COMBINATORIAL ANALYSIS OF GRAPHS AND FORMULATION OF THE NOTION OF GRAPH SPACE

V. Krishnan[1] and **Z. Sirajunisha**[2]
[1,2]PG & Research Department of Mathematics,
Jamal Mohamed College (Autonomous), Affiliated to Bharathidasan University,
Tiruchirappalli -620 020, Tamil Nadu, India
Email: [1]vkrishnan1987@gmail.com, [2]nishahussain2010@gmail.com

Three Star Paper

Abstract

This paper is concerned with the formulation of a combinatorial technique to count number of graphs that can be constructed using a finite number of labelled vertices. As an abstraction, the concept of 'Graph Space G_∞' is explained as a potentially enumerable number of graphs that could be constructed over infinite vertices.

Keywords: Graphs, Combinatorics, Graph Space, Cosmic Network

1. Introduction

This paper proposes study of the connectivity criteria of undirected simple graphs. Given the number of vertices 'V', one should be able to introduce connectivity between any two vertices by means of an edge. 'Simple graph' is the one where pairs of vertices are connected just by one edge. If any pair of vertices has more than one edge, then it is called 'multigraph'. In this research we are concerned with undirected simple graphs only. Multigraphs are just extensions of simple graphs. Given a finite number of vertices, one can construct potentially infinite multigraphs, of which number of simple graphs would be finite. Alternatively, if the number of vertices is increased, number of simple graphs constructed would also increase exponentially. For example, for a single vertex, 2 graphs could be constructed. For two vertices, one can construct 4 graphs. For three vertices, one can construct 50 graphs. For four vertices, one can construct 946 graphs. For five vertices, 31,714 graphs could be constructed. One can construct 2,064,322 graphs from six vertices. As number of vertices tends to infinity, potentially infinite simple graphs could be constructed which amounts to what we call as "**Graph Space**". All kinds of graphs are sub spaces of Graph Space, which is closed under all types of graph theoretic operations.

2. Literature Survey

Graphs were first used as a purely mathematical way to solve fun problems. One such problem was Königsberg Bridge Problem. There are four land masses separated by water with seven bridges connecting these landmasses in the former city of Königsberg, Prussia, currently Kaliningrad, Russia. The question that was raised was "Is it possible to traverse every one of the seven bridges without using the same bridge twice ? It was a fun riddle that the locals would ponder about and playfully try to solve by choosing various routes throughout the city. But in 1735, Leonhard Euler determined the answer abstractly. In doing so, he pioneered the field of graph theory. In his solution, Euler realized that the features of the land masses were irrelevant, so each landmass could be represented simply by a point called vertex. In essence, Euler reduced the problem to simply following paths between vertices and the edges (the bridges) that connect these vertices. Euler proved that, for any connected graph, to be able to traverse every edge exactly once, every vertex (with the possible exception of the starting and ending vertices) must have even degree (even number of edges connected to that one vertex). The graph model of Konigsberg Bridge problem has each vertex with odd degree. So, Euler concluded that the Königsberg Bridge Problem was an impossible problem to solve.

