Journal of Innovations in Computer Science and Engineering
Shahid Beheshti Unversity

January 2024, Volume 1, Issue 2

Graph-Theoretic Analysis of de Bruijn Graphs: Fault Resilience and Fragility

F. Safaei[1]✉, ORCID: 0000-0002-8546-3148
M. M. Emadi Kouchak[2], ORCID: 0009-0009-2572-3356
M. Moudi[3], ORCID: 0000-0002-9081-5347
1 Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran, f_safaei@sbu.ac.ir,
2 Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran, m.m.emadi@srbiau.ac.ir,
3 Department of Computer Engineering, University of Torbat Heydarieh, Razavi Khorasan Province, Iran, mmoudi@torbath.ac.ir

ABSTRACT

The de Bruijn graph, initially proposed as a topological architecture for interconnection networks, offers unique attributes. These graphs are regular, Eulerian, and Hamiltonian, boasting a small diameter close to optimal connectivity. Their low average distance, small diameter, and high connectivity contribute to remarkable fault tolerance against both node and edge failures. Comprehensively, these graphs serve as pivotal components in word-representing networks, finding applications in various scientific and engineering domains, particularly in genome assembly. They play a significant role in bioinformatics, information theory, coding, communication networks, and multiprocessors. Additionally, de Bruijn graphs are utilized in peer-to-peer (P2P) networks and distributed hash tables (DHT), demonstrating their versatility. Moreover, de Bruijn graphs can serve as a robust infrastructure for modeling online/offline user behavior. In this article, we delve into the different types of de Bruijn graphs and their unique properties from a graph theory perspective. Our focus is on evaluating the reliability of these graphs concerning resilience, fragility, and vulnerability to random failures and targeted attacks on both nodes and edges.

Keywords: Graph Theory, Interconnection Networks, de Bruijn Graph, Fault-Tolerance, Graph Resilience, Network Fragility

1. Introduction

Complex networks serve as the structural foundation for a diverse array of systems, encompassing social, ecological, biological, and technological domains. The interdisciplinary exploration of these systems has emerged as a focal point in recent years, attracting scientists from diverse fields. By developing new models aligned with established principles in physics and engineering, researchers have made significant contributions to comprehending and analyzing the dynamic and topological properties of complex networks. These networks, which encompass communication, biological, and social systems, are often aptly represented as graphs. In this representation, nodes symbolize independent entities, while edges indicate the connections between these entities. The construction of a successful communication network necessitates a nuanced consideration of various aspects. Key factors include cost, security, integrity, scalability, and notably, fault tolerance. The latter is especially critical in communication networks, prompting extensive research in recent years on the robustness and resilience of graphs and networks within the broader context of complex networks. This emphasis on robustness and resilience underscores their paramount importance in ensuring the functionality and reliability of complex communication systems.

A significant concern in any complex network revolves around the overall robustness of the system against various types of failures. A robust system is defined by its ability to maintain its fundamental functions even in the face of failures. In network

Submit Date: 2024-10-15
Revise Date: 2025-01-13
Accept Date: 2025-02-19
✉ Corresponding author

science, the concept of robustness pertains to the system's capacity to execute its tasks even when nodes or edges are removed. Additionally, resilience in networks implies the ability of their entities to endure a myriad of challenges and disturbances [1,2].

Errors and failures may manifest either as a random error that disrupts an element within the graph or as a result of a systematic and intentional attack, where an external agent deliberately inflicts damage on the system. It is noteworthy that tolerance to these forms of disruption may not always be a fundamental property of all complex networks; it typically exists in specific classes of networks. The presence of such tolerance often hinges on the integrity of the graph and its underlying network. Given the dependence on the graph's integrity, addressing the robustness of a system involves analyzing topological changes and structural alterations induced by the removal of nodes and edges. This article delves into exploring the impact of such structural and topological changes, specifically through the random removal of nodes and edges within the context of de Bruijn graphs.

De Bruijn graphs, characterized by regularity, Eulerian, and Hamiltonian properties, boast a low diameter in close proximity to optimal connectivity (OC). Their small average distance contributes to high fault tolerance, particularly against node-type failures. Essentially, de Bruijn graphs can effectively sustain their functionality even when some nodes are removed, as long as critical nodes remain intact against targeted attacks. Moreover, the equality and small size of input and output degrees for nodes in the de Bruijn graph imply that each node requires only a few connections. Consequently, there is no necessity for multiple edges between arbitrary nodes in the graph. This attribute results in short distances between nodes, translating to a negligible average distance. This characteristic not only aids in reducing congestion within the network but also enhances and simplifies routing algorithms. Simultaneously, the de Bruijn graph's abundance of different routes between nodes proves beneficial in fault tolerance and traffic load balancing, especially during saturated workloads. This diversity in routes not only promotes resilience against faults but also ensures an equitable distribution of traffic load across the network.

This article is structured into eight sections. In Section 2, we conduct a comprehensive review of the research field and related work. Moving on to Section 3, we present essential definitions and preliminaries, introducing various types of de Bruijn graphs while examining their properties. Section 4 delves into generalized de Bruijn graphs. Section 5 explores a family of graphs known as word-representable graphs. These graphs, utilized in word combinations and scheduling, are introduced, and their relationship with de Bruijn graphs is explored. The robustness and resilience of de Bruijn graphs take center stage in Section 6. Here, we propose and investigate various criteria for evaluating and analyzing the robustness of graphs. Section 7 adopts an experimental approach, verifying the topological properties and robustness of de Bruijn graphs against node and edge failures through simulation experiments. Comparative analyses with results obtained from analytical models further validate our findings. Finally, Section 8 draws conclusions from the presented findings. Additionally, we provide guidelines and suggestions for the continuation of current research as future work, offering a roadmap for further exploration in this domain.

2. Background and Related Work

De Bruijn graphs stand out as particularly suitable for numerous engineering and science applications owing to their characteristics such as small average distance, high fault tolerance, high connectivity, and low diameter. Notably, it has been demonstrated in [2-6] that generalized de Bruijn graphs exhibit an optimal diameter asymptotically. Furthermore, an analysis of Moore's graph in [7] enables the derivation of a formula establishing a lower bound for the average distance in de Bruijn graphs. This criterion plays a crucial role in determining the responsiveness and capacity of the network, closely tied to the efficiency of routing algorithms within the graph.

Esfahanian and Hakimi [8] have detailed the applications of de Bruijn graphs in the context of interconnection and multi-processor networks. Additionally, Collins [9] has provided insights into the utilization of de Bruijn graphs for constructing a fully parallel Viterbi decoder. These instances underscore the versatility and practical utility of de Bruijn graphs in various technological and computational domains.

In the study by Faizian [10], an evaluation and analysis of regular random graphs (RRG), a specific type of directed regular graphs (DRG), was conducted. These graphs find application as interconnection topologies for large-scale data centers and high-performance computing (HPC) clusters. In [10], simulation results reveal that RRG networks, also known as Jellyfish topology [11], with *k*-shortest path routing exhibit less-than-ideal performance concerning diameter and load balance. In contrast, generalized de Bruijn graphs, the deterministic counterpart of DRG, demonstrate close-to-optimal network characteristics across various configurations, including diameter, average distance, and load balance.

Moreover, another report [12] compares generalized RRG and de Bruijn networks under different workload traffics, exploring their robustness. Loguinov et al. [5] investigate the applications of de Bruijn graphs in the infrastructure of P2P and distributed hash table (DHT) networks. The authors demonstrate that these graphs can serve as suitable infrastructures for implementing online/offline user models. Several other studies [13-15] have utilized de Bruijn graphs in the construction of P2P networks, showcasing their versatility and effectiveness in diverse network scenarios.

In [5], a study on the local and global clustering coefficient for de Bruijn graphs was conducted, revealing that this parameter tends to be relatively small. Another crucial metric, the bisection width, has been widely employed to estimate the robustness and

efficiency of communication networks [5]. In essence, this metric signifies the level of difficulty in disconnecting and separating networks. In [5], the bisection width measure for generalized de Bruijn graphs is determined, further contributing to our understanding of their structural characteristics.

The construction of genomes and the utilization of de Bruijn graphs in assembling short-read sequences have become topics of significant interest to researchers [4, 16, 17]. In recent years, de Bruijn graphs have played a pivotal role in bioinformatics, particularly in the realm of de novo assembly. De novo assembly is a fundamental challenge in bioinformatics, involving datasets aimed at reconstructing an unknown genome sequence from a collection of short sequenced fragments. Consequently, researchers in the field of molecular biology have extensively employed de Bruijn graphs to assemble billions of short sequencing reads into a single genome. This application underscores the critical role of de Bruijn graphs in advancing research in molecular biology and bioinformatics.

The construction of a directed de Bruijn graph is a common starting point for most state-of-the-art assemblers. However, a significant computational bottleneck arises in storing and retrieving information for displaying de Bruijn graphs in memory. This bottleneck becomes particularly pronounced for many assemblers, as their workflows often require the underlying graph to be resident in memory, especially in the initial stages. This can pose a challenge for large genomes, where memory constraints become a major computational bottleneck. To address this, certain assemblers have adopted a distributed memory approach, employing numerous servers with large memories to manage the substantial data.

Consequently, a key focus in de Bruijn graph-related research has been the development of smaller and optimal data structures for storing these graphs to facilitate genome assembly [18]. Some researchers have proposed the use of navigational data structures to enhance memory efficiency. In [18], a comprehensive data structure named DBGFM has been designed, implemented, and applied to process the entire human genome dataset, representing a significant step forward in addressing the challenges associated with de Bruijn graph memory management.

Current next-generation sequencing methods generate reads that exhibit significant variability in length, although many technologies typically produce reads around 100 nucleotides [19, 20]. In the quest to assemble these reads into longer continuous sequences, one of the more straightforward methods mentioned in the literature involves utilizing these sequences for assembling the human genome. Implementing such a method, however, is a challenging task due to the inherent difficulty in generating the de Bruijn graph for a single run of the Illumina sequencer, which generates a multitude of reads. Consider that, for example, aligning a million reads requires pairwise alignment, and for a billion reads, staggering one quintillion (10^{18}) alignments are necessary. Regrettably, no optimal and efficient algorithm has been reported for finding Hamiltonian cycles in graphs with billions of nodes. Given the high computational complexity associated with finding Hamiltonian cycles that visit all nodes of a graph exactly once, it appears more tractable to find a cycle that traverses all edges of the graph exactly once—an Eulerian circuit. In this context, rather than assigning each substring within a read as a node, it is more practical to assign these substrings to the edges of the graph. This can be achieved through the construction of de Bruijn graphs, offering a more feasible and efficient approach to address the complexities associated with assembling genomes from vast amounts of sequencing data.

Practical strategies for applying de Bruijn graphs to real datasets are outlined in [17, 18]. Additionally, de Bruijn graphs find diverse applications in bioinformatics, addressing various challenges such as antibody sequencing [21], synteny block reconstruction [22], and RNA assembly [23]. In these applications, de Bruijn graphs serve as a representation of experimental data, facilitating tractable computational problem-solving. Beyond bioinformatics, de Bruijn graphs hold significance in combinatorics on words (finite sequences). The motivation for studying these graphs lies in their connections with concepts in algebra, graph theory, computer science, combinatorics on words, scheduling, and related fields. In [24], several studies delve into word-representable graphs, generalizing important classes such as 3-colorable graphs, circular graphs, and comparable graphs.

Notably, [24] demonstrates that simplified binary de Bruijn graphs are word-representable for strings of length greater than or equal to 1. However, the proof of whether simplified de Bruijn graphs for non-binary alphabets can be of word-representable type remains outstanding. In [24], conjectures are presented, suggesting that simplified de Bruijn graphs may not be word-representable for alphabets of size 3 or more and strings of length greater than 4. These discussions contribute to the ongoing exploration and understanding of the versatility and limitations of de Bruijn graphs in various contexts. Working with growing datasets can be a time-consuming and resource-intensive task. To optimize the processing of corresponding items in the dataset, researchers often find using the sequence (superstring) derived from the de Bruijn graph to be an intriguing option. Such sequences extracted from the de Bruijn graph offer the ability to encompass all possible combinations of data exactly once, facilitating efficient data processing.

