Reprint

ISSN 0973-9424

INTERNATIONAL JOURNAL OF MATHEMATICAL SCIENCES AND ENGINEERING APPLICATIONS

(IJMSEA)

ASCENT PUBLICATION
@
ASCENT
PUNE, INDIA

www.ascent-journals.com

International J. of Math. Sci. & Engg. Appls. (IJMSEA)
ISSN 0973-9424, Vol. 11 No. I (April, 2017), pp. 205-217

A SURVEY ON INTERCONNECTION NETWORKS: HANOI GRAPHS, BUTTERFLY GRAPHS AND SIERPINSKI GRAPHS

MANJUNATH A. S.[1] **and SUDIN S.**[2]
[1] Department of Mathematics, Government College Kottayam, Kerala, India
[2]Department of Mathematics, Government Engineering College, Idukki, Kerala, India.

Abstract

An interconnection network is a physical or virtual link between two or more networks, playing a crucial role in communication and computing systems. This survey paper provides a brief overview of three significant interconnection networks: Hanoi graphs, Butterfly graphs, and Sierpinski graphs.

1. Introduction

Interconnection networks [1] are essential to the effectiveness and performance of computer systems in the current day, especially those that are distributed and operate in parallel. The requirement for efficient communication methods between a system's different components has grown along with computer designs. By offering channels for data to move between processors, memory modules, and other system components, interconnection networks make this communication easier. Because they facilitate effective communication between processors, memory, and other devices, interconnection networks are essential to contemporary computer systems.

Key Words and Phrases : *Hanoi graph, Butterfly graph, Sierpinski graph.*
2000 AMS Subject Classification : 05C10, 68R10

Scalable, low-latency, and fault-tolerant interconnection networks are becoming more and more important as computing needs rise. These networks will keep developing, pushing the limits of computational power and efficiency, whether in data centers, distributed systems, or parallel processing systems. Their topology, architecture, and routing tactics will continue to play a significant role in determining how high-performance computing develops in the future.

An undirected graph with vertices representing CPUs or memory modules and edges representing communication links is typically used to model an interconnection network [2]. It is well recognized that the best metrics to assess the stability and synchronizability of an interconnection network are the symmetric properties of a graph, such as vertex regularity, vertex transitivity, edge transitivity, and arc transitivity. This review article attempts to present a brief survey of three important interconnection networks viz. Hanoi graphs, Butterfly graphs and Sierpinski graphs.

The renowned Tower of Hanoi puzzle [3] served as the inspiration for the intriguing mathematical constructions known as Hanoi graphs. A stack of discs of various sizes must be moved from one peg to another, according to certain criteria in this puzzle, which was first created in 1883 by the French mathematician Édouard Lucas. Although the challenge is well-known, its graph-theoretical representation—known as Hanoi graphs [4]—offers important new information about the puzzle and graph theory. By representing the Tower of Hanoi problem's allowed moves as vertices and edges, these graphs provide a fresh viewpoint on the problem's intricacy and wider mathematical and computer science ramifications [5].

The butterfly graph [6] is a prominent structure in graph theory that finds use in both theoretical mathematics and real-world computers. This graph, so named because it resembles a butterfly, is essential to several fields of study, such as error-correcting codes, network architecture, and parallel computing. With important uses in network architecture, parallel computing, and communication systems, butterfly graphs are a strong and adaptable tool in graph theory. They are perfect for simulating effective data transmission and routing in intricate networks because of their special qualities, which include high connectedness, small diameter, and regularity. Butterfly graphs [7] will probably continue to be at the forefront of theoretical and applied study as technology develops, spurring advancement in domains like distributed systems, cloud computing, and telecommunications.

Inspired by the well-known Sierpiński triangle, a fractal named for the Polish mathematician Wacław Sierpiński, the Sierpiński graph [8] is an intriguing structure in graph theory. From computer science to mathematics, mathematicians and scientists have been fascinated by the Sierpiński triangle because of its self-similar and recursive patterns. Sierpiński graphs provide a framework for examining intricate, recursive patterns in graph theory, bringing the concept of fractals to the field of discrete mathematics.

The second, third and fourth section deals with Hanoi graphs, Butterfly graphs and Sierpinski graphs respectively. For all basic definitions not mentioned in this article we refer to [9].

