph's properties were subsequently analyzed.

1 Graph Model Specification

Seemingly similar applications of graph theory might exhibit contradictory graph properties. For example, a network of Facebook users is an undirected graph as relations are defined by friendships, on the other hand Twitter and Instagram user base network is specified by who is following who, only representable by directed graphs. It is therefore important to understand these properties prior to accepting model assumptions. In this paper, the following standard notation will be used to represent basic graph properties, where by definition, an undirected graph is representable by any two of the three:

n - number of nodes
e - number of edges
R - ratio: number of edges / number of nodes

For each undirected graph, the maximum number of edges is bounded by the number of nodes

$$0 < R < \left(\frac{\binom{n}{2}}{n} = \frac{n-1}{2}\right) \quad (1)$$

A custom algorithm implemented to generate random graph structures was used such that for a specified input set of parameters (e.g. number of nodes $\mathrm{n} = 30$ and ratio of edges to nodes $R = 0.5$), an $n \times n$ adjacency matrix G is created. Each new edge, connecting two nodes, is then allocated to the i-th row and k-th column from those node pair combinations, that do not already have an edge connecting them. In practice, the algorithm looks for a random 0, representing an absence of an edge for the ordered pair of nodes combination. The non-existence of a node is often represented by a value of "Inf" (∞) instead of a "0", which is inevitable in analysis involving weighted graphs. Since this paper focuses on pseudo-static statistical graph analysis and no node-to-node distances are calculated, it is adequate to consider undirected graphs only. Hence only upper right triangle of the adjacency matrix is considered. By representing edges of weight "1" ("exists") only and excluding the matrix diagonal, the model only represents irreflexive, undirected (symmetric) unweighted graphs (Blum et al. 2020, Thulasiraman et al. 2016).

To enable analysis of the components, disconnected subgraphs in a graph, a MATLAB built-in function – **`concomp()`** was utilised. Which returned indexes of components per each node in the graph. After postprocessing this data, the custom source code prints the following console output:

```
Console output: Graph components
===========================
An adjacency matrix was created
Number of nodes:30
Number of edges:15
edges/nodes:0.5
===========================
Number of Components = 16
All these nodes are connected: 1 7 24
All these nodes are connected: 2
All these nodes are connected: 3 4 5 8 11 20 23 26
All these nodes are connected: 6
All these nodes are connected: 9 12 16
All these nodes are connected: 10 27
All these nodes are connected: 13
All these nodes are connected: 14
All these nodes are connected: 15
All these nodes are connected: 17
```

```
All these nodes are connected: 18 22
All these nodes are connected: 19 25
All these nodes are connected: 21
All these nodes are connected: 28
All these nodes are connected: 29
All these nodes are connected: 30
===========================
Model test, nodes in components == nodes in graph :30 == 30
This many COMPONENTS:          1x     2x     3x     10x    |
have got this many nodes:      8n     3n     2n     1n     |  SUM=30nodes
===========================
```

As presented in (Fig.1), the generated graph structure satisfies the requirements for number of edges and nodes. Double checked by comparing the summation from component nodes and overall graph number of nodes. As a result of true random graph properties, sixteen individual components were created as the nodes coalesced in the process of adding edges. Comprised of ten components each with one node, three components with two nodes each, two components with two nodes each and a one component with eight nodes, overall thirty nodes.

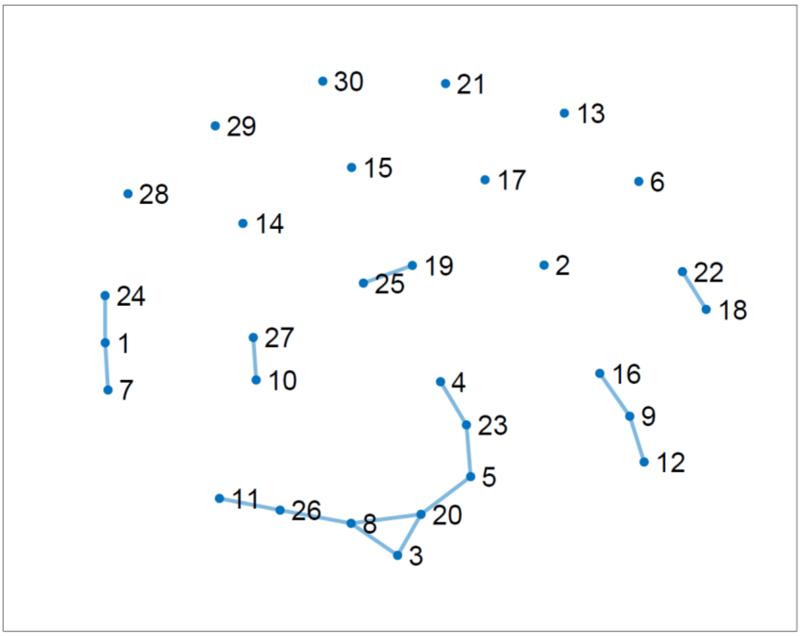

Fig.1. Structure of the generated random Graph, n=30, e=15.

Multiple graphs with $n = 1000$ and $R = 0.5, 1, 1.5$ were constructed this way, which yielded in the node – edge distribution in (Fig.2). Calculated as a histogram of the column-wise summations from the adjacency matrix. Due absence of directed edges, number of edges distribution is equivalent to the node total degree distribution as there are no directional node in/out-degrees defined.

Edge probability distribution as a function of the node 1-degree number

For all nodes in the graph, it is useful to first find the number of its closest 1-edge neighbours. The algorithm for calculation of the node degree number involves the aforementioned column-wise summation of the number of existing nodes for each i-th node – column.

For example, the first node has got 2 one-edge neighbours (node degree of two) and there are only 2 nodes, that have node degree of 3 (8^{th} and 20^{th} nodes) as evidenced in (Fig.3). The node degrees are stored in a vector A, presented in Console output below.