The conventional de Bruijn sequence is 1-D that can be represented by a toroidal array or ring interconnection network. In [25], efforts have been directed towards extending this sequence to higher dimensions. In 2-D mode, the de Bruijn graph becomes a torus network represented by a toroidal matrix. In 3-D mode, it is termed a hypertorus, represented by a hypertoroidal matrix. Roig et al. [25] delves into the main features of de Bruijn shapes in one, two, and three dimensions—patterns consisting of several alphabets with specific sizes that occur exactly once in the related shape. The nomenclature and representation of these shapes change based on the dimension used. These patterns find applications in various domains, including location detection, a fascinating problem in the multidimensional de Bruijn graph's applications in image processing, robotics, and machine intelligence [25]. The pattern associated with each item defines its location, and changes in the spotted pattern help identify the item's displacement and direction.

3. Definitions and Preliminaries

Graph models are a well-known method for representing interactions in any network. A graph G is defined by a set of nodes V and a set of edges E, which is a subset of the Cartesian product of nodes. An undirected graph is typically denoted as $G(V, E)$. However, in specific cases like de Bruijn graphs, the concept of directed graphs is essential. In a directed graph, it is represented as $G(V, A)$, where A refers to the set of arcs. Since the Cartesian product is ordered, if v_l and v_j form the two vertices of an edge, then the edge (v_l, v_j) will be different from the edge (v_j, v_l). Otherwise, the graph G is considered undirected.

If A is a set, G is termed a simple directed graph. If A is a multiset, then G is referred to as a directed multigraph [1, 7]. The distinction between these types of graphs reflects the nature of the relationships and interactions within the network, and understanding this distinction is crucial for accurately modeling and analyzing various systems, including de Bruijn graphs.

3.1. The de Bruijn Graphs

The de Bruijn graph, independently proposed by de Bruijn [3, 26] and Goode [3, 27] in 1946, was initially presented by Sclumberger [28] as a topological architecture for interconnection networks. The de Bruijn architecture offers distinct advantages over other interconnection network architectures, such as the hypercube. In this section and the subsequent subsections, we aim to describe the structural and topological properties of de Bruijn graphs along with their related applications.

Various topologies have been proposed in interconnection networks with the goal of minimizing the network diameter for a given number of nodes and a specified degree. The de Bruijn graph stands out as one of the most well-known topologies. This graph is regular, Eulerian, and Hamiltonian, with a small diameter close to optimal connectivity. Its recursive structure is simple, making it easy to design and implement routing algorithms. De Bruijn graphs with the same diameter and degree can support a larger number of processors compared to other interconnection network topologies. For instance, an 8-D hypercube can connect only 256 processors, whereas a de Bruijn network with a diameter and degree of eight can interconnect 65,536 processing elements.

While the degree and diameter invariants in de Bruijn graphs are not directly related, it is evident that networks with a large degree are not practically suitable for implementation in VLSI layout. With a diameter D and degree p, an architectural designer can design the number of vertices between p^{D-1} to p^D with a de Bruijn graph. The unique short paths between each ordered pair of vertices in the de Bruijn graph simplify the implementation of routing algorithms, providing an advantage that other interconnection topologies might lack.

Although we mentioned that the de Bruijn graph is Eulerian and Hamiltonian, it is worth noting that, for example, the hypercube interconnection network exhibits these properties only when its degree is even. The development of the de Bruijn graph does not increase the degree in the network, and any expansion of it does not necessarily require changes or improvements in the hardware equipment of the network. However, it is essential to acknowledge that the de Bruijn graph is not symmetric, and from this perspective, other interconnected networks such as torus and hypercube may perform better than the de Bruijn graph.

3.1.1. Three Equivalent Definitions for de Bruijn graphs

For integers $p \geq 2$ and $k \geq 1$, de Bruijn graph is represented by the symbol dBG(p,k) and can typically be defined in three forms. These three definitions are equivalent to each other.

Definition 1: *Suppose the alphabet set is defined as $\Sigma_p = \{0,1,\ldots,p-1\}$, $p \geq 2$ where p is the length of the alphabet set; i.e., the number of available distinct symbols. Also, let $\sum_p^k$ be a collection of all strings of length $k \geq 1$ built on the alphabet Σ_p. These strings are called p-ary k-length sequences.*

The vertex set of dBG(p,k) is defined as

$$V = \{x_1 \ldots x_k : x_i \in \{0,1,\ldots, p-1\}, i = 1,2,..,k\} \quad (1)$$

and the edge set A includes all edges from one vertex $x_1 \ldots x_k$ to p other vertices $x_2 \ldots x_k \alpha$, such that $\alpha \in \{0,1,\ldots, p-1\}$.

The first definition of de Bruijn graph is, in fact, an extension of the original definition of the binary de Bruijn graph, i.e., p=2, which was independently presented by de Bruijn [26] and Good [27]. Goode raised the question of the existence of a particular shift-register sequence where a binary cycle of length 2^k contains one distinct binary sequence of length k. In this context, dBG(p,k) is a de Bruijn graph of order k from the alphabet set in radix p and is considered a directed multigraph.

Let S be an infinite subset of $\sum_p^k$, then the universal cycle for S can be written as a sequence of length $|S|$ defined so that it contains any string of length k+1 encoded with the alphabet Σ_p. In the literature, these substrings are called (k+1)-mers. When examining such a sequence in a rotational manner, it should be noted that these sequences occur exactly once. It means that the set S includes all (k+1)-mers, and its cardinality will be $|S|=p^k$. In this case, the universal cycle for S is the de Bruijn graph dBG(p,k). The main feature of such sequences is that all possible p-ary k-length combinations occur in them exactly once.

Definition 2: *Using iterated line digraphs, the directed graph dBG(p,k) can be defined by the (k−1)-th iteration of the line graph* K_p^+ *; so that the symbol* K_p^+ *represents the directed graph resulting from the complete graph* K_p *by adding a self-loop to each of its vertices.*

Therefore, the directed de Bruijn graph, dBG(p,k), can be defined in the following recursive form

$$\mathrm{dBG}(p,1) = K_p^+; \mathrm{dBG}(p,k) = L^{k-1}(K_p^+), k \geq 2 \tag{2}$$

This definition was provided by Fiol et al. [29]. Using this definition, the following basic properties can be derived from the dBG(p,k) network

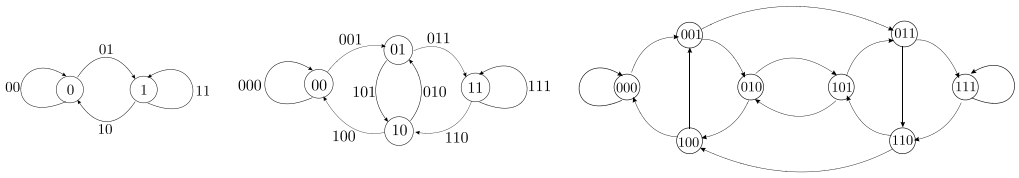
a) dBG(p,k) is a p-regular graph with p^k vertices because each k-tuple (k-mer) represents a node in the network
b) The de Bruijn graph has p^{k+1} edges because there are p^k nodes in the graph, and the output degree of each node is p; thus, each (k+1)-tuple or (k+1)-mer represents an edge in the network
c) The de Bruijn graph has p loop-back links
d) The node connectivity (see Definition 7) of the de Bruijn graph is equal to p-1
e) The diameter of the de Bruijn graph is k because the maximum distance between vertices 00...00 and 11...11 is equal to k, which is obtained from XOR between all 0 and 1 addresses
f) The output degree of each node in the de Bruijn graph is p since the successor of each node is of the form $x_2...x_k\alpha$, and there are p options for α
g) The input degree of each node in the graph is p because the predecessor of each node is of the form $\beta x_1 x_2...x_k$, and there are p options for β
h) Since the edges of the graph are explicitly defined, for a certain p and k, dBG(p,k) has a unique representation

The procedure used in Definition 2 is sometimes called doubling (expanding) the de Bruijn graph. With each doubling, the number of nodes is multiplied by p, and the number of edges is also multiplied by the output degree of the node (i.e., p). In addition, the number of pairs of adjacent edges in the original de Bruijn graph is equal to the product of the number of nodes in the input and output degrees of the node, i.e.,

$p^k.p.p=p^{k+2}$, which is equivalent to the number of edges in the doubled de Bruijn graph.

Definition 3 (Arithmetic method): *This definition was proposed by Imase and Itoh [30]. The set of vertices V and the set of arcs A in dBG(p,k) graph, respectively, are defined as*

$$\begin{cases} V = \{0,1,..., p^k - 1\} \\ A = \{(x,y) : y \overset{p^k}{\equiv} xp + \alpha, \quad \alpha = 0,1,..., p-1\} \end{cases} \tag{3}$$



(a) (b) (c)

Figure 1: Some examples of de Bruijn graphs; (a) dBG(2,1), (b) dBG(2,2) and (c) dBG(2,3); note that dBG(2,2) is the line graph of dBG(2,1) and dBG(2,3) is also the line graph of dBG(2,2).

Figure 1 displays three examples of de Bruijn graphs constructed using Definitions 1 and 2. As illustrated in the figure, line graphs are recursively applied to self-looped complete graphs. Mathematically, we can express this as

$$\mathrm{dBG}(2,1) = K_2^+, \mathrm{dBG}(2,2) = L(\mathrm{dBG}(2,1)) = L(K_2^+), B(2,3) = L(\mathrm{dBG}(2,2)) = L^2(K_2^+)$$

All three Definitions, 1, 2, and 3, are equivalent [3]. In other words, the directed de Bruijn graphs provided by these three definitions are isomorphic with each other. This highlights the isomorphism and equivalence among the various definitions, demonstrating that the choice of definition does not alter the fundamental structure and properties of the de Bruijn graph.

3.2. Eulerian and Hamiltonian Properties of de Bruijn graphs

The de Bruijn graphs are connected, and relating any two words in them creates a path of length k+1. Due to the equality of input and output degrees and its connectedness, dBG(p,k) is strongly connected, encompassing Eulerian circuits and Hamiltonian cycles. For this reason, in recent years, these types of graphs have found extensive applications in information theory and coding, bioinformatics, machine intelligence, and robotics [18, 25].

Definition 4 [7]: *A directed multigraph is termed Eulerian if it possesses a circuit in which every edge is utilized exactly once.*

Euler's theorem [7] posits that a weakly connected directed multigraph is Eulerian if and only if every node is balanced. A balanced graph is one in which the input and output degrees of all nodes are equal. This underscores the widespread utilization of de Bruijn graphs across various domains, leveraging their inherent features such as strong connectivity and the presence of Eulerian circuits and Hamiltonian cycles.

In its simplest form, if the alphabet is binary, the dBG(2,k) graphs are constructed based on binary codes. The construction principle relies on the similarity between two binary codes. Two binary codes are considered similar if the last k letters of the first code are identical to the first k letters of the second code. In other words, if we arrange k+1 letters (symbols) in a row and focus on the first k, we obtain the first code; similarly, by looking at the last k symbols, we get the second code. The significance of this coding technique lies in the fact that by appending a bit (a digit) to the end of each word, one can transition from one word to another in the graph. This becomes particularly valuable when examining a specific feature for all binary bits in the words.

In a conventional approach, if the codeword consists of k letters, there will be binary codewords, requiring the checking of all 2^k words of binary code, each consisting of k letters. This would entail a memory usage of $k.2^k$. However, with de Bruijn coding, one can examine k letters (symbols), add one bit (digit) to create a new codeword. This method only utilizes one additional bit to generate each new k-bit word. If bits (digits) can be systematically added one by one, starting from one codeword and reaching all the codes, then instead of using memory $k.2^k$, only 2^k memory will be necessary. In this context, constructing a directed graph where each k-letter codeword is connected to another k-letter codeword with a directed arc—based on the criterion that the last k symbols of the first code overlap with the first k symbols of the second code—simplifies the coding problem to finding the number of Eulerian tours in the graph.

An important distinction should be highlighted here, particularly in the context of discussing bioinformatics applications of de Bruijn graphs and genome assembly. In this domain, the input parameter k typically denotes the length of the edge label, not the length of the node label [9, 18, 24, 25], unlike in graph theory and applications of de Bruijn graphs in communication networks where the parameter k refers to the length of the labels of nodes, not edges [3, 5, 6]. To clarify further, in bioinformatics and genome assembly, scholars often use the version of de Bruijn's line graph, i.e., dBG(p,k+1), where edges are treated as equivalent to nodes, and nodes are equivalent to edges. This distinction may cause confusion for researchers who concurrently explore both fields.