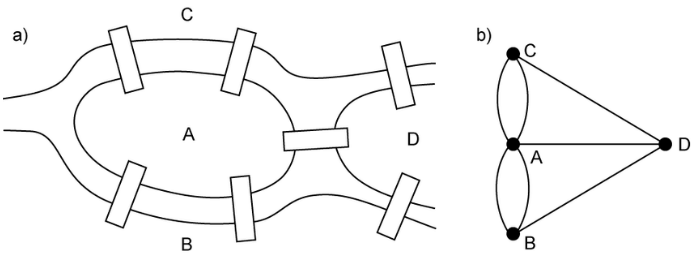


Fig. 1: Königsberg Bridge Problem

Since 1735, many advances in the field of graph theory and topology had taken place. Probably the biggest known application of graph theory is Social Network Analysis (SNA), which is the study of social networks, their structure and how knowing this structure can lead to better understanding of behaviour within social networks. The most well-known social network is, of course, 'Facebook'. There are many tools freely available on Facebook that allow one to study one's own social network structure and see patterns within one's list of friends. Graphs are used to predict interests of people within a social network. Based on these predictions, companies customize advertisements and present to their customers through their Facebook or Twitter feeds. 'Path Optimization and Logistics' is another application where graphs are used extensively. Assume an airline operate their planes flying between certain cities. Three conditions are imposed while planning the flights (i) least number of layovers, (ii) lowest cost, (iii) earliest arrival time. Now, cities are represented by nodes and flights between cities represented by edges. Then this problem is merely finding the most efficient path based on which of the three conditions are satisfied. This problem has a very straightforward solution using Dykstra's algorithm. Computer networks are probably the easiest thing to represent as a graph since networking uses graphs to represent the layout of any computer network. The figure given below shows a computer network which could be modelled as a graph.

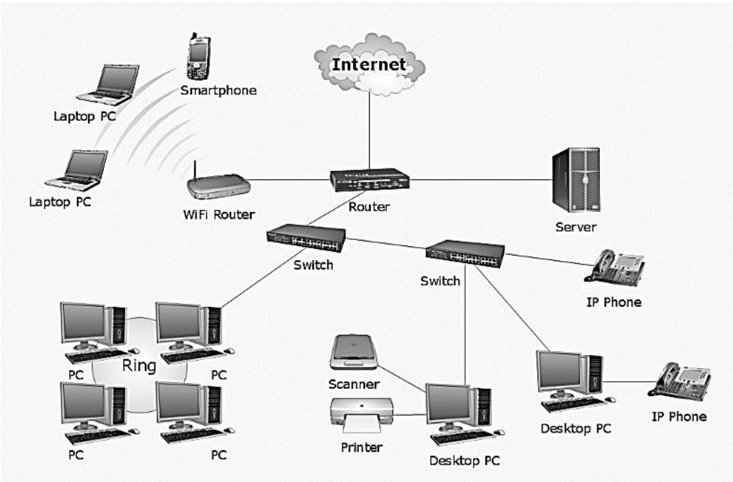


Fig. 2: Computer network modelled as a graph

This graph has nodes consisting of all the physical connections within a network (computers, routers, hubs, switches, bridges, etc.) and the edges in the network will be the actual connections (wireless or hardwired). Having a simple weighted network graph (weights on edges representing load capacities), we can determine if there are vulnerabilities within a network. Almost countless theorems and lemmas have been formulated since 1735.

Some of the most significant theorems often used in graph theory are: 2-factor theorem, Alspach's conjecture, Balinski's theorem, Berge's theorem, BEST theorem, Brooks' theorem, Cederbaum's maximum flow theorem, Circle packing theorem, De Bruijn–Erdős theorem, Burr–Erdős conjecture, Erdős–Gallai theorem, Erdős–Pósa theorem, Erdős–Stone theorem, Even circuit theorem, Fáry's theorem, Five color theorem, Fleischner's theorem, Four color theorem, Frucht's theorem, Fulkerson–Chen–Anstee theorem, Gale–Ryser theorem, Gallai–Hasse–Roy–Vitaver theorem, Geiringer–Laman theorem, Graph structure theorem, Grinberg's theorem, Grötzsch's theorem, Hall-type theorems for hypergraphs, Hall's marriage theorem, Heawood conjecture, Kirchhoff's theorem, Kotzig's theorem, Kuratowski's theorem, Markov theorem, Max-flow min-cut theorem, Menger's theorem, Ore's theorem, Perfect graph theorem, Petersen's theorem, Planar separator theorem, Ramsey's theorem, Road coloring theorem, Robbins' theorem, Robertson–Seymour theorem, Schnyder's theorem, Sims conjecture, Steinitz's theorem, Strong perfect graph theorem, Symmetric hypergraph theorem, Turán's theorem, Tutte theorem, Veblen's theorem, Vizing's theorem, and Wagner's theorem.