2. Hanoi Graph

With the vertices representing potential disc configurations and the edges representing permissible moves between them, a Hanoi graph [10] is a graph-theoretic model of the Tower of Hanoi problem. The Hanoi graph $H(n, k)$, in where n is the number of discs and k is the number of puzzle pegs, is the most well studied variant of these graphs. According to the Tower of Hanoi's principles, each vertex in the graph represents a legitimate configuration of the discs across the pegs, and each edge joins two configurations if they can be reached from one another by moving precisely one disc. According to the rules, a disc can only be placed on a larger disc or an empty peg, and no disc may be placed on a smaller one. Additionally, only one disc may be moved at a time.

For example, in the classic Tower of Hanoi puzzle with three pegs ($k = 3$) and three disks ($n = 3$), the graph $H(3, 3)$, (Figure 1), consists of all the possible configurations of the three disks spread across the three pegs, with edges representing the valid moves between these configurations. The number of vertices in a Hanoi graph is determined by the number of possible arrangements of the disks on the pegs. In the case of $H(n, k)$, there are k^n possible configurations, since each of the n disks can be placed on any of the k pegs. This exponential growth makes Hanoi graphs large and complex as the number of disks or pegs increases.

This class of graphs were investigated by many authors. Among them, the pioneer was A.M Hinz [11,12,13]. Parisse et. al [14,15] studied the planarity and colouring problem of Hanoi graphs. The following are some of the interesting properties of Hanoi graphs:

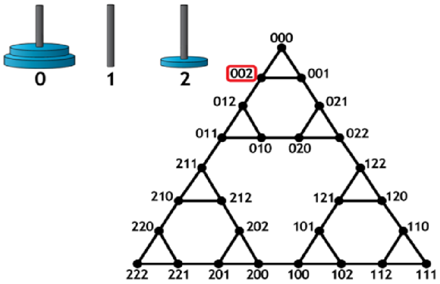

Figure 1: The Hanoi graph $H(3,3)$. The highlighted vertex corresponds to the Tower of Hanoi puzzle in the state shown.

- The recursive nature of Hanoi graphs is among their most intriguing characteristics. The recursive structure of the graph reflects the recursive nature of the Tower of Hanoi puzzle, where solving the problem for n discs frequently entail solving smaller subproblems for $n-1$ discs. In particular, the lawful configurations of the first $n-1$ discs can be represented by smaller sub-graphs of the Hanoi graph for n discs.

- The graph's connectedness is another important characteristic. There is a path connecting any two vertices in the graph since the Tower of Hanoi puzzle's objective is to move every disc from one peg to another. But depending on the number of discs involved, the shortest path between two vertices in the graph means the fewest number of moves needed to change from one configuration to another. The shortest path length on the graph, or the smallest number of steps needed to solve the problem for n discs, is known to be $2^{(n-1)}$.

- Furthermore, symmetries are present in Hanoi graphs, particularly when there are more than three pegs. In terms of the puzzle's mechanics, these symmetries result from the pegs' indistinguishability from one another. The result is a fairly symmetric graph with certain isomorphisms between subgraphs.

While Hanoi graphs originate from a recreational puzzle, they have far-reaching applications in computer science, combinatorics, and discrete mathematics.

- Algorithms and Recursion: The Tower of Hanoi problem is a well-known example of a recursive algorithm in computer science due to its recursive structure. Researchers can examine the effectiveness of these algorithms in terms of pathfinding and optimization thanks to the graph representation of the problem. The Tower of Hanoi puzzle, in particular, offers a good setting for investigating shortest-path algorithms since the shortest path in the graph corresponds to the minimal number of moves needed to complete it.

- Combinatorial Optimization: In the realm of combinatorial optimization, Hanoi graphs are also interesting. Similar graph topologies can be used to model finding optimal solutions to problems involving transitions between states, such scheduling or resource allocation. As a minimal-move problem, the Tower of Hanoi puzzle provides a straightforward yet effective illustration of these optimization issues.
- Network Design and Routing: Hanoi graphs are used in network theory as models for specific network design issues where reducing the number of steps or transitions between various network states is the aim. Routing and flow problems in networks, where resources must be transported in accordance with predetermined rules, are comparable to the Tower of Hanoi problem's restrictions on moving one disc at a time.