In essence, due to the characteristics of both Hamiltonian and Eulerian graphs, the construction and representation of such graphs can be based on either the Eulerian circuit or the Hamiltonian cycle. Given the parameter k, if constructing the de Bruijn graph based on the Hamiltonian cycle is of interest—where all the vertices of the graph are supposed to be visited only once—then the label of each vertex will be k. Even though edges are not critical in the Hamiltonian cycle traversal, as each edge connects two adjacent vertices of the graph, assuming the vertices of the graph are k, it is necessary for the labels of the edges to be k+1 to avoid overlap with each other, resulting in a k-tuple substring. Similarly, if traversal based on the Eulerian circuit is desired, where edges are traversed only once, it is necessary to add a unit to the given parameter k to create k-length labels for the edges, despite not dealing directly with nodes. Consequently, for their traversal, k-tuple substrings will be generated.

The current study adopts the standard notation of graph theory and interconnection networks. Specifically, the notation dBG(p, k) represents a 1-D de Bruijn graph wherein all node labels are k-ary, and all edge labels are (k+1)-ary, also known as k-mer and (k+1)-mer, respectively. In the context of bioinformatics and genome assembly, it is important to note that the parameter k is effectively one unit higher. Therefore, k needs to be adjusted by reducing it by one unit. To elaborate, in bioinformatics and genome assembly, k-mer refers to the labeling of edges and (k–1)-mer refers to the labeling of nodes with the specified value k.

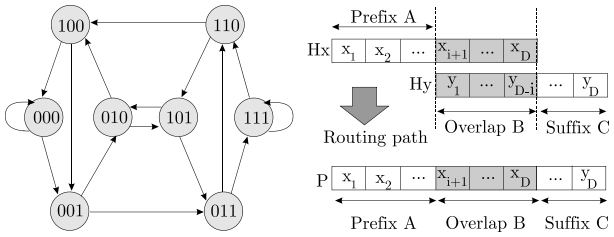


Figure 2: The left panel shows a dBG(2,3) example with p=2, k=3, D=3, and 8 nodes. In the right panel, the best path from H_x to H_y is displayed. The suffix H_x overlaps the prefix H_y, and the routing path is generated by combining the prefix, the overlap, and the suffix C [5]. It should be noted that any string of length k+1 contains a length k prefix and a length k suffix; in reality, the corresponding string is of length k+1 and is generated by overlapping two k-ary strings.

The de Bruijn graph depicted in Figure 2 is denoted by the symbol dBG(2,3) in the context of graph theory and interconnection networks. However, in bioinformatics and genome assembly, this graph is equivalently referred to as dBG(2,4). Throughout this study, we adhere to the standard notation of graph theory and its associated networks. In the de Bruijn graph, the adjacency of two words is reflected in each edge, representing a k+1 letter word, commonly referred to as a (k+1)-mer. To illustrate, the last letter of the second word is appended to the end of the first word, forming a k+1 letter codeword, as depicted in Figure 2. Thus, there exists a precise one-to-one correspondence between words with k+1 letters and edges within the de Bruijn graph. Each edge corresponds to a word with k+1 letters, and reciprocally, each word with k+1 letters corresponds to an edge, where the last k letters of the first word are connected to the first k letters of the second word.

An intriguing observation lies in considering a path within the de Bruijn graph and traversing its successive directed edges. Given that each codeword is composed of k+1 letters, traversing each directed edge reveals a word code with an exact overlap of k letters with the codeword, except for the last letter, which is changed. Notably, each node in the network possesses two inputs and two outputs, leading to k codewords represented by $2^{k+1}/2$ or 2^k vertices. The k-mers represent the codewords assigned to the vertices of the de Bruijn network. Consequently, all generated binary codes exhibit this property. Typically, 2^k memory words are required in such cases. Constructing the graph in this manner facilitates the generation of an Eulerian tour on all edges. As an Eulerian tour touches each edge only once, a binary word of k+1 letters is traversed once, necessitating exactly 2^{k+1} edges or memory words. In this graph, there is no requirement for k bits or new digits to generate the second word; only one bit (digit) suffices. This process mirrors the exploration of an Eulerian tour that visits all binary words in 2^{k+1} memory words.

The binary de Bruijn graph, denoted as dBG(2,k), is a 2-regular directed graph. It is evident that each vertex has both input and output degrees equal to 2. Consequently, the number of arcs can be calculated using the equation below.

$$\sum_{i=1}^{2^k} \deg v_i = 2 \cdot 2^k \Rightarrow |A| = 2^{k+1} \quad (4)$$

Thus, the de Bruijn graph has 2^{k+1} words (edges), where each edge corresponds to a k+1 symbol word (string). $U=\Sigma^k$ represents the universal set of all these (k+1)-mers, and the binary relationship $x \rightarrow y$ between two strings implies the existence of a precise suffix-prefix overlap of length (k+1) between y and x. If we represent this set as S, the de Bruijn graph S is a directed graph whose nodes are precisely k-mers of S, and edges are (k+1)-mers expressed by the relation "$\rightarrow$". The graphs reviewed and represented so far (Figures 1 to 3) are based on the binary alphabet. Instead of binary codes, one can use codes based on p (an alphabet with p letters). In this case, the input and output degrees of each vertex in the de Bruijn graph will be equal to p. These codes are called p-radix de Bruijn codes. The corresponding graph is defined as $\mathrm{dBG}(Z_p^{k-1}, A_k)$ where Z_p is the set of integers on radix p, Z_p={0,1,...,p-1}, and is also all sequences of length k-1, and A_k is the arc set of the graph, where each arc is equivalent to a codeword (k-ary string). The vertices of the graph are also in the form (b_1,...,b_{k-1}) and (c_1,...,c_{k-1}), so that b_i, $c_i \in Z_p$ if and only if $b_2=c_1$, $b_3=c_2$,...,$b_{k-1}= c_{k-2}$. The resulting graph is Eulerian and strongly connected. Moreover, the input and output degree of each node in it is equal to p.

3.3. The de Bruijn Sequences: 1-D

In the de Bruijn graph, dBG(p,k), circularly overlapping all k-ary strings results in a superstring of length pk. These superstrings are known as de Bruijn sequences, and they consist of all k-ary strings that appear precisely once in the superstring constructed of p-ary alphabets. De Bruijn sequences can be constructed algorithmically as well as visually. The goal of this part is to delve further into these sequences. The de Bruijn sequences corresponding to 1-D are linearly growing sequences. Each item can be in several places, making it suitable for angular encoding techniques [25]. De Bruijn networks perform similarly to ring interconnection networks in 1-D mode. The de Bruijn sequence, for example, with alphabet p=2 and string length k=2, is analogous to a 1-D toroidal array or a 4-node ring interconnection network. In this scenario, there are p^k bi-string patterns, presented as a wraparound linear array in a circular form, consisting of p-ary k-length strings precisely once. For instance, such strings can be 00, 01, 11, and 10. These strings

can be obtained by starting at the top and proceeding clockwise, or by traveling from left to right down the array and traversing on the wraparound link. In this situation, the number of distinct *p*-ary *k*-length occurrences is given by [31].

$$(p!)^{p^{k-1}} / p^k \tag{5}$$

For example, there is only one instance for 2-ary 2-length.

De Bruijn's study in the early half of the twentieth century was rooted in challenges related to combinations, particularly integer strings in which all substrings of a specific length appear only once. In 1946, he became interested in the superstring issue, which involves determining the smallest circular superstring that encompasses all feasible substrings of length *k* on a given alphabet. De Bruijn was the first to prove a hypothesis concerning the number of binary sequences. For this reason, and as a tribute to his discovery, such sequences were denoted by the dBG(p,*k*) notation. These substrings are known as *k*-mers. There are p^k *k*-mers in the alphabet with p symbols. There are 4^3=64 trinucleotides in gene expression where the alphabet contains four symbols *A*, *T*, *G*, and *C*. When a binary alphabet is assumed, 2-ary 3-mers are obtained. In other words, there are a total of 8 potential 3-mers, which are made up of eight 3-digit strings. The superstring 0001110100, for example, not only contains all 3-mers but is also the shortest, with each 3-mer appearing precisely once.

De Bruijn demonstrated that his graphs may include a predetermined number of Hamiltonian cycles or Eulerian circuits. There are precisely $2^{2^{k-1}-k}$ unique Hamiltonian cycles in binary de Bruijn networks with 2^k nodes. In other words, the number of 2-ary k-length sequences is equal to $2^{2^{k-1}-k}$ [6]. The noteworthy thing about this number is that it corresponds to the number of Hamiltonian paths in a binary de Bruijn network.

A *T*-net of order m is a network with m nodes and 2^m directed edges. A *T*-net has the feature that the degree of input and output of each node is equal to 2. As a result, the binary de Bruijn, dBG(2,*k*), is likewise a *T*-net of order 2^k with 2^k nodes and 2^{k+1} edges. The number of Eulerian circuits of a disconnected *T*-net is obviously 0. If |*N*| signifies the number of Eulerian circuits of a *T*-net with order m and this *T*-net is doubled using Definition 2, then the number of Eulerian circuits of a double *T*-net with order 2^m is equal to $2^{m-1}|N|$.

By using Definition 2, $\mathrm{dBG}(p,k) = L^{k-1}(K_k^+)$, and assuming that the set of vertices K_p includes {0,1,...,*p*-1}, then, due to the definition of the line graph, each vertex of dBG(*p*,*k*) will be an edge of dBG(*p*,*k*-1), which can be expressed with a *p*-ary *k*-digit string as $x_1...x_k$ so that $x_i \in \{0,1,..., p-1\}, i = 1,2,..., k$. Let $C = \{e_1, e_2, ..., e_\varepsilon\}$ be the Eulerian circuit of the graph dBG(*p*,*k*-1), so that we get

$$\begin{aligned} &e_i \in A(\mathrm{dBG}(p,k-1)), i = 1,2,..., \varepsilon \\ &\varepsilon = \varepsilon(\mathrm{dBG}(p,k-1)) = p^k \end{aligned} \tag{6}$$

In this case, two consecutive edges e_i and e_{i+1} from the set *C* must satisfy the property that if $e_i = x_1...x_k$ then $e_{i+1} = x_2...x_k\alpha$ in which $\alpha \in \{0,1,..., p-1\}$. Therefore, the *p*-ary p^k-digit sequence can be obtained as

$$M(p,k+1) = (x_{i1}x_{i2}...x_{i\varepsilon}) \tag{7}$$

Hence, the de Bruijn sequence, *M*, is a Hamiltonian cycle or an Eulerian circuit. Since there are p^k nodes in the de Bruijn graph, this sequence is necessarily as long as p^k symbols. Similarly, a *k*-ary sequence with p^k digits (*p*-ary p^k-digit) corresponds to a de Bruijn graph that can be constructed from the Eulerian circuit or Hamiltonian cycle of this graph. In fact, here the first digits of the labels of each edge have been employed. By successively removing the first digit of the label of each edge in the set *C*, the de Bruijn sequence *M* can be obtained. For example, the Eulerian circuit of the directed dBG(2,3) shown in Figure 3 is of the form $C = \{e_1, e_2, ..., e_{16}\}$. Therefore, the 16-digit binary de Bruijn sequence corresponding to this graph is given by

$$M(p,k+1)=M(2,4)=0000111100101101$$

Figure 3 shows the de Bruijn graph with *k*=3 and a binary alphabet. By traversing the Eulerian circuit in this graph, if one records the first digit of each edge label, the rotated superstring *M*(2,4) described above will be obtained.

As a result, a superstring (a de Bruijn sequence) is defined as a rotating string of p^k digits (symbols) in which each *k*-tuple (a string of length *k* containing *p* symbols) appears precisely once. According to this, the de Bruijn sequences are of the form $\{\alpha_i\}_{i=1}^{p^k}$ and $\{\beta_i\}_{i=1}^{p^k}$. According to the definition, these two sequences are equal if and only if $\exists c$ such that $\forall i \in \{1,..., p^k\}$, $\alpha_i \overset{p^k}{\equiv} \beta_{i+c}$.