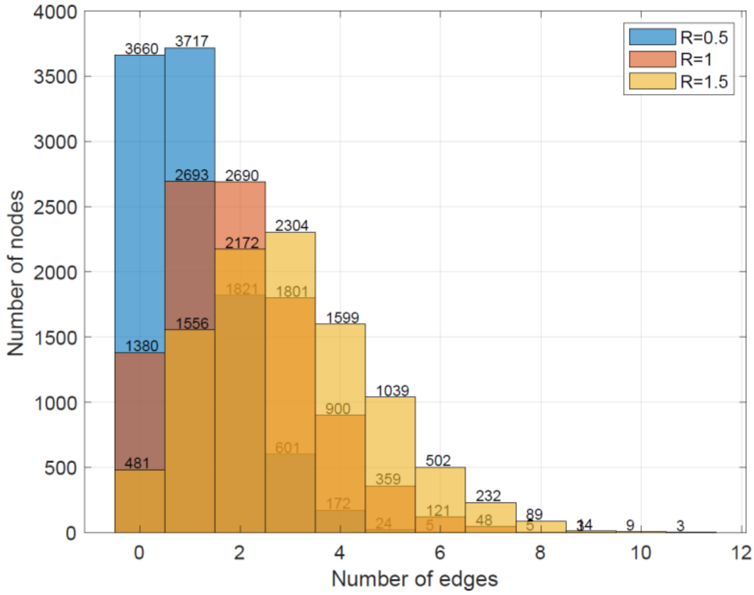

Fig.2. Edge degree distribution. (see online journal for coloured graph)

Console output: Vector A (30 nodes)									
A = 2	0	2	1	2	0	1	3	2	1
1	1	0	0	0	1	0	1	1	3
0	1	2	1	1	2	1	0	0	0

Before the next step, unity is added to the vector, to ensure a useful property exploited later, $A = A + 1$.

Console output: Vector A (30 nodes)									
A = 3	1	3	2	3	1	2	4	3	2
2	2	1	1.	1	2	1	2	2	4
1	2	3	2	2	3	2	1	1	1

The probability of an existing i-th node to gain a new friendship to another existing node is then rationed based on these numbers. It is a probability distribution B_k defined by relative node degrees(+1). Analytically obtained by equation $B_k = A_k / \sum A$.

Console output: Vector B (30 nodes)					
B = 0.0500	0.0167	0.0500	0.0333	0.0500	0.0167
0.0333	0.0667.	0.0500	0.0333	0.0333	0.0333
0.0167	0.0167	0.0167	0.0333.	0.0167	0.0333
0.0333	0.0667	0.0167	0.0333	0.0500	0.0333
0.0333	0.0500	0.0333	0.0167	0.0167	0.0167

To enable an index selection based on these relative probabilities, cumulative probability property is used, by splitting the interval of $\langle 0,1 \rangle$ to n intervals based on $O_k = \sum_{i=1}^{k} B_i$.

Console output: Vector O (30 nodes)						
O =	0.0500	0.0667	0.1167	0.1500	0.2000	0.2167
	0.2500	0.3167.	0.3667	0.4000	0.4333	0.4667
	0.4833	0.5000	0.5167	0.5500.	0.5667	0.6000
	0.6333	0.7000	0.7167	0.7500	0.8000	0.8333
	0.8667	0.9167	0.9500	0.9667	0.9833	1

A randomly generated number from $\langle 0,1\rangle$ is always smaller or equal to at least one of the O numbers. The source code identifies the first of the numbers k such that the O_k satisfies the condition. Resulting in that node index number k. Distribution of these node degree generated node index numbers is perfectly correlated to the number of closest neighbours – node degree numbers. The node with this index number is then allocated a new random probability defined edge (P-edge) to any other node not yet interconnected with k-th node. For example even a node from its existing component. The unity was added to the A matrix to aid node addition to those nodes, that are individually standing components, that is nodes with the degree 0. Otherwise these 0-degree nodes could have had new edges added only if another non-0-degree nodes were already selected and the 0-degree nodes were added randomly in the step 2 as an undirected *edge-end* node. The probability of a new edge in an empty n node graph would have been 0.

The algorithm is demonstrated in following Figures, where for a random graph with 30 nodes and the initial edges to nodes ratio of 0.5 P-edges were added (Fig.3).

At first, the node 26 is selected based on its initial relative degree number of 2. And then, randomly selected non-existing connection is made, with node 10, which is in a separate 2 node component. Therefore, in this case, an addition of one edge means addition of two nodes, see (Fig.4) This can vary from 0 nodes when connecting to a node within the component to the maximum component size in other graph components.

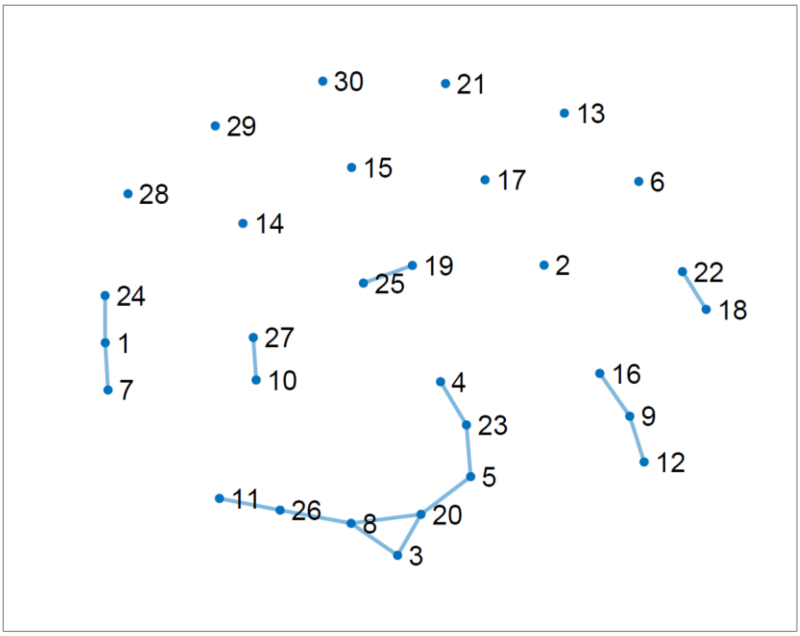

Fig.3. Structure of the generated random Graph, n=30, e=15.

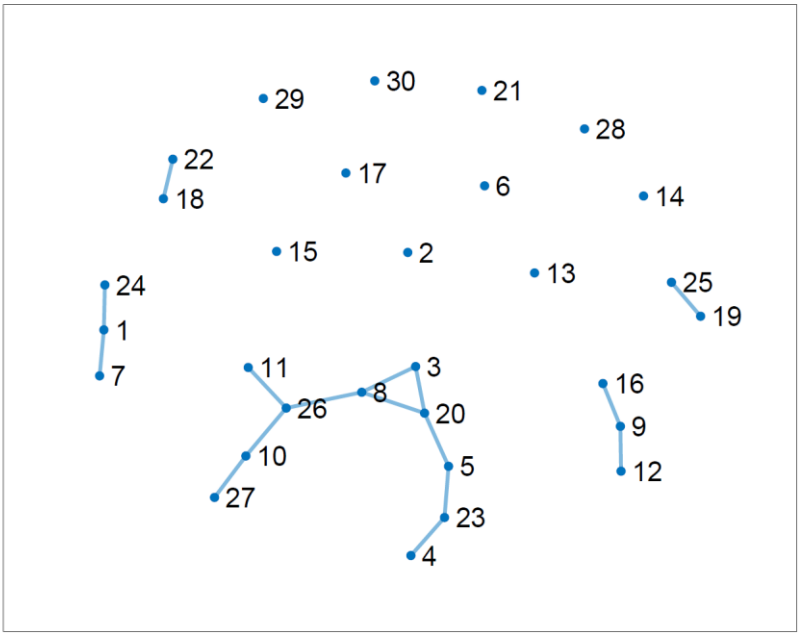

Fig.4. Structure of the generated Graph with an extra P-edge, n=30, e=16.