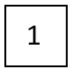
In spite of so much of work carried out in graph theory, it is hardly found in the literature, any 'combinatorial theorem' that stipulates maximum number of graphs that could be constructed using a finite number of vertices. On the other hand, problems have been modelled as graphs and solutions found using such theorems.

This has been considered as a motivating factor for research carried out and results reported in this paper.

3. Combinatorial Analysis of Graphs

An edge is formed between two vertices. A direct edge is an edge formed between two vertices without any intermediary vertex. A graph consisting only of direct edges is called a 'simple graph'. Else it is known as a 'random graph'. Here we are concerned only with simple graphs and devise a method of constructing all possible simple graphs using a given number of vertices. Self-loops are permitted in this formulation.

1-vertex graph space



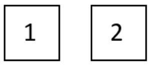
(Graph with no edge)

(Graph with a self-loop in vertex 1)

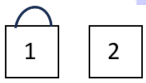
A total of 2 graphs have been constructed using a single labelled vertex 1. (NOTE: $2^1 = 2$)

2-vertex graph space

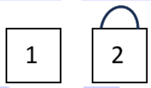
In a 2-vertex graph, one can construct just ${}^2C_2$ = 1 edge graph only. In a 2-vertex graph one can have the combinations of one self-loop at a time, two self-loops at a time amounts to ${}^2C_1 + {}^2C_2 = 2+1 = 3$ self-loops. This means three self-loops graphs. For every self-loop, one can have 1 graph. This means one can construct 3x1 = 3 graphs. One can have just one graph consisting of two vertices with no edges. So, one can construct 1+3+3+1 = 8 graphs from two vertices. This is called as **2-vertex graph space**. (NOTE: $2^3 = 8$)



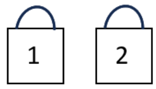
Graph with no edge



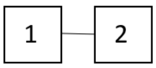
Graph with one self-loop in vertex 1



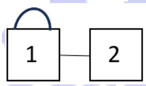
Graph with one self-loop in vertex 2



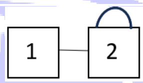
Graph with self-loops in vertices 1 and 2



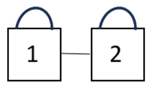
Graph with one edge {1,2}



Graph with one self-loop in vertex 1 and one edge {1,2}



Graph with one self-loop in vertex 2 and one edge {1,2}



Graph with self-loops in vertices 1 and 2 and one edge {1,2}

A total of 8 graphs have been constructed using two labelled vertices 1 and 2.

3-vertex graph space

Let us consider a case study of three vertices 1, 2 and 3 belonging to the set of vertices V. One can build a set of 64 graphs using these three vertices. Let us denote an edge by, say, {1,2} where 1 and 2 are vertices. Self-loops are also considered here. In a 3-vertex graph, m = 3, which means that one can construct ${}^3C_2 = 3$ maximum number of edges. The combinatorics of 3 edges having one edge at a time, two edges at a time, and so on up to 3 edges at a time amounts to ${}^3C_1 + {}^3C_2 + {}^3C_3 = 3+3+1 = 7$ edge graphs. In a 3-vertex graph one can have the combinations of one self-loop at a time, two self-loops at a time and three self-loops at a time, which amounts to ${}^3C_1 + {}^3C_2 + {}^3C_3 = 3+3+1 = 7$ self-loops. For every self-loop, one can associate 7 edge graphs. This means one can construct 7x7 = 49 graphs apart from 7 self-loops graphs and 7 edge graphs. This amounts to saying that one can construct 49+7+7 = 63 graphs using three vertices. One can also have just one graph consisting of three vertices but with no edges. So, one can construct a total of 63+1 = 64 graphs from three labelled vertices and this is called as **3-vertex graph space**. (NOTE: $2^6 = 64$). All the 64 graphs are listed below.