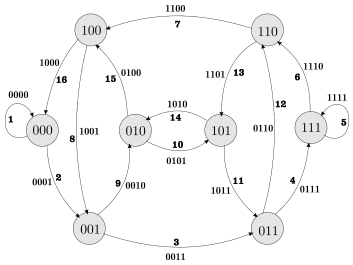


Figure 3: The de Bruijn graph, dBG(2, 3), and the associated Eulerian circuit generated by traversing the edges from 1 to 16; nodes are 3-mer and edges are 4-mer. The input and output degrees of the nodes are both equal to 2, and the graph contains both an Eulerian circuit and a Hamiltonian cycle. The circular superstring $M(2,4)$=0000110010111101 is generated by adding the first digit of the label of each traversed edge.

As previously stated, the sequence $M(p,k+1)$ is known as the de Bruijn sequence (superstring). When seen as a ring sequence, each $k+1$ successive digit represents an edge of the dBG(p,k) as well as a vertex of the dBG($p,k+1$) (line graph). The superstring of a de Bruijn network has numerous features similar to Gray coding in k-ary n-cube topologies (e.g., hypercube).

A theorem [3] states that if C is a directed circuit of length $m<k$ in the dBG(p,k) and also $x = x_1...x_k$ is a vertex in C, then $\forall i=1,2,...,k-m,\ x_i = x_{i+m}$. This theorem refers to the observation that in the de Bruijn graph, any circuit like C of length $m<k$ can be expressed using a ring sequence including m consecutive coordinates at each vertex of the set C. Similarly, any p-ary m-digit numerical ring sequence will represent an orbit of length m in the dBG(p,k), implying the fact that for any integer such as $m, 1\leq m\leq p^k$, the dBG(p,k) contains a directed circuit of length m [3]. It should be noted that in the binary version of de Bruijn graph, i.e., dBG(2,k), the number of de Bruijn sequences for $k\geq 1$ is equal to $2^{2^{k-1}-k}$, which is actually the same number of Hamiltonian cycles of the graph [3, 6].

In summary, de Bruijn's main problem is to construct such a superstring of all k-mers with an arbitrary value of k and an optional symbolic alphabet p. That is, the dBG(p,k) should be constructed in such a way that each k-mer is assigned to a node and connected to the next k-mer through a directed edge. However, this connection is established on the condition that there is a ($k+1$)-mer whose prefix belongs to the first node and whose suffix belongs to the second node. Edges in the de Bruijn graph represent all possible ($k+1$)-mers. In this case, the existence of an Eulerian circuit in such a graph represents the shortest circular superstring in which each ($k+1$)-mer has appeared exactly once (see Figure 3). It is noteworthy to mention that in genome assembly, the constructed de Bruijn graph does not use all possible ($k+1$)-mers as edges; instead, only those generated from reads are used.

From the algorithmic perspective, three different methods have been proposed for deriving the de Bruijn sequences analytically. For example, one can refer to the greedy methods and the construction of the shift register sequence with feedback (FKM) [32]. Also, an efficient solution is proposed by Wong [33], which is able to generate any sequence with appropriate time complexity. Rotational sequences (superstrings) have special applications in de Bruijn graphs. For example, for gene expression in the field of bioinformatics, test objectivesare used in the field of image processing, in coding and information theory, as well as the generation of pseudo-random values in the field of security and cryptography. However, the most recent use of these rotating sequences is related to shotgun DNA sequencing, which is used for the reassembly of large DNA strands from smaller sub-strands widely exploited in sequence analysis techniques in bioinformatics algorithms.

It is worth noting at the conclusion of this section that the notion of 1-D de Bruijn sequences may be extended to higher dimensions. It is also feasible to expand its uses to higher dimensions, such as coding, communication, pseudo-random arrays, or spectral imaging. Furthermore, location detection is one of the intriguing applications in which each pattern happens exactly once. When two dimensions are involved, 2-D patterns can be identified using a wraparound 2-D array. The 3-D patterns can also be employed in three dimensions.

A hypertorus is a 3-D de Bruijn graph. In this situation, the size of all dimensions of the de Bruijn hypertorus is the same, which is also known as the cubic de Bruijn's 3-D hypertorus. A study of the theoretical and practical concerns of various kinds of de Bruijn graphs is presented in [25]. The essential feature of these is that the current patterns are made up of p-ary alphabets of a specific size that appear precisely once in the relevant form. Furthermore, because de Bruijn's 3D hypertori are 3-D, they include patterns that are connected to one another. Hypertori and their patterns can be cubic, rectangular, or a mix of the two.

4. Generalized de Bruijn Graphs

In the previous sections, it was mentioned that de Bruijn graphs have many desirable properties. However, there is a hidden inherent flaw in them which is the limitation in the number of graph vertices. Due to the existence of a large gap between p^k and p^{k+1}, the network designer is faced with two extremes; that is, choosing too few vertices or choosing too many vertices. One of the ways to solve this problem is to develop de Bruijn graphs for a general number of vertices. If one substitutes the value of k instead of p^k in

Definition 3 of the de Bruijn graph, it can be developed for more general situations. Such a development is called a generalized directed de Bruijn graph or GdBG(p,k) for short. Specifically, for $k \geq p \geq 2$, GdBG(p,k) contains the following vertex set and edge set

$$\begin{cases} V = \{0,1,...,k-1\} \\ A = \{(x,y) : y \overset{k}{\equiv} xp + r, \quad r \in \{0,1,...,p-1\} \end{cases} \tag{8}$$

Figure 4 depicts a generalized de Bruijn graph, GdBG(2,7).

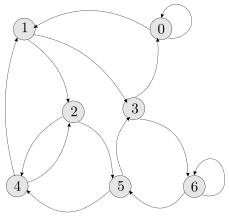


Figure 4: A generalized de Bruijn graph, GdBG(2,7), with k=7 vertices labeled 0 to 6.

In [10,12], random regular graphs (RRG), which are special cases of directed regular graphs (DRG), have been analyzed. Moreover, they have been generalized, evaluated, and compared with de Bruijn networks under different workload conditions. The empirical results show that RRG networks with k-shortest path routing are not ideal in terms of diameter and load balance. However, generalized de Bruijn graphs are close to the optimal networks in most network configurations, based on criteria such as diameter, average distance, and load balance.

During the past years, the problem of graph design with optimal diameter and fixed degree has been investigated. That is, it is assumed that there is an optimal graph with a constant degree p and diameter D. Now the important question is what is the maximum number of nodes, k, which can be packed in such a graph? In other words, what is the maximum number k of vertices that can be in a graph (directed graph)? Where the maximum degree (output degree in digraph mode) of each vertex is p and the diameter of the graph is D. One of the best answers is provided by Moore, which refers to Moore's bound. This problem is also known as (k,p,D).

Theorem 1 (Moore Bound) [3, 5, 7]: *Let G be a graph with k vertices, diameter D, and maximum degree p, then we get*

$$k \leq 1 + p + p(p-1) + ... + p(p-1)^{D-1} = \frac{1 + p[(p-1)^D - 1]}{p-2} \tag{9}$$

If G is a digraph with diameter D, the preceding theorem becomes $k \leq (p^{D+1} - 1)/(p-1)$.

Moore's bound can be obtained only for experimental values of p and D, and it cannot be obtained for any non-trivial graph. The generalized de Bruijn directed networks, of course, are extremely near to the Moore limit and may be built with $k=p^D$ or $k=p^{D(1+1/p)}$ nodes. However, as previously stated, the amount of this proximity is not entirely evident. Imase and Itoh [30] proposed generalized de Bruijn graphs that are close to the optimal diameter $\lceil \log_p^k \rceil$. GdBG(p,k) is a graph with a diameter $\log_p^k$, where p represents the fixed degree of each node and k represents the total number of nodes. As a result, it is near to the optimal graph, and because it is directed, each node has p input edges and p output edges. This is a characteristic shared by many distributed hash tables (DHTs) [5].

De Bruijn graphs are particularly advantageous in Peer-to-Peer (P2P) network architectures due to their small diameter, which sets an upper bound on the number of hops between each pair of network users (nodes). However, the diameter metric only provides an upper limit on the largest distance between two nodes. Therefore, it is often more meaningful to consider equitable measurements, such as the average distance between each pair of nodes in the network. The average distance reflects the efficiency users can expect when searching for content in the network. To calculate the average distance, one must first determine the probability mass function (pmf) for $d(x,y)$ and then compute the mathematical expectation. The asymptotic distribution of the shortest distances in generalized de Bruijn graphs can be approximated using the formula provided by [5].

$$f(i) \approx \frac{p^i}{k} - \frac{p^{2i-1}}{k^2} \geq \frac{p^i(1-1/p)}{k} \tag{10}$$

where $f(i)$ is the pmf function of distance distribution. It is worth noting that the aforementioned relationship is quite near to $f(i)$ for graphs with diameters less than 3 ($D \leq 3$). This inequality denotes how many neighbors are located in the shortest distance i from a particular node v in a graph of diameter D and degree p. The mathematical expectation of the distances pmf distribution in the graph may be used to determine the average distance measure. As a result, in the extended de Bruijn graph, the asymptotically average distance may be represented as [5]

$$\bar{d} = \sum_{i=0}^{D} i.f(i) \approx D - 1/(p-1) \tag{11}$$

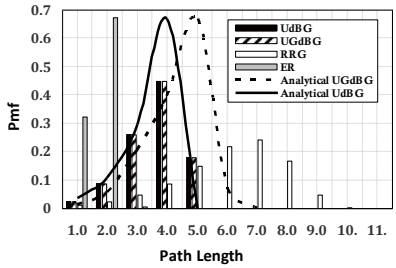


Figure 5: Probability mass function (pmf) of the shortest paths in various de Bruijn graphs, ER and RRG random networks; the horizontal axis represents the length of the shortest path; the number of nodes is assumed to be 243, and the network average degree is assumed to be 3. The numerical values represented by the analytical model of Equation (10) for UdBG and UGdBG are demonstrated as the solid and dashed curves, respectively.

Figure 5 depicts the results of the simulation of the frequency distribution of the shortest paths in various de Bruijn graphs, ER random network, and RRG random regular network. The number of nodes is assumed to be 243 and the average degree of the network is assumed to be 3. Also, to validate the analytical model of Equation (10), the numerical values obtained from this model are drawn next to the simulation experiments. The outcomes indicate the appropriate accuracy of the analytical Equation (10) for approximating the distance values in de Bruijn graphs. The results also show that most of the nodes in de Bruijn graphs are accessible in the shortest distance of diameter D from a given node like v. In other words, the average distance in de Bruijn graphs is very close to the diameter D, and the local structure of the graph at each node is similar to a tree. In other words, the number of short cycles is very small, and of course, as follows, the clustering feature works poorly in these graphs.

In addition to the diameter, the bisection width serves as a crucial metric for evaluating the robustness of graphs and interconnection networks. This index is particularly valuable for comparing graphs, especially in terms of fault tolerance. The bisection width is defined as the minimum number of edges situated between any two partitions of the graph of equal size [3, 5]. Essentially, this measure indicates the challenge of dividing the graph into giant components by removing its edges, and it is closely linked to network congestion and bottlenecks in information transmission. In other words, it establishes an upper limit for the attainable capacity of the graph, offering insights into effective network routing. The bisection width of generalized de Bruijn graphs is constrained by [5].

$$\frac{pk}{2\log_p^k} \le bw_{\mathrm{UGdBG}(p,k)} \le \frac{2pk}{\log_p^k} \tag{12}$$

In short, the important characteristics of GdBG(p,k) for $k \ge p \ge 2$ can be summarized as follows

- The generalized de Bruijn graph is p-regular and has self-loop
- The number of self-loops in GdBG(p,k) is at most $2p$ [10, 12] and when $p \square k$, the effect of these self-loops can be ignored
- This graph has a walk (i,j) of length m for any two arbitrary vertices i, j; if and only if there are m integers like r_l such that [3]

$$j \equiv^{k} ip^m + r_1 p^{m-1} + \ldots + r_{m-1}p + r_m, \;\; r_l = 0,1,\ldots,p-1, l = 1,2,\ldots,m \tag{13}$$

- The GdBG(p,k) is strongly connected
- If D is the diameter of GdBG(p,k), then $D = \left\lceil \log_p^k \right\rceil$ and $p^{D-1} < k \le p^D$
- If $D \le 4$, then the node connectivity of the generalized de Bruijn graph is equal to $p-1$

5. Undirected (Simplified) de Bruijn Graphs

Esfahanian and Hakimi [3, 6] proposed a version of the de Bruijn graph, which is called undirected (simplified) de Bruijn graph denoted by UdBG(p,k). This version is the modification of the classical de Bruijn graph, dBG(p,k) such that all self-loops are removed and if there is a directional edge $\vec{\alpha\beta}$ in dBG(p,k), then it will be converted into a non-directional edge $\overline{\alpha\beta}$ in UdBG(p,k). Finally, if both edges $\vec{\alpha\beta}$ and $\vec{\beta\alpha}$ are available in dBG(p,k), they will be replaced by only a single edge $\overline{\alpha\beta}$. For $p \ge 2$, $k \ge 1$, the UdBG(p,k) has the following properties [3, 6]

- The maximum degree of the graph is $\Delta = 2p$ and its minimum degree is $\delta = 2(p-1)$

- The diameter of the graph is equal to k; referring to the fact that removing the self-loop and multiple edges in the graph has no effect on the diameter
- Node connectivity of the graph is equal to $\lambda=2(p-1)$
- If the number of faulty nodes in the graph is less than or equal to $2p-3$, then the maximum length of the shortest path between any two non-faulty nodes will be less than or equal to $2k$

5.1. Simplified Word-Representable de Bruijn Graphs

Word-representable graphs are a generalization of several significant graph types, including 3-colorable, cyclic, and comparable graphs. Several investigations on these graphs have been described in the literature [24]. One of the most compelling reasons to investigate such graphs is their relationship to issues in algebra, graph theory, computer sciences, word combinations, scheduling, and so on.