Repeating this 14 more times, a graph in (Fig.5) with a master component was obtained.

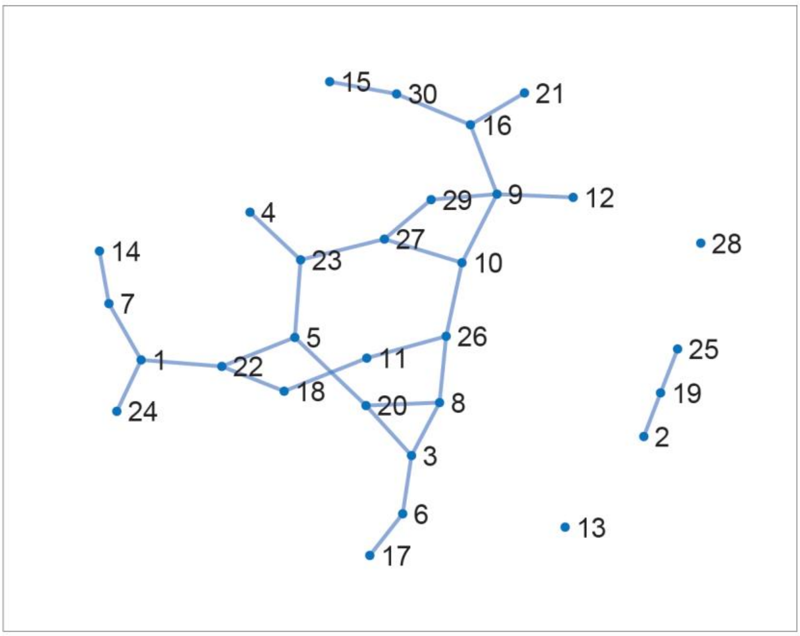

Fig.5. Structure of the generated Graph with an extra P-edge, n=30, e=30.

The output of the program successfully identifies node relations to the components, as previewed in console output below.

Console output: Graph components

```
===========================
Number of Components = 4
All these nodes are connected: 1 3 4 5 6 7 8 9 10 11 12 14 15 16 17 18 20 21 22
23 24 26 27 29 30
All these nodes are connected: 2 19 25
All these nodes are connected: 13
All these nodes are connected: 28
===========================
Model test, nodes in components==nodes in graph :30==30
This many COMPONENTS:         1x      1x      2x      |
have got this many nodes:     25n     3n      1n      |   SUM=30nodes
===========================
```

As presented in (Fig.5), an addition of edges based on the degree of the nodes, a master component quickly emerges.

For a number of nodes $n = 1000$ and original number of edges $e = 500$, (Fig.6), it is clear that after increasing the edges to nodes ratio from $R = 0.5$ to $R = 0.7$, the number of nodes in the largest component size rapidly increases, (Fig.7).

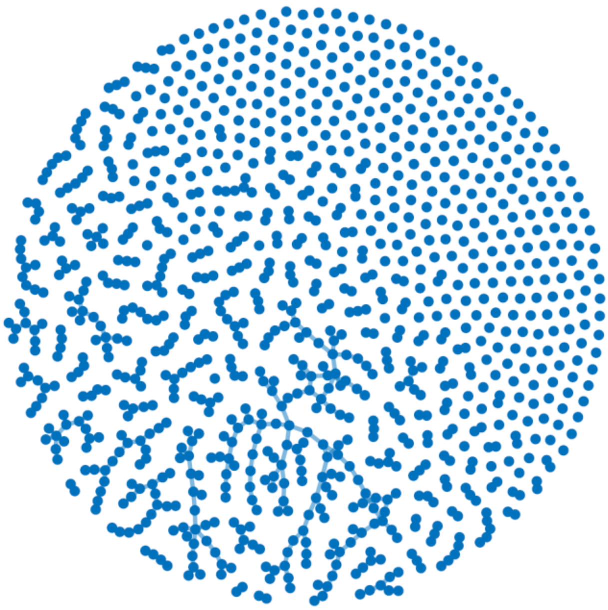

Fig.6. n=1000, e=500.

The discrepancy between structure of the graph in (Fig.7) generated by having 500 random edges and additional 200 node degree probability defined edges and a random graph with no P-edges, but 700 random edges instead, is small but not negligible, as evidenced in (Fig.8).

This is an interesting finding, as the distribution of a truly random graph highly correlates to the distribution of an initially lighter graph (less edges) with added nodes according to the custom node degree derived algorithm.

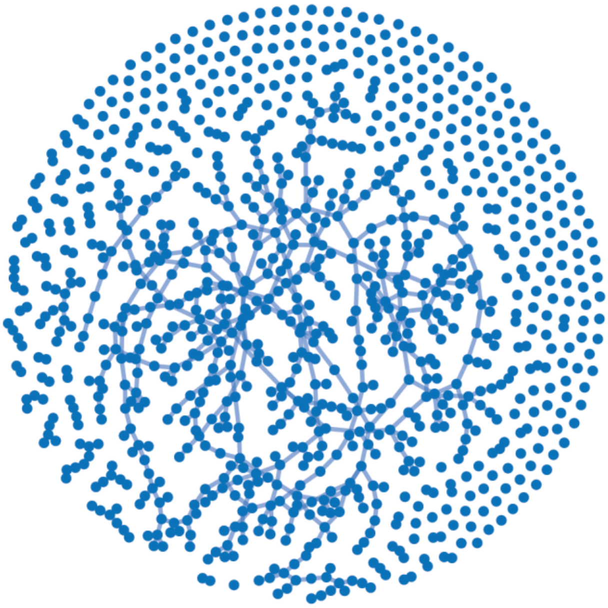

Fig.7. n=1000, e=500 + 200 P-edges.