3. Butterfly Graph

A butterfly graph, also known as a Benes graph or de Bruijn graph in some contexts, is a directed graph that is constructed to model data flow in interconnection networks, especially those used in parallel computing systems. The most common version is the butterfly network [6] , which is a special type of multistage interconnection network (MIN). The butterfly graph is typically defined based on two key parameters:

- The number of levels or stages, denoted by n, which corresponds to the number of nodes or vertices in each level.
- The number of nodes in each level, which is often $2^{(n-1)}$.

For example, in a butterfly graph with 3 levels, BF (3), (Figure 2) each level will have 8 nodes (since $2^3 = 8$), and each node in one level is connected to exactly two nodes in the next level. The resulting graph has highly regular structure, with connections (or edges) between vertices following a systematic pattern that ensures efficiency in data routing and communication.

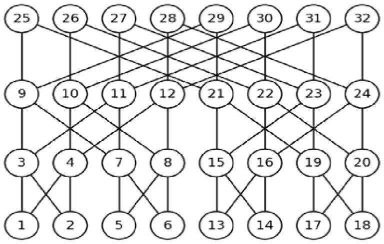

Figure 2: The Butterfly graph $BF(3)$

Each vertex in a butterfly graph can be labelled by a pair (i, x), where i represents the stage (or level) and x represents the node within that level [16]. The directed edges between vertices define the rules for transitions from one stage to the next, often with specific constraints, such as routing based on bit sequences or specific positional patterns. A butterfly graph is frequently visualised as a collection of interconnected layers, where each layer stands for a step in data transmission or a stage in a calculation. The way connections fan out from a source, converge at intermediate nodes, and then spread out again, resembling a butterfly's wings and body, gives rise to the "butterfly" shape.

Butterfly graphs have several important mathematical properties that make them valuable in theoretical and applied contexts:

- Bipartite Structure: A butterfly graph's vertices can be separated into two distinct sets so that no two vertices in the same set are contiguous. This is known as a bipartite structure. Applications in parallel computing, where work must be split equally among several processors or stages without redundancy or overlap, depend heavily on this feature.

- High Connectivity [17]: There are several pathways connecting any two nodes in the butterfly network, indicating its high connectivity. Reliability in network systems depends on the graph's fault tolerance, which is ensured by its strong connectedness. Data can still get to its destination through other channels even if one connection fails.

- Regularity and Symmetry: Depending on the network's stage, each vertex in a butterfly graph has an equal amount of incoming and outgoing edges, indicating that the graph is extremely regular. This regularity makes it easier to construct network algorithms that depend on dependable communication patterns and guarantees predictable performance.
- Recursive Structure: The recursive structure of butterfly graphs is another important characteristic. Butterfly graphs are appropriate for hierarchical communication networks and recursive algorithms because they may be created by joining two smaller graphs for $n - 1$ levels to create a graph for n levels.
- Small Diameter: The butterfly graph's tiny diameter in relation to its node count is one of its main advantages. The greatest number of edges that must be crossed to travel from one vertex to any other vertex is known as a graph's diameter. Butterfly graphs are effective for data transmission in big networks since this value is logarithmic with regard to the number of nodes.

Butterfly graphs have numerous applications in computer science, particularly in areas involving communication, network design, and parallel computing.

- Data Routing and Switching: Butterfly graphs are used in data routing and telecommunications to simulate the information flow via switches and routers. The graph's modest diameter and strong connectedness make it perfect for creating fault-tolerant networks that can effectively route data even when there are network outages.
- Distributed Systems and Cloud Computing: Butterfly graphs offer an effective means of arranging communication between nodes in distributed computing systems, when tasks are dispersed among several computers or nodes. They are frequently used for network topologies in cloud computing settings because of their recursive structure, which guarantees fast and dependable data transmission.
- Genomic Sequencing: Overlaps in DNA sequences can be modelled using butterfly-like structures in bioinformatics, namely in genomic sequencing. This enables

scientists to use the recursive and layered structure of the network for effective computational analysis, reassembling bigger DNA sequences from smaller bits.