Sl. No.	Edges of graphs	Remarks
1	1, 2, 3	No edge
2	{1,1}, 2, 3	One self-loop
3	1, {2,2}, 3	One self-loop
4	1, 2, {3,3}	One self-loop
5	{1,1}, {2,2}, 3	Two self-loops
6	{1,1}, 2, {3,3}	Two self-loops
7	1, {2,2}, {3,3}	Two self-loops
8	**{1,1}, {2,2}, {3,3}**	**Three self-loops**
9	{1,2}, 3	One edge graph

10	1, {2,3}	One edge graph
11	{1,3}, 2	One edge graph
12	{1,2}, {2,3}	Two edges graph
13	{1,2}, {1,3}	Two edges graph
14	{1,3}, {2,3}	Two edges graph
15	**{1,2}, {2,3}, {1,3}**	**Three edges graph**
16	{1,1}, {1,2}, 3	One self-loop and one edge graph
17	{1,1}, {2,3}	One self-loop and one edge graph
18	{1,1}, {1,3}, 2	One self-loop and one edge graph
19	{1,1}, {1,2}, {2,3}	One self-loop and two edges graph
20	{1,1}, {1,2}, {1,3}	One self-loop and two edges graph
21	{1,1}, {1,3}, {2,3}	One self-loop and two edges graph
22	**{1,1}, {1,2}, {2,3}, {1,3}**	**One self-loop and three edges graph**
23	{2,2}, {1,2}, 3	One self-loop and one edge graph
24	1, {2,2}, {2,3}	One self-loop and one edge graph
25	{2,2}, {1,3}	One self-loop and one edge graph
26	{2,2}, {1,2}, {2,3}	One self-loop and two edges graph
27	{2,2}, {1,2}, {1,3}	One self-loop and two edges graph
28	{2,2}, {1,3}, {2,3}	One self-loop and two edges graph
29	**{2,2}, {1,2}, {2,3}, {1,3}**	**One self-loop and three edges graph**
30	{3,3}, {1,2}	One self-loop and one edge graph
31	{3,3}, {2,3}	One self-loop and one edge graph
32	{3,3}, {1,3}, 2	One self-loop and one edge graph
33	{3,3}, {1,2}, {2,3}	One self-loop and two edges graph
34	{3,3}, {1,2}, {1,3}	One self-loop and two edges graph
35	{3,3}, {1,3}, {2,3}	One self-loop and two edges graph
36	**{3,3}, {1,2}, {2,3}, {1,3}**	**One self-loop and three edges graph**
37	{1,1}, {2,2}, {1,2}, 3	Two self-loops and one edge graph
38	{1,1}, {2,2}, {2,3}	Two self-loops and one edge graph
39	{1,1}, {2,2}, {1,3}	Two self-loops and one edge graph
40	{1,1}, {2,2}, {1,2}, {2,3}	Two self-loops and two edges graph
41	{1,1}, {2,2}, {1,2}, {1,3}	Two self-loops and two edges graph
42	{1,1}, {2,2}, {1,3}, {2,3}	Two self-loops and two edges graph
43	**{1,1}, {2,2}, {1,2}, {2,3}, {1,3}**	**Two self-loops and three edges graph**
44	{1,1}, {3,3}, {1,2}	Two self-loops and one edge graph
45	{1,1}, {3,3}, {2,3}	Two self-loops and one edge graph
46	{1,1}, {3,3}, {1,3}	Two self-loops and one edge graph
47	{1,1}, {3,3}, {1,2}, {2,3}	Two self-loops and two edges graph