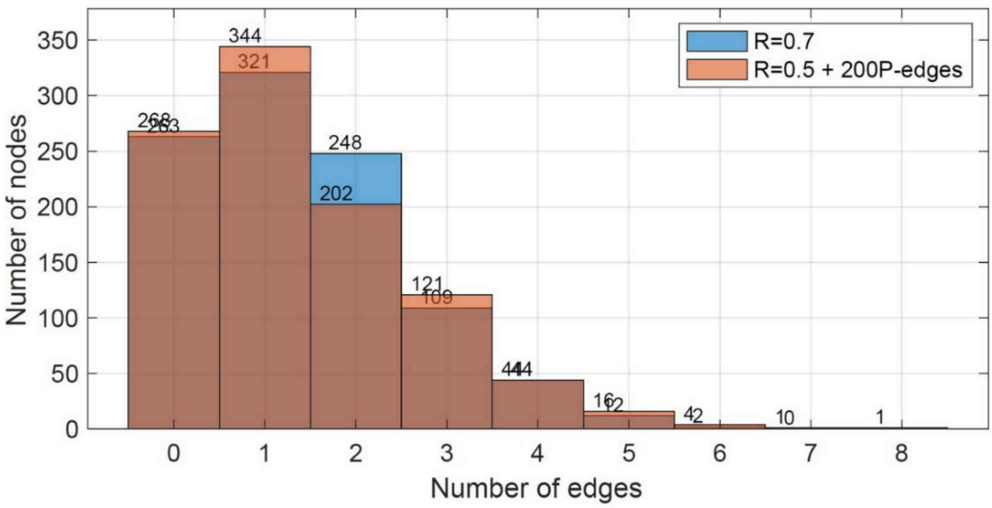

Fig.8. Node degree distributions for two graphs both with n=1000, e=700
(edges to nodes ratio R=0.7)
orange graph has got original R=0.5 and 200 extra Preferential edges.

Generally, the first node in the new node pair in custom P-edge algorithm is selected according to the node degree relative number and the other to create the edge is selected randomly from nodes it is not already connected with. Whereas truly random graph algorithm selects all non-existent edges (unordered node pairs). The comparison of the calculations below, show discrepancy in one of the new edge nodes probability distributions.

$$G(n=5, e=12) = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Probability of the node becoming a new edge source node:

$A(G=1) = [3\ 3\ 1\ 3\ 2]$ $A+1 = [4\ 4\ 2\ 4\ 3], \sum A+1 = 17$ $P_{\text{new P-edge}} = \left[\frac{4}{17}, \frac{4}{17}, \frac{2}{17}, \frac{4}{17}, \frac{3}{17}\right]$	$A(G=0)_{\text{irreflexive}} = [1\ 1\ 3\ 1\ 2], \sum A = 8$ $P_{\text{new random edge}} = \left[\frac{4}{8}, \frac{4}{8}, \frac{2}{8}, \frac{4}{8}, \frac{3}{8}\right]$

The specific probability distributions for this graph structure are visualised in (Fig.9). Although probabilities for individual nodes may vary significantly, the resulting structures are not too dissimilar.

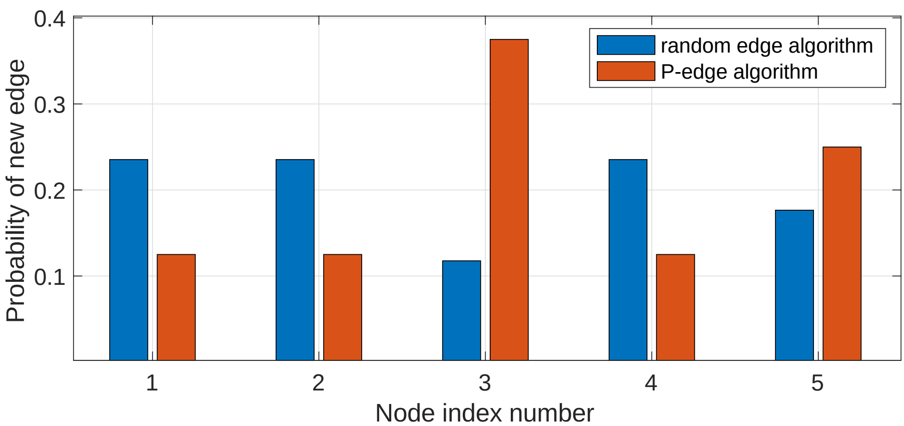

Fig.9. Probability of the node acting as an undirected edge source.
(P-edge = preferential edge algorithm generated, it is every second column)

2 Stability of Solution

Upon repeated execution of the P-edges algorithm, for identical input settings with number of nodes $n = 1000$ and number of edges $e = 500$, the algorithm of the P-edge allocation produces similar graph structures, observed in (Fig.10). Curves for data1 through data7 depict 7 independent simulation runs.

(Fig.11) demonstrates the same data, both axes were adjusted to logarithmic scale. The apparent linearity in the log-log axes, demonstrates the inverse proportionality such that number of nodes $n \propto$ inverse of the number of graph components and vice versa.

Detailed analysis of a larger number of nodes graphs at various edge to node ratios led to a conclusion that the simulation of the node relationships with this Monte Carlo probability method leads to expected results and deducing from that, the model is valid.

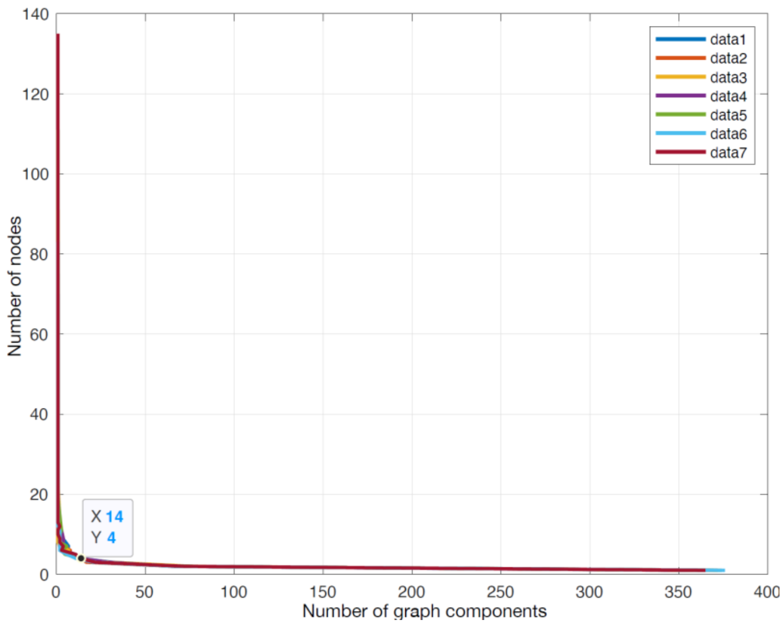

Fig.10. Distribution of the number of components versus number of nodes for an initial graph with n=1000, e=500 and additional 100 P-edges.