Definition 5 [24]: *Assume that G(V,E) is a simple graph. The graph G is termed a word-representable graph if there exists a word, say w, on the alphabet V such that for any $x\neq y$, the letters x and y in w alternate if and only if $xy\in E$. The letters x and y alternate in a word w if after eliminating all letters except copies of x and y in w, one may build the words xyxy or yxyx in an even or odd length,.*

In [24], the word-representability of undirected and simplified de Bruijn graphs is investigated. As mentioned in Section 3, the UdBG(p,k) is a simplified undirected graph, which is obtained from the original dBG(p,k) by removing the directions and self-loops and replacing multiple links between a pair of vertices with a single edge. In [14, 34] it is shown that the minimal graph that is not word-representable is W5, a six-vertex wheel graph, which is obtained by adding a vertex to the cycle graph, C_5, so that it is adjacent to all its vertices. The wheel graph, W_5, is isomorphic to the induced subgraph of the simplified de Bruijn graph, UdBG(3,2).

The de Bruijn graphs are among the key objects in combinatorics on words. Therefore, they have many applications in sciences, especially in genome assembly. In [24], it is shown that the UdBG(2,k) is word-representable for $k\geq1$; while UdBG(p,2), UdBG(p,3) are not word-representable for $p\geq3$. In [24], a conjecture is given that the simplified de Bruijn graphs, UdBG(p,k), are not word-representable for $k\geq4$, $p\geq3$.

Definition 6 [24, 34]: *A directed graph G(V,E) is semi-transitive if it has no directed cycles and for any directed path, $v_0\rightarrow v_1\rightarrow\ldots\rightarrow v_k$, $k\geq4$, $v_i\in V$, either $v_1v_k\notin E$ or for all $0\leq i<j\leq k$, $v_i\rightarrow v_j$ is an arc; i.e. $v_iv_j\in E$. When v_1v_k is an edge, we say it is a shortcut.*

In other words, a semi-transitive graph is a graph that allows a semi-transitive direction and has no any cycle. For example, the Holt graph is the smallest semi-transitive graph has 27 vertices with degree 4 and does not have the property of reflection symmetry; that is, edges are not equivalent to their inverse. This graph is sometimes called Doyle where its chromatic number is 3. Moreover, the Holt graph is Hamiltonian and unit-distance. A theorem states that a graph is word-representable if and only if it is semi-transitive [24, 34, 35]. The corollary of this theorem also implies that every 3-colorable graph is necessarily word-representable. Binary simplified de Bruijn graphs, UdBG(2,k), are semi-transitive and 3-colorable, so they are word-representable. The following theorem refers to this issue. However, the graphs UdBG(p,2) and UdBG(p,3) are not word-representable for $p\geq3$.

Theorem 2 [24, 34]: *The UdBG(2,k) is a word-representable for $k\geq1$.*

It has been conjectured that all graphs with a maximum degree of 4 are not word-representable [24, 34]. it is shown that the number of graphs with n vertices that can be word-representable is approximately $2^{n^2/3+O(n^2)}$ [24]. Word-representable graphs exhibit interesting features, including the maximum clique, which can be calculated in polynomial time. It has been demonstrated that a graph admits a half-transitive orientation if and only if it is word-representable. For example, the Petersen graph [7] is both a word-representable and a 3-regular graph. Notable examples also include the complete graph K_4. According to [24], if each letter occurs exactly k times, the graph is k-word representable. For instance, the Petersen graph is a 3-word representable graph. A graph is 1-representable if and only if it is complete, and it is 2-representable if and only if it is a cycle. Figure 6 illustrates an induced subgraph of UdBG(3,3), which is minimal with the lowest number of vertices not word-representable.

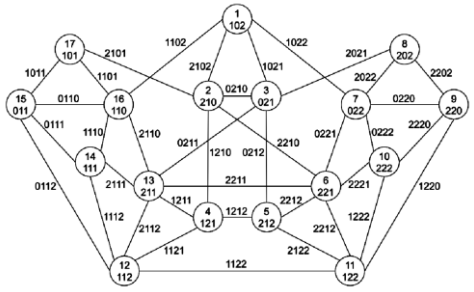


Figure 6: An inducted subgraph of UdBG(3,3) that is not word-representable [24].

6. Fault Resilience and Fragility Analysis

In recent years, the study of resilience in graphs, complex networks, and communication infrastructures has gained popularity, emerging as a rapidly growing academic discipline. This field aims to identify methods, processes, and criteria to enhance network connectivity against random failures and systematic attacks. Consequently, numerous methods have been proposed to evaluate the robustness and resilience of graphs and complex networks [1-3]. The resilience of a network is often determined by its fault tolerance and vulnerability to random failures and targeted attacks. In this section, we will first discuss some of the most essential criteria and then explain how these criteria can be employed to assess the resilience of various types of de Bruijn graphs. Some of these criteria fall under the category of min-cut, which will be further examined below.

6.1. Min-Cuts based Resilience Measures

The criteria based on the min-cut can be viewed as derivatives of the classical connectivity problem. In these metrics, the primary concern is to identify the minimum number of components in the network, the removal of which causes the network to disintegrate. In some cases, especially for large graphs, this problem may fall into the category of NP-complete. Connectivity itself can be divided into two categories: edge connectivity and node connectivity. These measures play a crucial role in evaluating the robustness of graphs. In this article, our objective is to establish a connection between node connectivity and the symmetry property in de Bruijn graphs. We demonstrate that, to achieve a robust network, it is necessary to simultaneously satisfy two conditions: optimal connectivity (OC) and node similarity (NS) in a given graph.

Definition 7 (node connectivity κ) [3]: *The minimum number of nodes whose removal causes the graph G to become disconnected.*

Definition 8 (edge connectivity λ) [3]: *The minimum number of edges whose removal causes the graph G to become disconnected.*

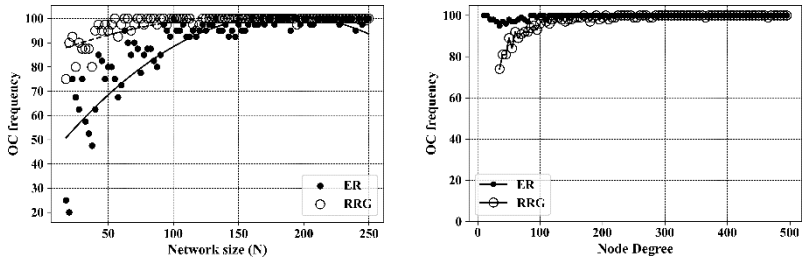
For example, in complete graph, K_n, $\kappa=\lambda=n-1$.

Theorem 3 [36]: *For a graph G, we obtain $\kappa \leq \lambda \leq \delta$, where δ is the minimum degree of the graph G.*

It should be noted that the above inequality does not hold for the complete graph K_n; since it cannot be disconnected by removing the vertices. In order to establish this inequality for K_n, node connectivity is usually defined as $\kappa - 1$.

Definition 9 [36]: *For a given graph G, if $\kappa=\lambda=\delta$, then G is called optimally connected.*

We denote it by OC throughout this article.



(a) (b)

Figure 7: The frequency of ER and RRG model random graphs with OC feature. A total of 1000 graphs of each type have been generated. In panel (a), the average degree of graphs is assumed to be 3 and the frequency of OC is plotted according to the size of the network, while in panel (b) the frequency of the number of graphs with OC feature is plotted according to the degree of nodes in the graph. For more clarity, the regression curve is fitted to the points captured through the simulations.

Investigating the presence of OC feature in graphs is important because in such graphs the number of links, the diversity of nodes and connections is high, so the network can be strengthened against random failures and targeted attacks. According to a theorem [36], in any random graph of size N, when the connection probability tends to 1, in the limiting case $N \to \infty$, the OC condition will be fulfilled. It should be noted that in such a situation, the discrete degree distribution in the graph follows Poisson. According to the central limit theorem (CLT) with the increase of the degree size and the number of samples, it will follow the normal distribution in

the continuous state. To validate this issue, 1000 random graphs of ER and RRG models were generated and then the frequency of the number of graphs with OC feature was counted. In panel (a) of Figure 7, the average degree of the graphs is assumed to be 3 and the frequency of OC is plotted according to the change in the network size. While in panel (b), the frequency of OC for these two models is plotted in terms of changes in the nodes degree. These two figures show how the OC feature in graphs changes according to the size and degree of the node. The graphs obtained in both panels (a) and (b) show that by increasing the size and degree of nodes, the OC condition in random graphs will be fulfilled in the limiting case.

Definition 10 [37]: *A group is a finite/infinite set of elements with binary operations with the four properties of closure, associativity, identity, and inverse member existence. A group whose every member is invertible is called a monoid.*

Definition 11 [37]: *Permutation group is a finite group whose elements are permutations of a given set and the group operation is a combination of permutations in graph G.*

Definition 12 [37]: *Automorphism is a permutation like π of the set of VG vertices that preserves links. That means, if u and v are two adjacent vertices (edges) of graph G, then π_u and π_v will also be adjacent (edges of G). It means that the $\pi \in S_{|V_G|}$ where set S_n is the permutation group on n vertices of G that preserves the proximity/non-adjacency of the vertices. In other words, G is identical to itself. So, we get*

$$\begin{aligned} &\mathrm{Aut}_G \times V_G \to V_G \\ &\mathrm{Aut}_G = \{\exists \pi : V_G \xrightarrow[\text{onto}]{1\text{–}1} V_G \ \ni \ \forall u, v \in V_G, (u,v) \in E_G \Rightarrow (\pi_u, \pi_v) \in E_G\} \end{aligned} \tag{14}$$

It should be noted that automorphism is a form of symmetry. That is, the graph is mapped into itself and at the same time its node-edge connectivity is also preserved. Hence, automorphism can be considered as symmetry in an object can be interpreted as a method of mapping an object into itself so that the entire structure of the desired object remains unchanged. In fact, automorphism is a permutation of the vertices number of a graph. That is, if a graph has vertices labeled 1 to *n*-1 and each permutation of these numbers forms a graph, then these graphs will be isomorphic with each other. It is necessary to mention that the set of automorphisms of a graph form a group under combination of functions. The identity mapping of a graph into itself is always an automorphism, which is sometimes called a trivial automorphism.

Definition 13 [36, 37]: *A graph G is called node similarity (NS) if and only if the following relation holds for any two arbitrary nodes u and v*

$$\forall u, v \in V_G, \exists \pi \in \mathrm{Aut}_G \ \ni \ \pi_u = v \tag{15}$$

Thus, NS property in a graph means, all the nodes look similar to each other. Usually, the NS property is also called vertex-transitive. In fact, a vertex-transitive graph is a graph in which every pair of nodes are equivalent to each other under some elements of the automorphism group. It should be noted that NS does not necessarily mean the existence of symmetry in the graph. This feature can be widely used in parallel processing applications, optimal routing algorithms and traffic load balancing in networks. The property of NS in graphs can also mean that the importance of nodes is seen equally from the attacker's point of view; this leads to more fault tolerance and resilience against random failures and targeted attacks.