48	{1,1}, {3,3}, {1,2}, {1,3}	Two self-loops and two edges graph
49	{1,1}, {3,3}, {1,3}, {2,3}	Two self-loops and two edges graph
50	**{1,1}, {3,3}, {1,2}, {2,3}, {1,3}**	**Two self-loops and three edges graph**
51	{2,2}, {3,3}, {1,2}	Two self-loops and one edge graph
52	{2,2}, {3,3}, {2,3}	Two self-loops and one edge graph
53	{2,2}, {3,3}, {1,3}	Two self-loops and one edge graph
54	{2,2}, {3,3}, {1,2}, {2,3}	Two self-loops and two edges graph
55	{2,2}, {3,3}, {1,2}, {1,3}	Two self-loops and two edges graph
56	{2,2}, {3,3}, {1,3}, {2,3}	Two self-loops and two edges graph
57	**{2,2}, {3,3}, {1,2}, {2,3}, {1,3}**	**Two self-loops and three edges graph**
58	{1,1}, {2,2}, {3,3}, {1,2}	Three self-loops and one edge graph
59	{1,1}, {2,2}, {3,3}, {2,3}	Three self-loops and one edge graph
60	{1,1}, {2,2}, {3,3}, {1,3}	Three self-loops and one edge graph
61	{1,1}, {2,2}, {3,3}, {1,2}, {2,3}	Three self-loops and two edges graph
62	{1,1}, {2,2}, {3,3}, {1,2}, {1,3}	Three self-loops and two edges graph
63	{1,1}, {2,2}, {3,3}, {1,3}, {2,3}	Three self-loops and two edges graph
64	**{1,1}, {2,2}, {3,3}, {1,2}, {2,3}, {1,3}**	**Three self-loops & three edges graph**

4-vertex graph space

Let us consider a case study of four vertices 1, 2, 3 and 4 belonging to the set of vertices V. One can build a set of 1024 graphs using these four vertices. Let us denote an edge by, say, {1,2} where 1 and 2 are vertices. Self-loops are also considered here. In a 4-vertex graph, m = 6, which means that one can construct ${}^4C_2 = 6$ maximum number of edges. The combinatorics of 6 edges having one edge at a time, two edges at a time, and so on up to 6 edges at a time amounts to ${}^6C_1+{}^6C_2+{}^6C_3+{}^6C_4+{}^6C_5+{}^6C_6=6+15+20+15+6+1 = 63$ edge graphs. In a 4-vertex graph one can have the combinations of one self-loop at a time, two self-loops at a time and so on up to 4 self-loops at a time, which amounts to ${}^4C_1+{}^4C_2+{}^4C_3\ {}^4C_4 = 4+6+4+1 = 15$ self-loops. For every self-loop, one can associate 63 edge graphs. This means one can construct 15x63 = 945 graphs with self-loops, apart from 15 self-loops graphs and 63 edge graphs without self-loops. This amounts to saying that one can construct 945+63+15 = 1023 graphs using four vertices. One can also have just one graph consisting of four vertices but with no edges. So, one can construct a total of 1023+1 = 1024 graphs from four labelled vertices. This is called as **4-vertex graph space**. (NOTE: $2^{10} = 1024$)

5-vertex graph space

Let us consider a case study of five vertices 1, 2, 3, 4 and 5 belonging to the set of vertices V. One can build a set of 32,768 graphs using these five vertices. Let us denote an edge by, say, {1,2} where 1 and 2 are vertices. Self-loops are also considered here. In a 5-vertex graph, m = 10, which means that one can construct ${}^5C_2 = 10$ maximum number of edges. The combinatorics of 10 edges having one edge at a time, two edges at a time, and so on up to 10 edges at a time amounts to ${}^{10}C_1+{}^{10}C_2+{}^{10}C_3+{}^{10}C_4+{}^{10}C_5+{}^{10}C_6+{}^{10}C_7+{}^{10}C_8+{}^{10}C_9+{}^{10}C_{10} = 10+45+120+210+252+210+120+45+10+1 = 1023$ edge graphs. In a 5-vertex graph one can have the combinations of one self-loop at a time, two self-loops at a time and so on up to 5 self-loops at a time, which amounts to ${}^5C_1+{}^5C_2+{}^5C_3+{}^5C_4+{}^5C_5 = 5+10+10+5+1 = 31$ self-loops. For every self-loop, one can associate 1023 edge graphs. This means one can construct 31x1023 = 31,713 graphs apart from 31 self-loops graphs and 1023 edge graphs. This amounts to saying that one can construct 31,713+1023+31 = 32,767 graphs using five vertices. One can also have just one graph consisting of three vertices but with no edges. So, one can construct a total of 32,767 +1 = 32,768 graphs from five labelled vertices. This is called as **5-vertex graph space**. (NOTE: $2^{15} = 32{,}768$)