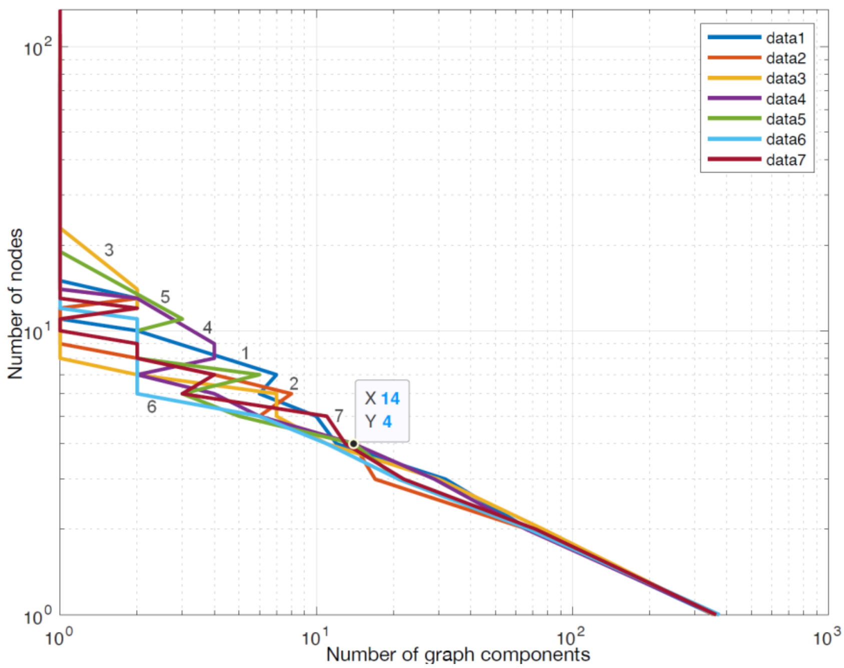

Fig.11. DETTO log-log distribution of the number of components versus number of nodes for an initial graph with n=1000, e=500 and additional 100 P-edges.
(refer to the online journal issue in colour)

3 Large Graph Analysis

In this section, the percentage fraction of the individually standing nodes (one node components, 0-degree nodes) is the primary observed quantity. The independent variables are number of nodes and number of edges.

A randomly generated graph with a specific number of nodes n and an edges to nodes ratio of $R = 0.5$, will always have $36 - 37\%$ of its nodes with node degree of 0. In other words, the 36% of the nodes are individually standing components. This is has been demonstrated both analytically and experimentaly. The experiment results are presented in the (Table 1).

Table 1. Convergence of the fraction of the individually standing nodes in graphs with R=0.5

Number of nodes	0-degree nodes	Number of edges
10	30%	5
100	39%	50
1000	36.1%	500
10000	36.38%	5000
15000	5538=36.92%	7500

Analogously, the analysis for the edges to nodes ratio of $R = 0.8$, resulted in a fraction of apporximately 20% of the nodes standing individually, (Table 2).

Table 2. Convergence of the fraction of the individually standing nodes in graphs with R=0.8

Number of nodes	0-degree nodes	Number of edges
10	2	8
100	19	80
1000	216	800
10000	2034	8000
15000	3023	12000

The fraction of individually standing nodes in random graphs with an edges to nodes ratio $R = 1.1$ converges to around 11%, (Table 3).

Table 3. Convergence of the fraction of the individually standing nodes in graphs with R=1.1.

Number of nodes	0-degree nodes	Number of edges
10	1	11
100	12	110
1000	106	1100
10000	1122	11000
15000	1655	16500

Finally, various simulations with a large number of nodes are summarised in the (Table 4). Presented graphically in (Fig.12). The relationship is logarithmic as validated in following paragraphs.

$$\log\left(\frac{0 - degree\ nodes}{n\ number\ of\ nodes}\right) \propto R$$

Table 4. Number of 0-degree nodes as a function of edges to nodes ratio R.

Number of nodes	0-degree nodes	Edges to nodes ratio R
30000	11028	0.5
30000	9013	0.6
30000	6063	0.8
30000	4955	0.9
30000	4056	1
30000	3346	1.1
30000	1888	1.4
30000	1122	1.6
30000	637	1.9

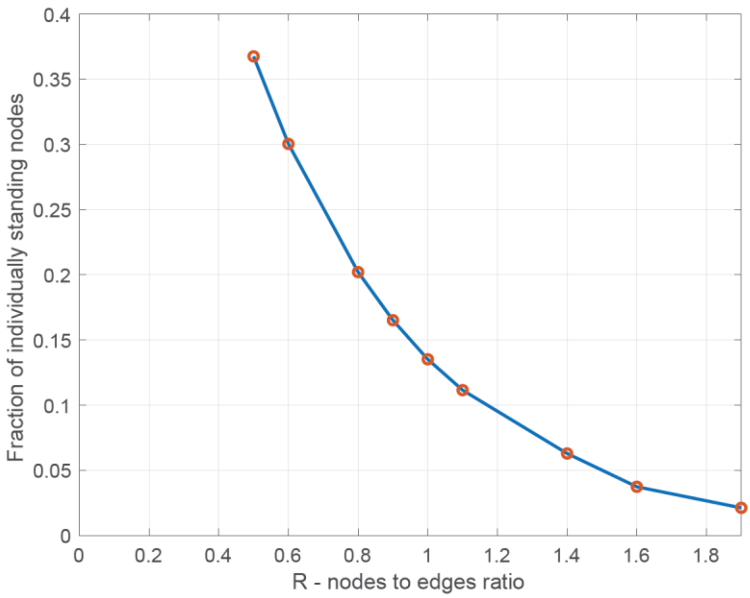

Fig.12. Experimental fraction of individually standing nodes in a graph G(n,nR) as a function of the edges to nodes ratio R.

The experimental solution converges to a curve analytically derived from Erdős–Rényi graph model. A useful notation of a truly random graph according to Erdős–Rényi is the notation of $G(n, M)$, which represents a randomly selected graph G from set of all possible graphs with a number of nodes n and a number of nodes M. The order of the edges matters, two graphs obtained by a permutation of the other (replace two nodes one by the other) are distinct. There exist 3 possible graphs for a number of nodes $n = 2$ and a number of edges $e = 2$, as presented in (Fig.13). The selection for e edges is made from $K = \frac{n^2-n}{2}$ places, corresponding to the elements of the upper right triangle of the adjacency matrix $A(n)$, as that relates to undirected simple graphs.