- Error-Correcting Codes: The construction of error-correcting codes is one area of coding theory where butterfly graphs are very pertinent. Butterfly graphs are ideal for creating graph-based codes, which are used to identify and fix faults in data transmission over noisy channels, due to their high degree of connectedness and redundancy.
- Parallel Computing Networks: Creating multistage interconnection networks for parallel computing systems is one of the main uses for butterfly graphs. These graphs offer a productive framework for establishing a balanced data flow and reducing communication delays between processors and memory units. Fast Fourier transform (FFT) methods, where the recursive structure of the graph corresponds with the recursive nature of the calculation, make extensive use of butterfly networks in particular.

4. Sierpinski Graph

A Sierpiński graph is a recursive graph that resembles the structure of the Sierpiński triangle. It is represented by $S(n)$, where n is the graph's order. Recursive subdivision can be used to construct it, much like the Sierpiński triangle, which is made by gradually eliminating equilateral triangles from a starting big triangle.

The Sierpiński graph $S(n)$ [18] of order n is defined based on a complete graph of three vertices, denoted as K_3, for the base case when $n = 1$. As the order increases, the graph expands recursively. Specifically, for each successive order n, $S(n)$ is constructed by taking three copies of $S(n-1)$ and connecting them in a manner that mirrors the recursive structure of the Sierpiński triangle. This process results in a self-similar, fractal-like graph that grows exponentially in size with each iteration. The three vertices and three edges of the entire graph K_3 make up $S(1)$ in the basic case. The recursive construction process causes the number of vertices and edges to increase at higher levels. As n increases, more complicated structures are created since the number of vertices in $S(n)$ may be stated as 3^n, and the number of edges increases proportionately. The recursive and self-similar structure of Sierpiński graphs (Figure 3) is one of its distinguishing features. $S(n-1)$ is exactly duplicated in smaller subgraphs found in

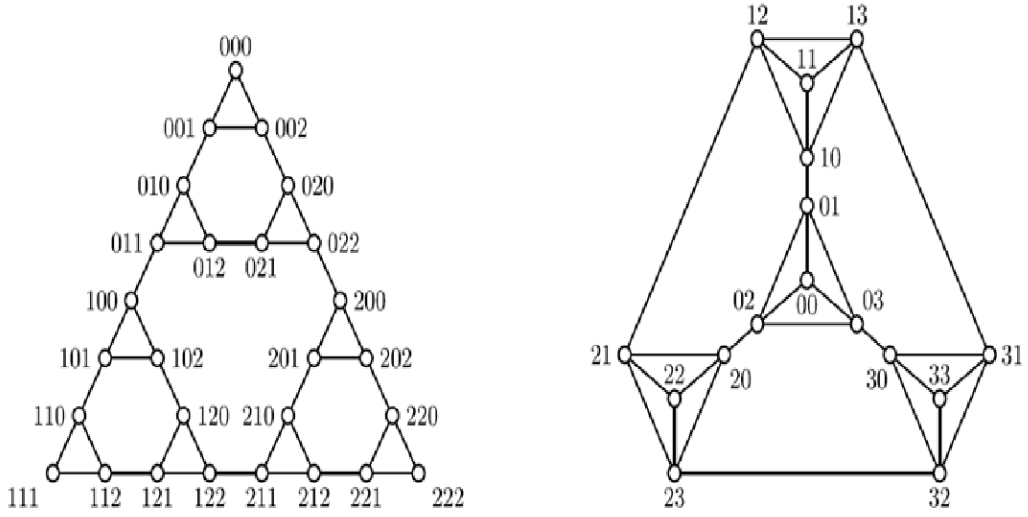

Figure 3: Sierpiński graph S_3^2 (left) and S_4^2 (right)

every graph of rank n. The fractal geometry of the Sierpiński triangle, in which each triangular unit contains smaller triangles, repeats indefinitely, is directly analogous to this recursive nature. One interesting property of Sierpiński graphs is that they are planar, meaning that they can be embedded in the plane without any edges crossing. This makes them easier to visualize and analyze in two dimensions, which is particularly useful when studying their properties and applications.

Sierpiński graphs exhibit several key properties that make them an important topic of study in graph theory, particularly in relation to fractals and recursive structures [19, 20].

- Hamiltonian Paths and Cycles: A Hamiltonian path in a graph is a path that visits each vertex exactly once, and a Hamiltonian cycle is a cycle that visits each vertex exactly once and returns to the starting point. Sierpiński graphs are Hamiltonian, meaning that they contain Hamiltonian paths and cycles. This property is significant in optimization problems, such as the traveling salesperson problem, where the goal is to find the shortest possible route that visits each point exactly once.