Theorem 4 [36]: *For every connected graph with NS characteristic, the following relations hold*

1) $\lambda = \delta$
2) $\kappa \geq 2(\delta+1)/3$
3) if $\delta \leq 4$ then $\kappa = \delta$
4) if G is symmetric then $\kappa = \delta$

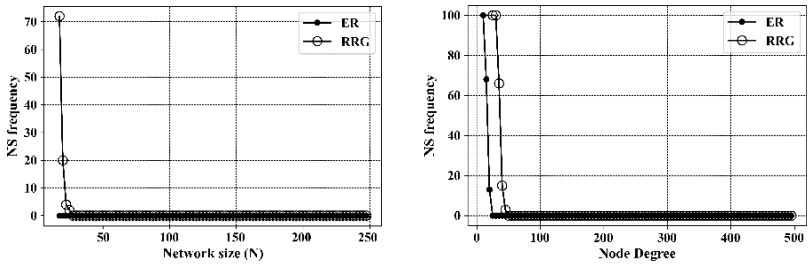
Figure 8 illustrates the frequency count of ER and RRG random graph models with NS characteristics in terms of node degree and network size. The graph depicts a noticeable decrease in the number of NS graphs as the degree and size of the random networks increase. In this scenario, 1000 random graphs from both models were generated, and the frequency of graphs exhibiting NS characteristics was tallied. Panel (a) presents the average degree of the nodes, while panel (b) displays the frequency of NS for the two models in relation to network size. Both panels vividly portray the substantial reduction in the NS characteristic in random graphs as both the average degree and the graph size experience an increase.

Definition 14 [37]: *The graph G is symmetric if and only if the following relation holds for both links (u,v) and (x,y) of the set of edges of G*

$$\forall (u,v), (x,y) \in E_G, \exists \pi \in \mathrm{Aut}_G \ \ni \ \pi_u = x \wedge \pi_v = y \tag{16}$$

Hence, the symmetry property preserves links and its existence can mean that all links look similar to each other. In other words, for both links $e_1=(u,v)$ and $e_2=(x,y)$ of the edge set of the graph G, there is a permutation like π such that $\pi e1=e2$. For this reason, the

symmetry property is sometimes referred to as edge-transitive. The complete graph, K_n, and the cycle graph, C_n, are both symmetric. The $p \times q$ torus network also has the NS property under the assumption of $p,q \geq 3$; if $p=q$, it is also symmetric. The q-D torus is the optimally connected (OC) with the condition $\kappa=\lambda=\delta=2q$.



(a) (b)

Figure 8: The frequency counts of ER and RRG random graphs with NS characteristic; a total of 1000 graphs of each type were created. In panel (a), the average degree of the graphs is assumed to be 3 and the frequency of NS is plotted with respect to the network size (N), while in panel (b), the frequency of graphs with NS feature is plotted with respect to the degree of the nodes.

If the graph G is symmetric and connected, it must have the property of NS (vertex-transitive) and it is also regular. The symmetry of a graph can therefore also mean its regularity. However, the reverse is not always true. That is, the graph G can be regular, but not NS. For example, de Bruijn graphs are regular, but do not have the property of symmetry. A theorem [36, 37] states that for any connected graph with NS characteristic and minimum degree $\delta \leq 4$, the relation $\kappa=\delta$ is hold. Moreover, if a graph is symmetric, regular and connected but does not have the NS property, then it must be a bipartite graph. The graph $K_{n,n}$ is both vertex- and edge-transitive; however, the graph $K_{m,n}$, $m \neq n$, is not vertex-transitive. The star graph ($K_{1,n}$) is edge-transitive, but not vertex-transitive (NS). Although it exhibits a high degree of heterogeneity among the graphs, it has a low fault tolerance and robustness.

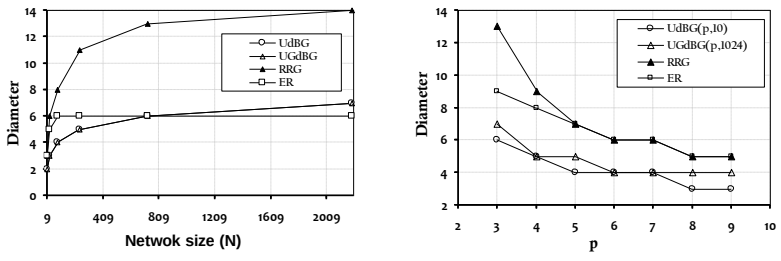
7. Numerical Results

In this section, we assess the topological properties and robustness of de Bruijn graphs against random failures and targeted attacks through experimental simulations. To facilitate meaningful comparisons, we contrast various de Bruijn graph types, as detailed in preceding sections, with the Erdős–Rényi random graph (ER) [38] and the regular random graph (RRG) [10, 12]. Our evaluation encompasses various aspects of resilience parameters.

Figure 9 displays various diameter values obtained from simulation experiments on different de Bruijn graph topologies, alongside ER and RRG. To ensure a fair comparison, the networks are assumed to be nearly equal in size. In panel (a) on the right, the parameter p is held constant at 3, while the values of parameter k (number of switches) vary for de Bruijn networks. In panel (b) on the left, the network size is set to 1024 nodes, resulting in a parameter k of 10 for UdBG and 1024 for UGdBG networks. The horizontal axis represents different values of parameter p (alphabet size or number of switch ports).

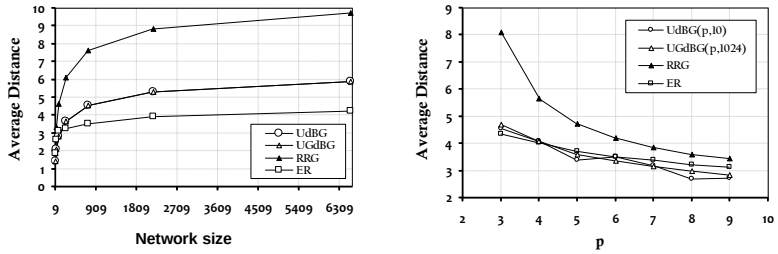
As observed in panel (a), the diameter increases as the values of k change, assuming p is constant, and as the network size increases. Notably, the increase in diameter is smallest for de Bruijn graphs compared to other networks, with ER graphs exhibiting the largest diameter. The plots in panel (b) demonstrate that, for constant k (network size) and variable parameter p, a decrease in diameter occurs with an increasing degree in the graphs. However, UdBG and GdBG networks exhibit much smaller diameters compared to ER and RRG graphs. This important feature underscores the agreement between experimental results and theoretical findings.

Figure 10 compares the average distance in various de Bruijn graphs alongside ER and RRG graphs under two scenarios. In panel (a), p is held constant at 3, while k varies. In panel (b), k is maintained at 1024, while p is variable. Examining the plots reveals that, when the network size changes, ER graphs have a smaller average distance than other networks. Conversely, de Bruijn graphs exhibit a smaller average distance. However, in panel (b), the advantageous property of small average distance for random graphs diminishes, and the average distance becomes higher compared to de Bruijn graphs, though still smaller than that of the RRG graph.



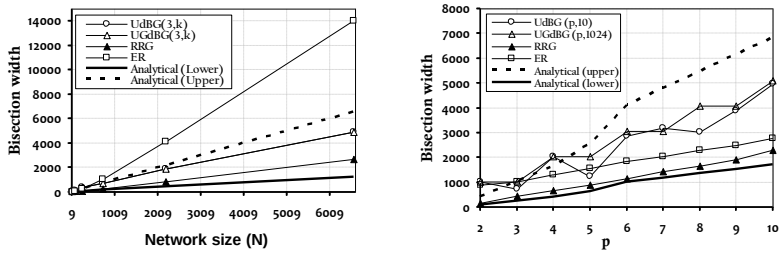
(a) (b)

Figure 9: The diameter of different types of de Bruijn topologies, Erdős–Rényi Random Graph (ER) and Regular Random Graph (RRG); (a) the parameter *k* (number of switches in the network) is assumed to be variable, while the parameter *p* (degree of nodes or number of switch ports) is assumed to be constant and equal to 3; (b) the parameter k is assumed to be constant and equal to 1024, while the parameter p is assumed to be variable.



(a) (b)

Figure 10: Average distance for different types of de Bruijn topologies, Erdős–Rényi Random Graph (ER) and Regular Random Graph (RRG); (a) the parameter *p* (the degree of the nodes or the number of switch ports) is assumed to be constant and equal to 3, while the horizontal axis represents the size of the network (*N*). For the UdbG network, the parameter k is equal to the number of switches in the network, $k = \log_3 N$, and for the UGdBG network, this parameter is equal to the size of the network; (b) the size of the networks is assumed to be fixed and equal to 1024 nodes. The parameter *k* is equal to 10 and 1024 for UdBG and UGdBG, respectively; while *p* varies between 2 and 10.



(a) (b)

Figure 11: Bisection width for different types of de Bruijn topologies, Erdős-Rényi Random Graph (ER) and Regular Random Graph (RRG); (a) in de Bruijn graphs, the parameter *p* (number of switch ports) is assumed to be constant and equal to 3, while the size of the network is variable; (b) the network size is assumed to be fixed and equal to 1024 nodes; the parameter *k* for the UdBG and UdGB-UdBG networks is assumed to be 10 and 1024 respectively; while the parameter p is variable. The lower and upper bounds in the analytical Equation (12) are shown as solid and dotted lines, respectively.

When implementing interconnection networks, a common challenge arises from the physical constraints, primarily the available wiring area. This constraint is dictated by packaging techniques, a crucial consideration for VLSI systems that often contend with wiring limitations [3]. The silicon area required for these systems is defined by the communication area, and the delay introduced by internal connections can impede the efficiency of communication networks. Additionally, as these networks are scaled to larger dimensions, the available wiring space grows exponentially with k, concomitant with the increase in network traffic volume. Consequently, networks cannot be scaled to arbitrarily large sizes without encountering wiring limitations. The selection of network dimension is influenced by how effectively the resulting topology can be scaled within the available area. A key metric for assessing the efficiency and reliability of networks is the bisection width. This measure signifies the minimum number of wires that must be removed to divide the network into two equal sets of nodes, with the collective bandwidth on these wires referred to as the bisection width.

Figure 11 illustrates the bisection width of various de Bruijn graphs, specifically UdBG and UGdBG, in comparison with ER and RRG graphs. In panel (a), the network size is held constant, while in panel (b), the parameter *p*, representing the degree of nodes (i.e., the number of switch ports), is kept fixed. Observing panel (a), it becomes evident that the RRG graph exhibits the lowest bisection width, while the ER graph displays the highest. De Bruijn graphs fall between these extremes, with UdBG and UGdBG having nearly identical bisection widths. In panel (b), the plots demonstrate an increase in bisection width for all graphs, with de Bruijn graphs experiencing a greater rate of increase compared to the other graphs. To validate the analytical relationships against simulation results, the upper and lower limits of the bisection width, as expressed in Equation (12), are depicted as a continuous line and a dashed line, respectively.

The degree of a node in the network may not always provide sufficient information about the communication patterns among its neighbors. The clustering coefficient, a parameter that assesses the probability of neighbors being connected to each other, offers insights into the network's local structure. The clustering coefficient for each node is derived from the degrees of its neighboring nodes. In essence, the local clustering coefficient for a node signifies the likelihood that two of its neighbors are connected. This metric measures the local density of network links within the vicinity of a given node. A higher clustering coefficient indicates a greater internal connection density among a node's neighbors. By averaging the clustering coefficients of all nodes, the global clustering coefficient of the network is obtained, representing the probability that two neighbors of a randomly selected node are connected. In summary, the local clustering coefficient delves into "how well do friends know each other?" while the global clustering coefficient generalizes this concept to "how well do friends know each other's friends?"