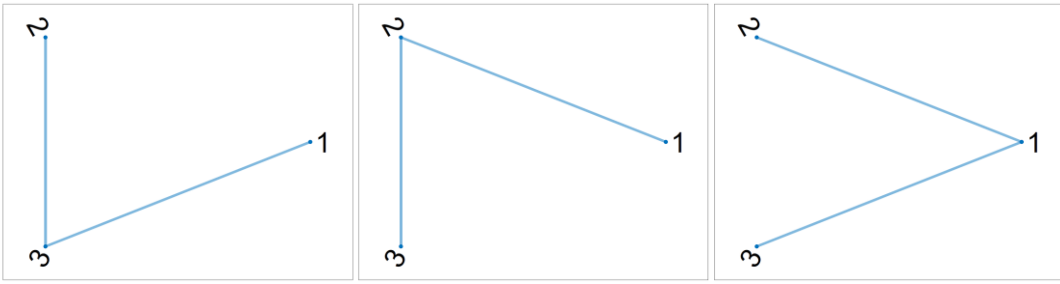

Fig.13. All Erdős–Rényi $G(3,2)$ model graphs.

Evaluating the equation (Eq.2) results in the number g of all possible graphs $\mathrm{G(n, M)}$, that might incur using the number of nodes n and a number of edges e, where $\mathrm{M = e}$.

$$g = \binom{K}{e} = \binom{\binom{n}{2}}{e} = \binom{n(n-1)/2}{e} \tag{2}$$

Similarly, the number of all possible graphs, which do have an isolated node is a function as per (Eq.3).

$$B_1 = \binom{\binom{n-1}{2}}{e} \tag{3}$$

The probability that the specific node is an individually standing 1-node component is modelled by (Eq.4).

$$P = \frac{B_1}{g} = \frac{\binom{\binom{n-1}{2}}{e}}{\binom{\binom{n}{2}}{e}} \tag{4}$$

The numerical computation of these double factorial equations for even small graphs reaches standard, computational memory limits quickly $(n = 170)! = 7.2574E306$ and a truncation rounding error $(n > 170)! = \infty$. After research on the possibilities of working with very large numbers, an approach of analytical rearrangement was followed. An interesting equivalent mathematical notation was discovered by writing out the factors of the 2-level deep binomial coefficients. These product of series equations were found to be computationally more effective. Hence, the number of all possible Erdős–Rényi graphs that isolate a node becomes (Eq.6).

$$B_1 = \binom{\binom{n-1}{2}}{e} = \binom{\frac{n^2-3n+2}{2}}{e} \tag{5}$$

$$= \frac{\frac{n^2-3n+2}{2} \cdot \frac{n^2-3n+2-2\cdot 1}{2} \cdot \frac{n^2-3n+2-2\cdot 2}{2} \cdot \ldots \cdot \frac{n^2-3n+2-2(e-1)}{2} \cdot \boxed{\frac{n^2-3n+2-2e}{2}!}}{\boxed{\left(\frac{n^2-3n+2}{2} - e\right)!} \cdot e!}$$

$$= \prod_{a=0}^{e-1} \frac{n^2-3n+2-2a}{2(a+1)} \tag{6}$$

And the number of all possible Erdős–Rény graphs is represented by (Eq.8):

$$g = \binom{\binom{n}{2}}{e} = \binom{\frac{n^2 - n}{2}}{e} \tag{7}$$

$$= \frac{\frac{n^2-n}{2} \cdot \frac{n^2-n-2\cdot 1}{2} \cdot \frac{n^2-n-2\cdot 2}{2} \cdot \ldots \cdot \frac{n^2-n-2(e-1)}{2} \cdot \boxed{\frac{n^2-n-2e}{2}!}}{\boxed{\left(\frac{n^2-3n+2}{2} - e\right)!} \cdot e!}$$

$$= \prod_{a=0}^{e-1} \frac{n^2 - n - 2a}{2(a+1)} \tag{8}$$

The difficulty now decreased to a product series equivalent of a single factorial calculation. Combining the (4), (6) and (8), the resulting probability of any node in randomly generated Erdős–Rényi graph is P in form of a product of series in Equation (9).

$$P = \frac{B_1}{g} = \frac{\binom{\binom{n-1}{2}}{e}}{\binom{\binom{n}{2}}{e}} = \prod_{a=0}^{e-1} \frac{n^2 - 3n + 2 - 2a}{n^2 - n - 2a} \tag{9}$$

Evaluating for $n = 120$ and $e = 60$, the ratio becomes $P = 36.33\%$. Similarly, for $n = 120$ and $e = 132$ the analytical expected value is $P = 10.65\%$, which when compared to the experimental value of 11.15% from the (Tab.4) above, represents a relative error 4.7%, upon the basis of the analytical. Similarly, evaluating Eq.(9) at $n = 3000$ a result of 0.1106 is obtained which results in a much smaller relative error of 0.81%.

Calculation for larger integers of number of nodes does yield in a much more stable, converging solution. The stability of the evaluation of equation (9) for nodes number $n = 1000$ through $n = 9000$ is presented in (Fig.14), where the curves for varying number of nodes are highly mutually congruent. The experimental values obtained by simulating a graph of 30000 nodes at a set of edges to nodes ratios also overlay the analytical solutions. In a semi-log figure, the curve becomes linear, which concludes in the fact, the fraction of the individually standing nodes (0-degree nodes) is an exponential function of the edges to nodes ratio.

$$\frac{n_{0-\text{degree}}}{n_{\text{ALL}}} \propto e^{-\text{aR}} \tag{10}$$

After model fit analysis, it was found that experimental values fit an exponential $a \cdot e^{bx}$ model with values of a and b 0.9993 and -2 respectively (0.9775, 1.021) and (-2.031, -1.969) within 95% confidence interval. This is an experimental proof, that the fraction of 0-degree nodes is in fact not only proportional as per equation (10) but equal to decaying exponential function of $2R$.

$$\frac{n_{0-\text{degree}}}{n_{\text{ALL}}} = e^{-2\text{R}} \tag{11}$$

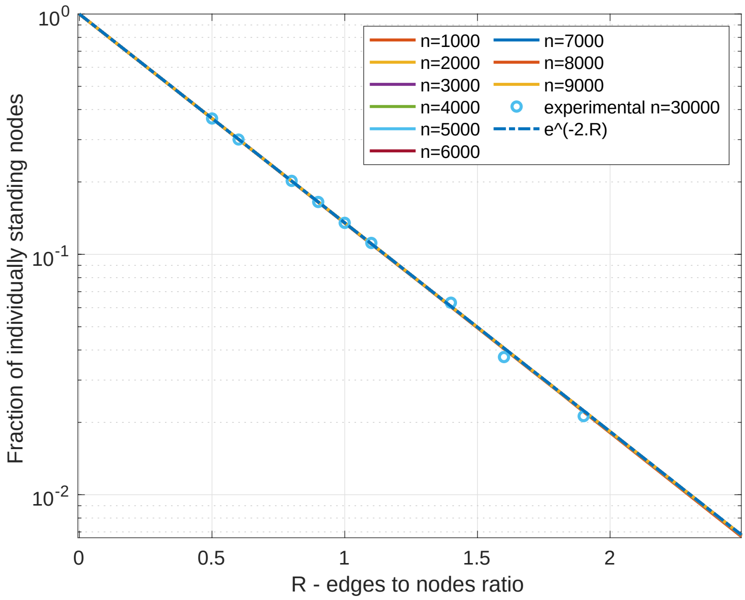

Fig.14. logY – Analytical results (curves) compared to the experimental results, based on Fig.12.