- Self-Similarity: A characteristic shared by all fractals, self-similarity is the most notable aspect of Sierpiński graphs. A graph is said to be self-similar if any portion of it resembles the full graph when scaled correctly. Sierpiński graphs are characterised by this recursive structure, which enables sophisticated mathematical analysis through the use of recursive algorithms.

- Fractal Dimension: Sierpiński graphs, like their geometric counterparts, can be linked to a fractal dimension that measures their complexity as they get bigger. This idea is relevant for discrete structures like graphs, although it is more commonly used for continuous fractals like the Sierpiński triangle. A Sierpiński graph's fractal dimension, which is a measurement of the graph's recursive growth pattern, can be calculated.

- Diameter: The largest distance between any two vertices, expressed in terms of the number of edges in the shortest path between them, is a graph's diameter. The efficiency of the structure of Sierpiński graphs is reflected in the width, which increases logarithmically with the number of vertices. In particular, the diameter of $S(n)$ is proportional to 2^n, which implies that the longest distance between two vertices rises relatively slowly even as the graph grows exponentially.

- Graph Colouring: One of the key topics in graph theory is graph colouring, which is the process of giving a graph's vertices different colours so that no two neighbouring vertices have the same colour. Colouring problems are strongly related to Sierpiński graphs, such as the Sierpiński triangle. The Sierpiński's chromatic number of a Sierpiński graph (the minimum number of colors required to color the vertices) is 3, reflecting the graph's triangular base structure.

Sierpiński graphs, like other fractal-based graphs, have a range of applications in both theoretical and applied fields. Their recursive structure and self-similar nature make them ideal for modelling complex systems and solving problems in diverse domains.

- Algorithm Design: Recursive algorithms are frequently tested and developed using Sierpiński graphs. The form of the graph naturally allows for recursive decomposition, making it the perfect environment for investigating the effectiveness of algorithms created to address issues with recursive data structures. This is especially helpful in theoretical computer science, where recursive structures are frequently employed in data processing, and computer graphics, where fractal designs are created using recursive algorithms.

- Combinatorial Optimisation: Sierpiński graphs' Hamiltonian characteristics make them applicable to combinatorial optimisation issues like network routing and the travelling salesperson problem. These issues entail determining the best routes or cycles over a network, and Sierpiński graphs' effective structure offers a helpful paradigm for creating answers to these challenges.
- Communication and Network construct: Sierpiński graphs are used to construct effective communication networks, especially when redundancy and fault tolerance are crucial. Because of their recursive, highly linked topology, which permits many paths between any two vertices, the network can continue to function even in the case of failures. Because of this characteristic, they can be used to create scalable, reliable networks for communication and data transfer.
- Fractal Geometry and Modelling: Sierpiński graphs are a discrete counterpart of fractal geometry, which makes them helpful for simulating fractal-like natural events. Geological formations, biological structures, and even certain patterns in social networks and economy are examples of these phenomena. Through the study of Sierpiński graphs, scholars can learn more about the fundamental ideas behind these intricate systems.
- Parallel and Distributed Computing: Sierpiński graphs are ideal for modelling hierarchical systems, such those in parallel and distributed computing, because of their recursive nature. Similar to the graph's recursive structure, tasks in these systems are frequently broken down into smaller ones that are handled separately. These systems' communication networks can be modelled using Sierpiński graphs, which maximise data flow and reduce latency.

5. Conclusion

Even though the Tower of Hanoi puzzle is straightforward, there are a number of difficult mathematical and computer science problems with the accompanying Hanoi graphs. For example, the graph's complexity increases exponentially with the number of discs or pegs, creating challenging computation, optimization, and traversal issues. In various variations, including ones with more than three pegs (generalized Tower of Hanoi), researchers are still investigating effective methods for solving the puzzle. There are other unanswered concerns about how to characterize Hanoi graphs, especially when it comes to graph embeddings and isomorphisms. Gaining knowledge of these graphs' structural characteristics may help advance theoretical and applied graph theory.