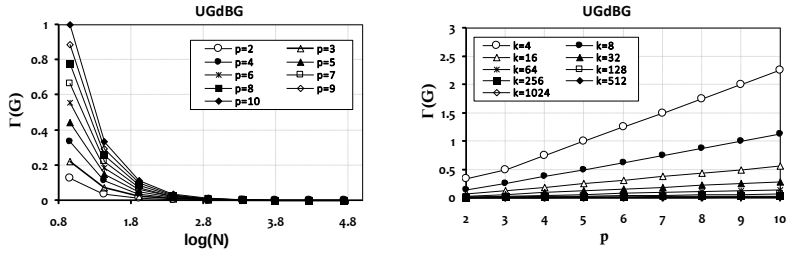
The clustering coefficient of the generalized de Bruijn graph can be expressed as [5]

$$\Gamma(G) = \begin{cases} (p-1)/k & p \geq 3 \\ 1/(k-2) & p = 2 \end{cases} \tag{17}$$

The clustering coefficient parameter quantifies the number of triad relationships within the network. For random graphs, it has been established that this parameter is equivalent to the average ratio of the degree to the network size [5]. Figure 12 illustrates the parameter $\Gamma(G)$ in the UdBG network for various values of k and p. The graph indicates an inverse relationship between the clustering coefficient and the network size, as it decreases with an increase in the network size. In panel (b), an increase in p corresponds to an increase in the clustering coefficient, while an increase in k results in a decrease in the clustering coefficient across all network models. Thus, the local clustering coefficient of nodes, and consequently the global clustering of the network, is inversely proportional to k, as explicitly stated in the analytical Equation (17).

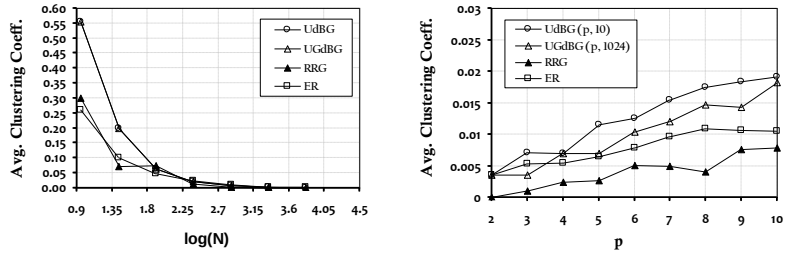
Figure 13 presents clustering coefficient plots for various de Bruijn graphs, including UdBG and UGdBG, as well as ER and RRG networks. In panel (a), the size of the network changes, while the parameter p (the number of switch ports) is fixed in all diagrams. In panel (b), the parameter k (the number of nodes) is assumed to be fixed. As depicted in the diagrams in panel (a), with a constant value of p=3, the clustering coefficient in all models decreases as the network size increases. This aligns with the expected behavior, indicating that the local clustering coefficient of nodes and, consequently, the global clustering of the network in de Bruijn graphs will decrease proportionally to $1/k$. This relationship is also explicitly expressed in the analytical Equation (12), which shows the clustering coefficient has an inverse relationship with the network size. The panels (a) and (b) in Figure 13 demonstrate that ER and RRG graphs have lower clustering coefficients than de Bruijn graphs.

According to [5], if a ball of radius n is constructed around a node like v, encompassing all nodes reachable from v within a maximum of n hops, then the upper bound of the ball for the de Bruijn graph will be equal to k–1. This is because the neighborhood expansion is retained for all balls smaller than the graph itself, and due to the nature of global clustering; the path overlap in de Bruijn graphs is not extensive. In other words, there is a high probability of finding short parallel pathways to any desired destination in the graph. De Bruijn graphs are not considered graphs with high node connectivity due to the presence of self-loops. That is, there are no distinct disjoint pathways between any two nodes in the graph.



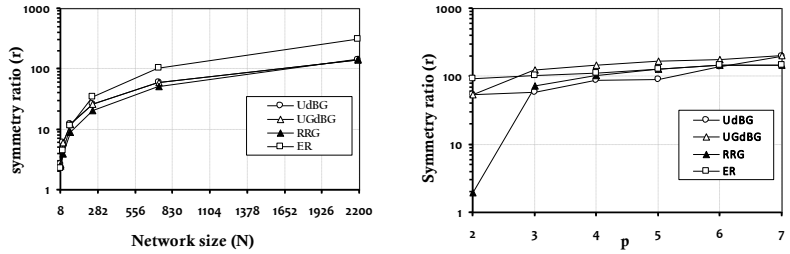
(a) (b)

Figure 12: The clustering coefficients that were extracted from analytical Equation (17) for UGdBG; (a) The diagrams showing the clustering coefficients in terms of the logarithm of the network size for various values of the parameter p; (b) the clustering coefficients that were obtained from analytical Equation (17) for UGdBG when the parameter k was changed.



(a) (b)

Figure 13: The average clustering coefficient for various types of de Bruijn topologies, including Erdős–Rényi Random Graph (ER) and Regular Random Graph (RRG); (a) the parameter *k* is variable, although the parameter *p* is constant and equal to 3; the degree in the RRG graph is considered to be constant and equal to 4. The number of nodes in the ER graph is equal to the number of nodes in the UGdBG, and the values of the connection probability are assumed so that the number of edges is about comparable to the number of edges in other graphs for a fair comparison; panel (b) assumes that value *k* is constant and equal to 10, whereas parameter p is variable.



(a) (b)

Figure 14: The symmetry ratio and efficiency of de Bruijn graphs, Erdős–Rényi Random Graph (ER) and Regular Random Graph (RRG); (a) the network size varies while the number of switch ports remains constant; (b) the number of switch ports remains constant while the network size equals 1024 nodes; the vertical axis in both panels (a) and (b) is logarithmic.

In the subsequent sections, we introduce and employ a novel criterion for evaluating symmetry, termed the symmetry ratio. This criterion can be formulated as [39]

$$r = \varepsilon / (D+1) \tag{18}$$

where, the parameter ε is the number of distinct eigenvalues of the adjacency matrix and D is the diameter of the graph.

As discussed earlier, enhancing network efficiency through the symmetry characteristic can improve its resilience. However, determining how to characterize network symmetry has always been a question. Counting the number of automorphisms in a network, i.e., the size of the automorphism group is a simple solution. However, some networks have a high number of automorphisms, making it an unreliable measure for assessing graph symmetry. Therefore, the symmetry ratio criterion can be employed effectively to evaluate network resilience to attacks.

Figure 14 illustrates the symmetry ratio for various de Bruijn graphs, ER, and RRG networks. In panel (a), this measure is presented concerning network size for different graph types, assuming a constant number of ports on network switches. Panel (b) shows the symmetry ratio for various graphs based on the number of ports in the switch, while keeping the network size fixed at 1024 nodes. The vertical axis is represented logarithmically in both panels. As observed in the plots, the symmetry ratio increases with the size of de Bruijn graphs, ER, and RRG networks. This increase indicates a reduction in the symmetry feature within the graphs, leading to increased fragility, decreased resilience, and lower efficiency (i.e., fewer messages successfully reaching the destination). The symmetry ratio is bounded between 1 and $n/3$ [39] for any graph of size $n \geq 3$, where $r=1$ signifies a high level of symmetry in the graph. It has been demonstrated that in random graphs, as the number of nodes increases, r tends to the size of the graph [39].

The upcoming sections will introduce and analyze fragility and resilience metrics for various de Bruijn graphs, ER, and RRG random networks. Before delving into these metrics, we need to discuss percolation in complex networks, a concept related to these conditions. Percolation involves deleting a subset of nodes or edges from a network [1]. In network theory, the giant component (GC) concept is employed to assess the robustness and resilience of networks [1]. The GC signifies the connectivity of network nodes and is useful for describing information propagation in a network's structure. We specifically focus on the dynamic changes in the network structure resulting from the removal of nodes and how these changes impact the GC. A small GC suggests potential compromise, indicating low system robustness and high fragility. Let G_q be a subgraph of G created by removing a proportion of q nodes from G, with size k. If G'_q denotes the connected component larger than G_q, the fraction σ_q can be defined as

$$\sigma_q = |G'_q| / k \tag{19}$$

The value of σ enables us to assess how graph G responds to the removal of a fraction q of nodes. After selecting a centrality criterion for the percolation process, nodes can be ranked based on this criterion. The process then involves gradually increasing the fraction of nodes to be removed from the graph, depending on the type of failure (attack), and calculating σ_q. In the literature [1], the measure of robustness is known as the R-index, defined by averaging over the function.

$$R = \frac{1}{k}\sum_{i=1}^{k}\sigma_{i/k} \tag{20}$$

in which, the normalization factor $1/k$ allows for the comparison of the resilience of networks of varying sizes. The quantity R has a minimum value of $1/k$, which can be calculated using star graphs. Conversely, the maximum value of R for the complete graph is $(1-1/k)/2$. Thus, for each graph, $R \in [0,1/2]$. It is worth noting that occasionally, instead of R, its complement, i.e., the index $V=1/2-R$, is used, referred to as network vulnerability.

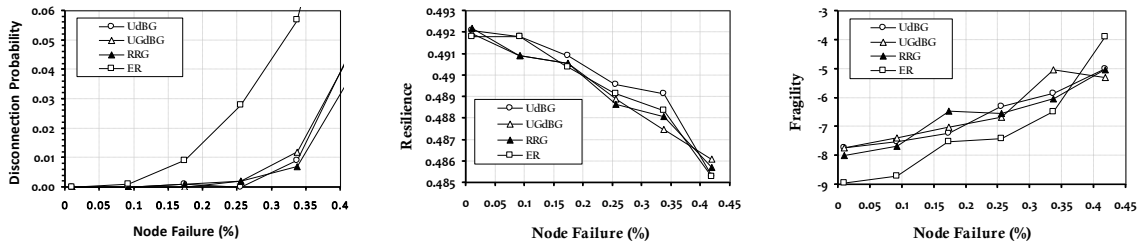
The objective of this section is to determine whether random failures and targeted assaults have the same impact on network resilience and fragility, or if these two criteria are fundamentally different. As per the definition, network fragility refers to the sensitivity of a network to the loss of nodes. Typically, the fragility measure is determined by calculating the greatest distance between the eigenvalues of the adjacency matrix instability and the x-axis. The network fragility can be expressed as [40]

$$F = -\max\{\mu_i(A)\}_{i=1}^{N} \tag{21}$$

where $\mu_i(A)$ represents the i-th eigenvalue of the graph G adjacency matrix A, and the F (fragility) also signifies the greatest distance between the eigenvalues of the adjacency matrix instability and the imaginary axis. It should be noted that if all of A's eigenvalues are negative, the system is said to be unstable. The greater the value of F, the more the network moves away from its stability region. As a result, lower F values indicate that the network is more fragile. It should also be emphasized that the biggest eigenvalue in a graph reveals its structure and hence its robustness. The fragility and resilience of networks are demonstrated to have a linear negative correlation in double logarithmic coordinates in [40]. This indicates that focused assaults have a greater influence on network resilience than they do on vulnerability.

The numerical results of the resilience and fragility metrics on various de Bruijn graphs, ER, and RRG network models are shown in Figure 15. Panel (a) depicts the likelihood of network disconnection as a random failure of nodes. It can be observed, when the failure rate of nodes grows, so does the likelihood of disconnection. This rise is more for ER graphs and smaller for RRG graphs than for others. Panel (b) depicts network resilience in terms of node failure rate using Equation (21). Panel (c) depicts network fragility in terms of node failure rate.

It is obvious that raising the failure rate of nodes reduces the amount of resilience while increasing the fragility. The crucial aspect to emphasize here is that the link between network resilience and fragility is not linear. To demonstrate a link between resilience and fragility, we ran a series of simulations on various underlying networks, so that each outcome is the product of 1000 trials and then averaged. This issue is demonstrated in Figure 16 by simulation results on several network topologies with random node rates of 10%, 20%, and 50%.



(a) (b) (c)

Figure 15: The resilience of de Bruijn graphs, ER, and RRG networks; the network is supposed to have 243 nodes. The simulation outcomes were run 1000 times and then averaged: (a) the likelihood of network disconnection in terms of node failure rate; (b) the resilience of underlying networks in terms of node failure rate; and (c) network fragility in terms of node failure rate.

One of the critical characteristics for ensuring that an interconnection network architecture works properly with the routing algorithm and its variants is how to distribute routes in a balanced manner over all connections in that topology. In an ideal world, for each uniformly chosen pair of nodes, all network links should be picked with an equal likelihood to take these pathways. The maximum traffic load for all-to-all communication reflects this attribute. Typically, the initial value of the load is zero, and one unit of traffic is transferred between each pair of nodes. Then, for each pair of nodes that uses x routes to transfer traffic load, if one of these routes uses the assumed connection, the traffic load of this link will rise by $1/x$. Thus, the maximum link load is the total traffic load on that link after accounting for all node pairs in the network.