4 Node Degree Distribution Analysis

The next part of the analysis investigates the distribution of the number of nodes with respect to their neighbouring nodes, that is nodes in relation to respective degree numbers. Node degree histogram distributions were selected as a suitable methodology.

In this section, a randomly generated graph with a node number $n = 10000$ and initial number of edges to nodes ratio $R = 1.1$ is amended by the P-edge algorithm, such that nodes with a higher degree have got larger probability of gaining a new edge with any other node.

Histogram in (Fig.15) visualises the distribution of the number of closest neighbours (node degree number) for an initial, truly random graph. The distribution of such structure is binomial in essence.

The right-skewed distribution after adding 1000 P-edges is presented in (Fig.16). Following the algorithm, the distribution has skewed to the right, such that the average shifts to the higher node degrees and the actual magnitude decreases as the distribution spreads out. There are fewer nodes with low node degrees and more nodes with higher node degrees. The sum of the elements remains constant as the number of nodes does not change.

Analogously, after the addition of further 1000 P-edges, the distribution shifts again, the result of this intermediate step is presented in (Fig.17).

Another execution of the algorithm for extra 4000 P-edges resulted in a solution of a graph with initial number of nodes $n = 10000$ and a number of edges $e = 11000$, with additional 6000 P-edges, (Fig.18). The solution is equivalent to the process of generating 6000 P-edges for an initial graph at once, since the P-edge generation is iterative with associative property.

Conclusion is made, that the addition of P-edges to a graph expands the node degree count distribution and reduces the overall average count of nodes with a subjectively low node degree number and vice versa, best evidenced in 2-comparison distributions in (Fig.19). According to Blum et. al, the probability of any vertex having a degree less than a certain number decreases with any added edge in graph as a function of power law distribution (Blum et al. 2020).

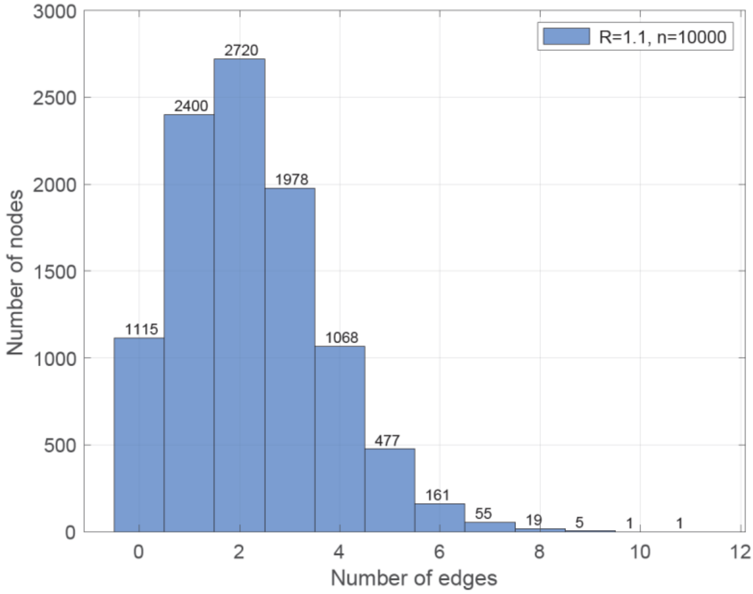

Fig.15. Node degree distribution for an initial graph, n=10000 and e=11000.

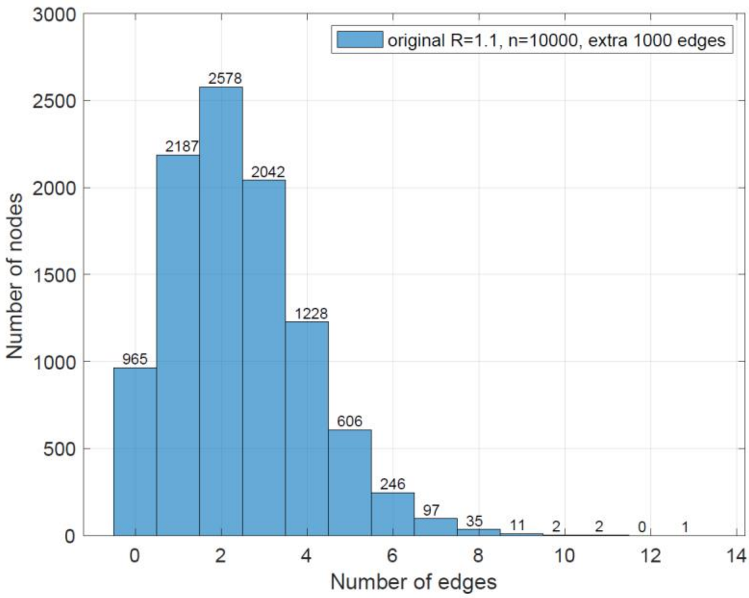

Fig.16. Node degree distribution for an initial graph, n=10000 and e=11000, with additional 1000 P-edges.

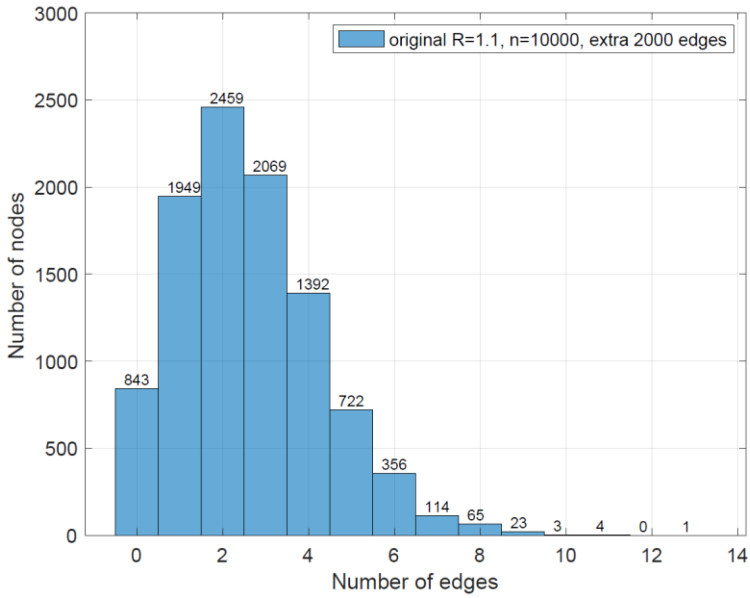

Fig.17. Node degree distribution for an initial graph, n=10000 and e=11000, with additional 2000 P-edges.