6-vertex graph space

Let us consider a case study of six vertices 1, 2, 3, 4, 5 and 6 belonging to the set of vertices V. One can build a set of 2,097,152 graphs using these six vertices. Let us denote an edge by, say, {1,2} where 1 and 2 are vertices. Self-loops are also considered here. In a 6-vertex graph, m = 15, which means that one can construct ${}^6C_2 = 15$

maximum number of edges. The combinatorics of 15 edges having one edge at a time, two edges at a time, and so on up to 15 edges at a time amounts to ${}^{15}C_1+{}^{15}C_2+{}^{15}C_3+{}^{15}C_4+{}^{15}C_5+{}^{15}C_6+{}^{15}C_7+{}^{15}C_8+{}^{15}C_9+{}^{15}C_{10}+{}^{15}C_{11}+{}^{15}C_{12}+{}^{15}C_{13}+{}^{15}C_{14}+{}^{15}C_{15}$ = 15+105 +455 +1365 +3003 +5005 +6435 +6435 +5005+3003+ 1365+455+105+15+1 = 32,767 edge graphs. In a 6-vertex graph one can have the combinations of one self-loop at a time, two self-loops at a time and so on up to 6 self-loops at a time, which amounts to ${}^{6}C_1+{}^{6}C_2+{}^{6}C_3+{}^{6}C_4+{}^{6}C_5+{}^{6}C_6$ = 6+15+20+15+6+1 = 63 self-loops. For every self-loop, one can associate 32,767 edge graphs. This means one can construct 63x32,767 = 2,064,321 graphs apart from 63 self-loops graphs and 32,767 edge graphs. This amounts to saying that one can construct 2,064,321+32,767+63 = 2,097,151 graphs using six vertices. One can also have just one graph consisting of six vertices but with no edges. So, one can construct a total of 2,097,151+1 = 2,097,152 graphs from six labelled vertices. This is called as **6-vertex graph space**. (NOTE: 2^{21} = 2097152). One may generalize this to **n-vertex graph space**.

n-vertex graph space

In a n-vertex graph, one can construct ${}^{n}C_2$ edges. The combinatorics of ${}^{n}C_2$ edges having one edge at a time, two edges at a time, and so on up to ${}^{n}C_n$ edges at a time amounts to

$$\sum_{k=1}^{\binom{n}{2}} \binom{\binom{n}{2}}{k} \quad \text{edge graphs without self-loops.}$$

In an n-vertex graph, one can have the combinations of one self-loop at a time, two self-loops at a time and so on up to n self-loops at a time amounts to

$${}^{n}C_1+{}^{n}C_2+{}^{n}C_3+{}^{n}C_4+..+{}^{n}C_n = \sum_{k=1}^{n}\binom{n}{k} \quad \text{self-loops.}$$

For every self-loop, one can associate

$$\sum_{k=1}^{\binom{n}{2}} \binom{\binom{n}{2}}{k} \quad \text{edge graphs.}$$

This means one can construct

$$\sum_{k=1}^{n}\binom{n}{k} \sum_{k=1}^{\binom{n}{2}} \binom{\binom{n}{2}}{k} \quad \text{graphs with self-loops.}$$

One can have just one graph consisting of n vertices with no edges.