Butterfly graphs are beneficial, but there are a number of theoretical and practical issues with them. Making sure it's scalable is one of the biggest obstacles. Maintaining the characteristics that make butterfly graphs effective gets harder as the network gets bigger, especially in real-world systems where physical limitations like latency and bandwidth are present. The optimization of routing algorithms for butterfly networks is another topic of current research. Although the structure of the graph facilitates effective communication, research is still ongoing to create algorithms that can fully utilize the graph's potential, particularly in the fields of high-performance computing and next-generation network design.

Combining the intricacy of recursive graph topologies with the beauty of fractal geometry, Sierpiński graphs are a rich and fascinating topic in graph theory. They are an effective tool for researching a variety of issues in network theory, computer science, and mathematics because of their self-similar patterns and recursive growth. The nature of complicated systems and the techniques used to traverse and optimize them will probably become clearer as more research is done on these graphs. Sierpiński graphs provide a strong framework for comprehending and resolving complicated problems, whether they are in network architecture, parallel computing, or combinatorial optimization.

References

[1] José Duato, Sudhakar Yalamanchili and Lionel Ni, Interconnection Networks-An Engineering Approach, A volume in :The Morgan Kaufmann Series in Computer Architecture and Design, 2003

[2] Akers SB, Krishnamurthy B (1989) A group-theoretic model for symmetric interconnection networks. IEEE Trans Comput C-38(4):555–566

[3] X. Chen, J. Shen, On the Frame-Stewart conjecture about the Towers of Hanoi, SIAM J. Comput. 33 (2004), 584–589

[4] A.M. Hinz, S. Klavžar, U. Milutinović, C. Petr, The Tower of Hanoi—Myths and Maths, Birkhäuser/Springer, Basel, 2013.

[5] J.S. Frame, Solution to advanced problem 3918, Amer. Math. Monthly 48 (1941) 216–217.

[6] F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays Trees Hypercubes, Computer Science, Mathematics Electrical Engineering, Morgan Kaufmann, Los Altos, CA, 1992.

[7] R.S. Scorer, P.M. Grundy, C.A.B. Smith, Some binary games, Math. Gaz. 28 (1944) 96–103.

[8] S. Klavžar, U. Milutinović, Graphs S(n,k) and a variant of the Tower of Hanoi problem, Czechoslovak Math. J. 47 (122) (1997) 95–104

[9] D.B. West, Introduction to Graph Theory, Pearson Education Inc, 2001

[10] M.A. Alekseyev, T. Berger, Solving the tower of Hanoi with random moves, in: J. Beineke, J. Rosenhouse (Eds.), The Mathematics of Various Entertaining Subjects: Research in Recreational Math, Princeton University Press, Princeton NJ, 2016, pp. 65–79

[11] A.M. Hinz, The Tower of Hanoi, Enseign. Math. (2) 35 (1989) 289–321.

[12] A.M. Hinz, Graph theory of tower tasks, Behav. Neurol. 25 (2012) 13–22

[13] A.M. Hinz, A. Kostov, F. Kneißl, F. Sürer, A. Danek, A mathematical model and a computer tool for the Tower of Hanoi and Tower of London puzzles,Inform. Sci. 179 (2009) 2934–2947.

[14] A.M. Hinz, D. Parisse, On the planarity of Hanoi graphs, Expo. Math. 20 (2002) 263–268.

[15] A.M. Hinz, D. Parisse, Coloring Hanoi and Sierpiński graphs, Discrete Math. 312 (2012) 1521–1535.

[16] D. Barth, A. Raspaud, Two edge-disjoint hamiltonian cycles in the Butterfly graph, Inform. Process. Lett. 51 (1994) 175-179.

[17] J-C. Bermond, C. Peyrat, Broadcasting in de Bruijn networks, in: Proc. 19th S-E Conf. Combinat., Congr. Numer. 66, Graph theory, and Computing, Florida, 1988, pp. 283-292.

[18] A.M. Hinz, A. Schief, The average distance on the Sierpiński gasket, Probab. Theory Related Fields 87 (1990) 129–138.

[19] B. Xue, L. Zuo, G. Wang, G. Li, Shortest paths in Sierpiński graphs, Discrete Appl. Math. 162 (2014) 314–321.

[20] A.M. Teguia, A.P. Godbole, Sierpiński gasket graphs and some of their properties, Australas. J. Combin. 35 (2006) 181–192.