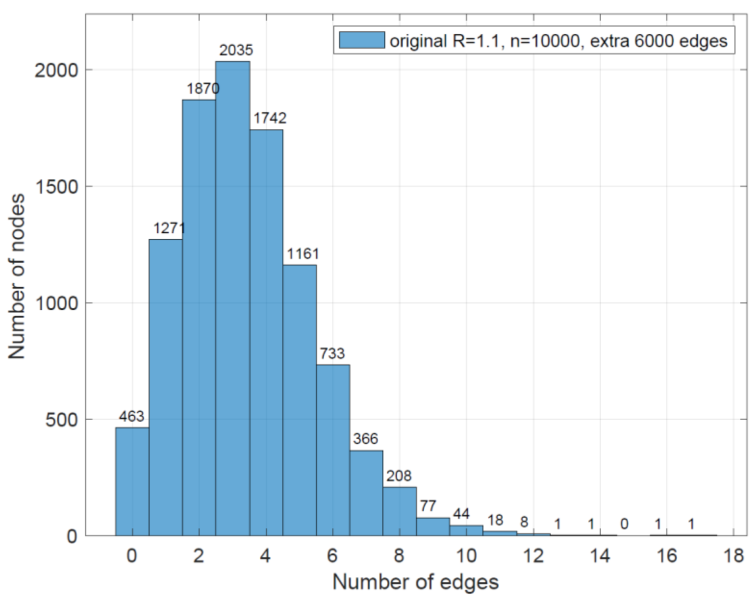

Fig.18. Node degree distribution for an initial graph, n=10000 and e=11000, with additional 6000 P-edges.

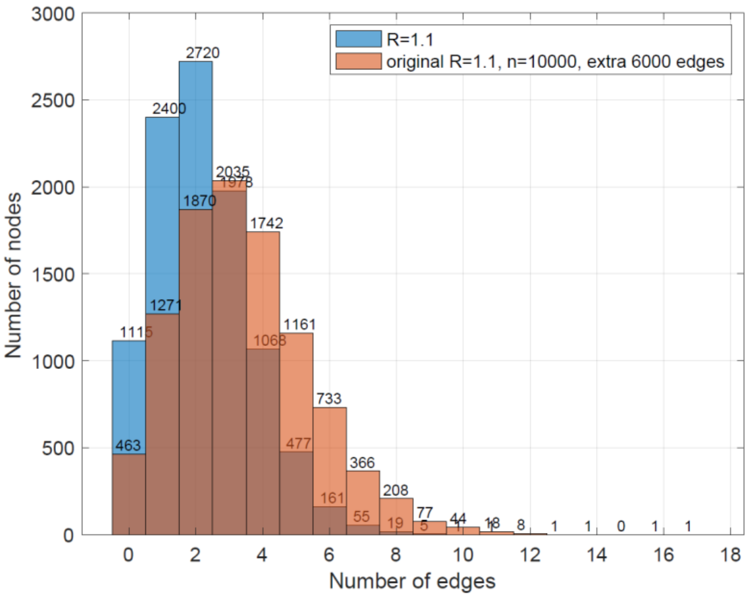

Fig.19. Comparison of the initial graph and a 6000 P-edge complemented graph, n=10000 and e1=11000, e2=17000 edges.

Conclusion

Custom random graph model and preferential edge attachment models were created and validated using the derived equations from graph theory. Several analytical features of complex networks were then experimentally confirmed. Linearly enlarged graphs with constant edges to nodes ratio render equivalent relative node degree distributions. Experimental fit of simulated random graphs presents a decaying exponential trend with respect to edges to nodes ratio. Preferential edge attachment percolation according to custom rules was simulated and analysed.

The approach of simulating quasi-static structured graph models will be expanded in future work, with focus on agent spreading in networks.

Acknowledgement

The baseline of these experiments was initially developed as a student assignment within the Modelling and Simulation module (Štefanovič, 2021). Subsequent work shall proceed with a bachelor thesis level project under the supervision of Juraj Štefanovič, at the Faculty of Informatics, Pan-European University in Bratislava, Slovakia.

References

[1] Pelánek, R. (2011) *Simulace komplexních systémů*. (Simulation of Complex Systems), Masarykova univerzita, Brno, Czech Republic, ISBN: 9788021053182.

[2] Štefanovič, J. (2021) Prednáška 4 - Vlastnosti komplexných sietí. (Lecture 4 - Features of Complex Networks), *the course: Modeling and Simulation*, Pan-European University, Bratislava, Slovakia.

[3] Réka, A. and Barabási A.-L. (2002) Statistical mechanics of complex networks. Department of Physics, University of Notre Dame, *Reviews of Modern Physics*, Volume 74, pp. 47-97. http://barabasi.com/publications/4/complex-networks

[4] Thulasiraman, K., Arumugam, S., Brandstädt, A., Nishizeki, T. (2016) Handbook of graph theory, combinatorial optimization, and algorithms. In: *Handbook of Graph Theory, Combinatorial Optimization, and Algorithms*, Chapman and Hall, doi: 10.1201/b19163.

[5] Palko, V. (2017) *Diskrétna matematika pre informatikov.* (Discrete Mathematics for IT). OZ Hlbiny, ISBN: 9788089743247.

[6] Blum, A., Hopcroft, J., Kannan, R. (2020) *Foundations of Data Science*. Cambridge University Press, ISBN: 9781108755528, doi: 10.1017/9781108755528.

Authors

MSc. David Šablatúra
Student at the Faculty of Informatics, Pan-European University,
Bratislava, Slovakia
sablatura.d@gmail.com

Completed his engineering studies at the University of Hertfordshire (Aerospace technology with pilot studies, BSc) and Cranfield University (Aerospace Dynamics, MSc), with individual projects on bird wing aerodynamics with numerical-empirical approach and an analytic study of hydrofoil stability. Studying at the Faculty of Informatics, he is currently focused on modelling and simulation from applied informatics perspective.