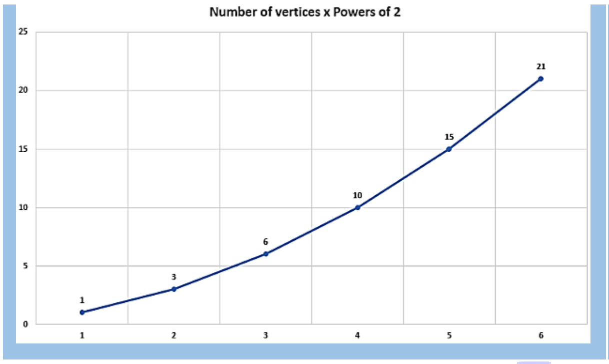
So, one can construct

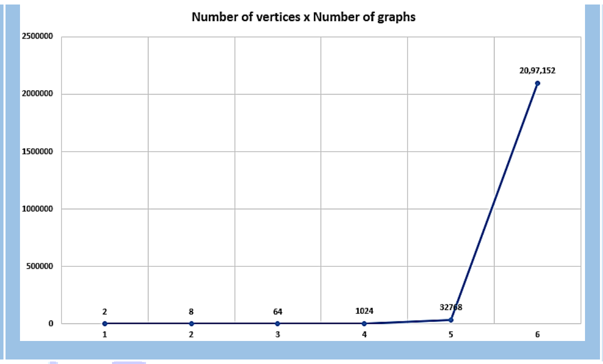
$$\sum_{k=1}^{n}\binom{n}{k} + \sum_{k=1}^{\binom{n}{2}} \binom{\binom{n}{2}}{k} + \sum_{k=1}^{n}\binom{n}{k} \sum_{k=1}^{\binom{n}{2}} \binom{\binom{n}{2}}{k} + 1 \quad \text{graphs from n labelled vertices.}$$

This is called as n-vertex graph space.

No. of Vertices	Exponents of 2	Powers of 2	No. of Graphs
1	1	2^1	2
2	3	2^3	8
3	6	2^6	64
4	10	2^{10}	1,024
5	15	2^{15}	32,768
6	21	2^{21}	20,97,152
7	28		
8	36		
9	45		
10	55		
11	66		
12	78		
	and so on		
	↓	↓	↓

NOTE: As the number of vertices increases linearly, the exponents of 2 increase exponentially. If ‘p’ is number of vertices and ‘q’ is the exponent of 2, then 2^q number of graphs could be constructed using ‘p’ vertices. Then for the p+1 vertices, the exponent of 2 would be (q+p+1). Then, one can construct $2^{(q+p+1)}$ graphs using p+1 vertices. A table showing number of vertices and total number of graphs using the formula $2^{(q+p+1)}$ is given on the left. Graph given below on the left shows number of vertices and a formula that determines total number of graphs as power of 2. Graph given below on the right shows the total number of graphs as against every set of vertices.





4. Formulation of the Notion of Graph Space

$$\lim_{n\to\infty} \sum_{k=1}^{n}\binom{n}{k} + \sum_{k=1}^{\binom{n}{2}}\binom{\binom{n}{2}}{k} + \sum_{k=1}^{n}\binom{n}{k}\sum_{k=1}^{\binom{n}{2}}\binom{\binom{n}{2}}{k} + 1 = G_\infty$$

As n tends to infinity, the resulting abstract infinite vertex graph space is called **Graph Space G_∞**. In philosophical terms, one can visualize **Graph Space G_∞** as **'Cosmic Network'**. All networks are subspaces of **Graph Space G_∞**, be it a neural network, or electrical network, or a traffic network, etc. Various graph theoretic operations can be performed in **Graph Space G_∞**.

Markov claims that the 'infinite' is introduced in mathematics by abstraction (idealization). He distinguishes between the "unclear" *abstraction of the actual infinity*, which is used to introduce (unintuitable) complete infinite totalities, and the *abstraction of potential realizability* that abstracts away from any practical spatial, temporal or material limitations in our capacity of constructing (concrete or abstract) mathematical objects. This abstraction enables us to conduct reasoning on as lengthy constructive processes and as large constructive objects as required. Thereby, as constructive objects can be considered only those, which are not generated by abstractions more powerful than the abstraction of potential realizability. *https://journals.openedition.org/philosophiascientiae/ 1054*. Markov assumes a philosophical stand about abstractions in the late 1950s: Abstractions are necessary in mathematics; however, they must not be devised for their own sake and lead where there is no return down to "earth". We should always remember to pass from abstract thinking to practice, as a necessary step of human cognition of objective reality. In case that the possibility of such a passage is turned out to be too doubtful, it is necessary to reconsider the abstractions applied and try to modify them. Proceeding in line with this thesis, he understands LEJ Brouwer's mental constructions as potentially realizable, since they have realizable material constructions as archetypes. In this way, Markov actually reinterprets Brouwer's idea of *potential infinite* in terms of his own concept of abstraction of potential realizability, in an attempt to "return down to earth".

5. Concluding Remarks

Using Markov's philosophy, the notion of 'Graph Space' is potentially realizable and constructive. This amounts to saying that the Graph Space **G_∞** is a constructive analogue of what is already known as 'Hilbert Space', which is nonconstructive.

References

1. Billera, L. J., S. P. Holmes, and K. Vogtmann (2001). Geometry of the space of phylogenetic trees. Advances in Applied Mathematics 27(4), 733–767.
2. Bunke, H. and K. Riesen (2011). Improving vector space embedding of graphs through feature selection algorithms. Pattern Recognition 44(9), 1928–1940.
3. Calissano, A., A. Feragen, and S. Vantini (2020). Graphspace python package. https://github.com/annacalissano/ GraphSpace.git.
4. Chowdhury, S. and F. Memoli (2018). The metric space of networks. ´ arXiv preprint arXiv:1804.02820.
5. Conte, D., P. Foggia, C. Sansone, and M. Vento (2004). Thirty years of graph matching in pattern recognition. International journal of pattern recognition and artificial intelligence 18(03), 265–298.
6. Feragen, A., F. Lauze, P. Lo, M. de Bruijne, and M. Nielsen (2010). Geometries on spaces of treelike shapes. In Asian Conference on Computer Vision, pp. 160–173. Springer.
7. Kolaczyk, E., L. Lin, S. Rosenberg, J. Xu, and J. Walters (2017). Averages of unlabeled networks: Geometric characterization and asymptotic behavior. arXiv preprint arXiv:1709.02793.

8. Nye, T. M. W. (2014). An algorithm for constructing principal geodesics in phylogenetic treespace. IEEE/ACM Trans. Comput. Biology Bioinform. 11(2), 304–315.
9. Sanfeliu, A. and K.-S. Fu (1983). A distance measure between attributed relational graphs for pattern recognition. IEEE transactions on systems, man, and cybernetics 3, 353–362. 34
10. Sturm, K. T. (2003). Probability measures on metric spaces of nonpositive. Heat Kernels and Analysis on Manifolds, Graphs, and Metric Spaces 338, 357–391.

About the Authors

Dr. V. Krishnan is an Assistant Professor of Mathematics at Jamal Mohamed College, Trichy, with over 15 years of teaching experience. He received his Ph.D. in Mathematics from Srimad Andavan Arts and Science College (2019) and completed post-doctoral research at Pentagram Technologies, Hyderabad (2024). His academic journey includes an M.Sc. (2009), M.Phil. (2010), and B.Sc. in Mathematics (2007), all affiliated with Bharathidasan University. He has published 11 research papers in national and international journals, authored a book chapter (2023), and he actively participates in seminars and workshops. As a life member of IMS, RMS, and MALA, he also supervises four part-time research scholars. His expertise spans topology, probability, operations research, and numerical methods, mathematical modelling and graph theory.

Z. Sirajunisha is an Assistant Professor of Mathematics at Jamal Mohamed College, Trichy, with over 8 years of teaching experience. I pursuing Ph.D. in, Mathematics at Jamal Mohamed College, under guidance of Dr. V. Krishnan. Her academic journey includes an M.Sc. (2014), M.Phil. (2015), and B.Sc. in Mathematics (2009), all affiliated with Bharathidasan University. I have published 7 research papers in national, international journals and proceedings with ISBN number and actively participate in seminars ,workshops and Refresher Course. Her expertise spans Statistics, probability, operations research, Number theory and numerical methods, with research interests graph theory.