Table 2 includes the characteristics of various types of de Bruijn graphs, ER, and RRG networks in terms of network size (N), number of edges, diameter (D), optimal connectivity (OC), node similarity (NS), average traffic load ($<L>$), maximum load (L_{max}), word-representability, and half-transitivity. Random or targeted attacks on the graph nodes and edges have a major influence

on the graph disintegration and, as a result, weaken its resilience. Furthermore, failure at the network nodes and edges has a significant impact on the traffic load of the links.

Table 1: Algebraic equations of fitting polynomials (see Figure 16) to demonstrate the relationship between resilience and fragility in various de Bruijn graphs, ER, and RRG network models.

	Node failure rate	Fitting polynomial
UdBG (3,5)	0.1	$0.49288 - 0.00088\,x - 0.00006\,x^2 - 0.00005\,x^3$
	0.3	$0.491203 - 0.00145\,x - 0.000028\,x^2 - 0.00025\,x^3$
	0.5	$0.473667 - 0.01424\,x - 0.004741\,x^2 + 0.002272\,x^3$
UGdBG (3,243)	0.1	$0.492996 - 0.000658\,x + 0.0000065\,x^2 + 0.000013\,x^3$
	0.3	$0.491061 - 0.00154\,x - 0.000071\,x^2 + 0.000051\,x^3$
	0.5	$0.465253 - 0.03196\,x - 0.011855\,x^2 + 0.002623\,x^3$
RRG	0.1	$0.4931848 - 0.000539\,x + 0.000083\,x^2 - 0.000046\,x^3$
	0.3	$0.239008 - 0.27676\,x + 0.01029\,x^2 + 0.09101\,x^3$
	0.5	$0.016315 - 0.03299\,x + 0.054936\,x^2 - 0.049538\,x^3$
ER	0.1	$0.49299 - 0.000424\,x + 0.000168\,x^2 - 0.000043\,x^3$
	0.3	$0.455711 - 0.027911\,x - 0.025932\,x^2 - 0.034291\,x^3$
	0.5	$0.270723 - 0.170284\,x - 0.03038\,x^2 - 0.086709\,x^3$

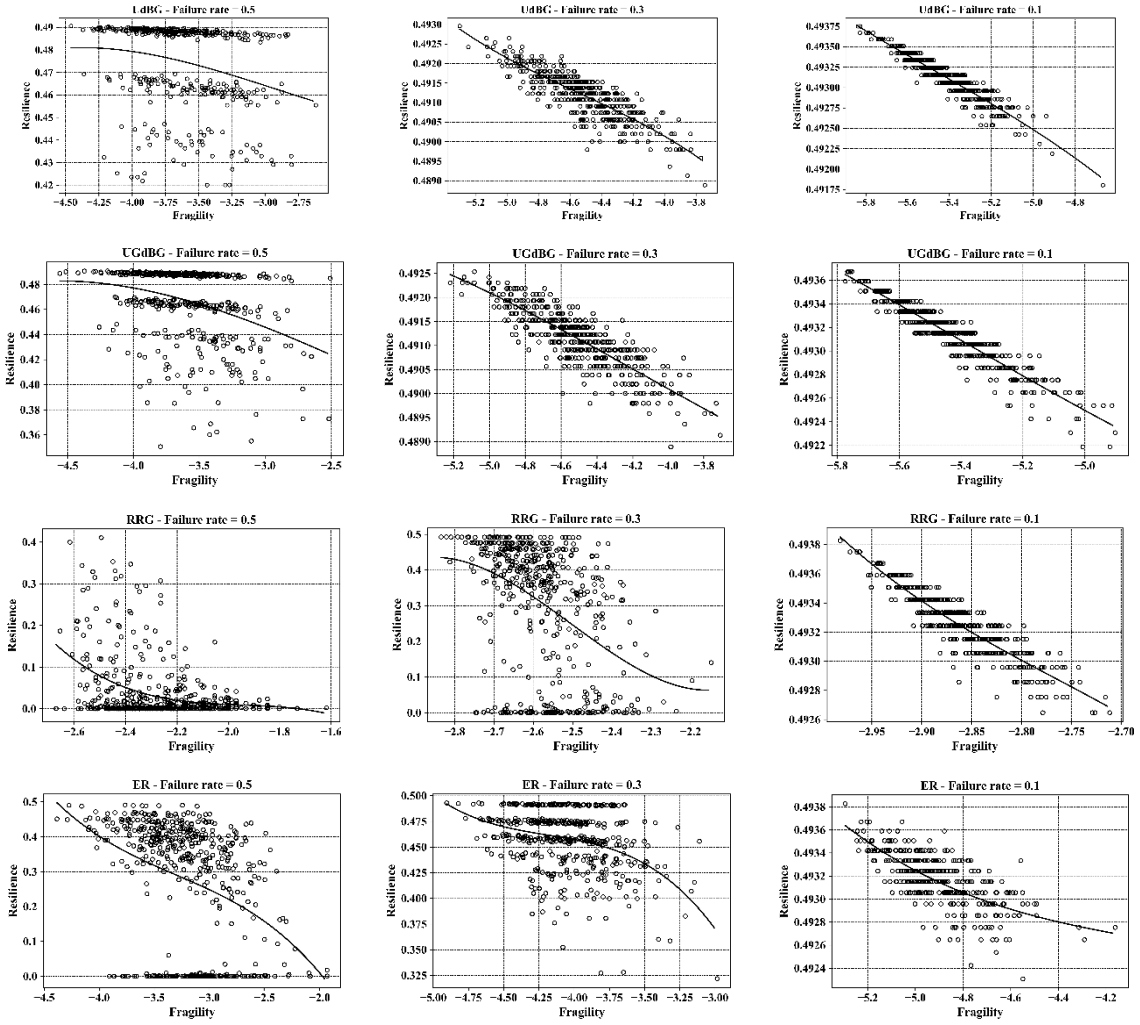


Figure 16: The relationship between resilience and fragility of networks at different node failure rates (10%, 30%, and 50%) in UdBG(3,5), UGdBG(3,243), ER, and RRG networks with 243 nodes, average degree equal to 3, and the connecting probability equal

to 0.054; additionally, to illustrate the nonlinear relationship between resilience and fragility, a polynomial of degree 3 has been fitted to the data obtained from the simulation experiments. Table 1 shows the algebraic equations for these polynomials.

Table 2: A summary of the network size (N), number of edges, diameter (D), optimal connectivity (OC), node similarity (NS), average traffic load ($<L>$), maximum load (L_{max}), word-representability, and semi-transitive properties of de Bruijn graphs, random regular graphs (RRG), and Erdős–Rényi random graphs (ER).

Graph	N	#Edges	D	OC	NS	Symmetry	$<L>$	L_{max}	Word-Rep.	Semi Transitive
UdBG (2,4)	16	29	4	True	False	False	8.86	14	True	True
UdBG (2,5)	32	61	5	True	False	False	22.39	38	True	True
UdBG (2,6)	64	125	6	True	False	False	55.7	91	True	True
UdBG (3,4)	81	237	4	True	False	False	38.73	55	False	True
UGdBG (3,81)	81	237	4	True	False	False	38.73	55	False	True
RRG	81	123	8	True	False	False	125.22	228	False	True
ER	81	180	6	True	False	False	56.34	134	False	True

As a result, establishing the OC feature in the network might be critical in decreasing the impact of component failure on network traffic load. Edge connectivity, in general, indicates that there exist λ edge-channels connecting network nodes and that traffic load may be spread on the shortest of these paths, avoiding network congestion to a considerable extent. When the vertex connectivity is less than the edge connectivity ($\kappa < \lambda$), the number of different pathways of the edge may see a severe decline in the network once a number of nodes are destroyed.

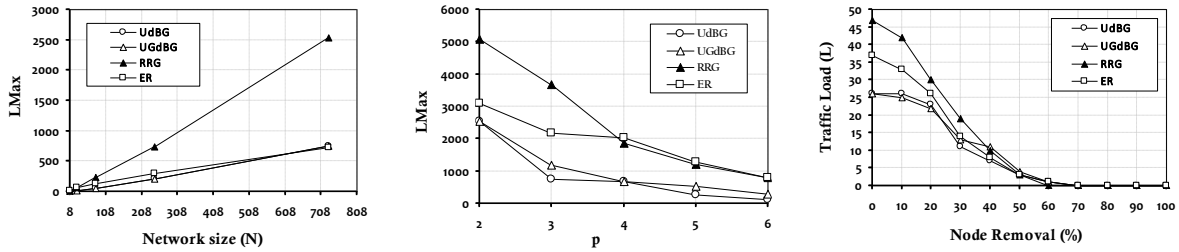
However, when the OC condition is established in a graph, node failure may only render one of the disjoint pathways of the edges connecting these nodes invalid. Further, providing the symmetry characteristic (link similarity) in networks helps to balance the traffic load in them. That is, each connection bears a fixed L traffic load. As a result of the asymmetry in the graph, certain connections must carry higher traffic load than others (L_{max}), resulting in increased traffic and congestion in the graph.

Theorem 5 [36]: *If G is a graph of size N, dimension D, and minimal degree δ with the characteristic NS and symmetry, then its traffic load is given by*

$$L = \bar{d}(N-1)/\delta \tag{22}$$

Corollary [36]: *If graph G is not symmetric, the minimum traffic load is equal to L ($L_{max} \geq L$), which must be at least $ND/(2\delta)$.*

Figure 17 depicts the maximum load for various de Bruijn topologies, RRG, and ER networks in order to compare load balancing features. The shortest distance routing algorithm is used in all topologies. Panel (a) depicts the changes in maximum load of the connections as the network size grows, while the parameter p remains constant and equal to 3. Panel (b) depicts the nodal degree when the number of switches (network size) is fixed at 1024. To determine the maximum traffic load, the number of routes between each pair of nodes in the relevant topology is averaged.



(a) (b) (c)

Figure 17: Maximum network traffic load for various de Bruijn topologies, random regular graphs (RRG), and Erdős–Rényi random graphs (ER); (a) The horizontal axis is in terms of network size, with the parameter p (number of switch ports) considered to be constant and equal to 3; (b) The horizontal axis is in terms of network size, with the parameter p assumed to be variable.

Both panels (a) and (b) show that as network size grows, so does the maximum traffic load. It is also evident that there is a significant difference between the maximum link loads in the RRG graph and other graphs. Furthermore, UdBG and UGdBG topologies spread traffic load across all network links significantly more equitably. De Bruijn topologies attain a nearly ideal maximum link load in all circumstances in this manner.

Tables 2 and 3 highlight the impact of network node and edge failures on rising traffic load in de Bruijn topologies, RRG, and ER networks. The findings of these two tables show that none of the networks exhibit either the symmetry or the NS features,

and that only the OC feature is present in all of them. The value <*l*> in the table represents the average traffic load across all edges or nodes. In reality, it is frequently possible to rank edges and nodes based on greatest traffic in these circumstances.

To compute the percentage of traffic load changes (Δ<*l*>%), the average traffic load difference before and after the attack on all edges or nodes of the graph is calculated and then normalize it to the network average traffic load before the assault. It should also be noted that the ma*x*imum traffic load of the network after the attack (L^{+}_{max}) may be computed in the same way by dividing the ma*x*imum traffic load before and after the assault by the difference. Hundreds of random attacks were carried out in the simulation trials and then averaged.

Figure 18 displays the traffic load variations for de Bruijn graphs, ER, and RRG networks in terms of network nodes and edges before and after attack. The vertical a*x*is in all plots reflects traffic load, while the horizontal a*x*is displays the number of edges or nodes that are ranked depending on the network traffic load rating. The panels on the left, (a) and (c), indicate how many network edges or nodes have a specific quantity of traffic, whereas the panels on the right, (b) and (d), show the traffic distribution following a 20% random attack on the network components. Edge and node attacks are also possible. Because no node is deleted in edge attack mode, the total network traffic burden stays constant. However, the number of network edges is reduced as a result of edge removal. As a result, the ratio of total edge traffic to the number of edges grows. This means that, in general, we witness an increase in average network traffic load changes, as seen in Table 4.

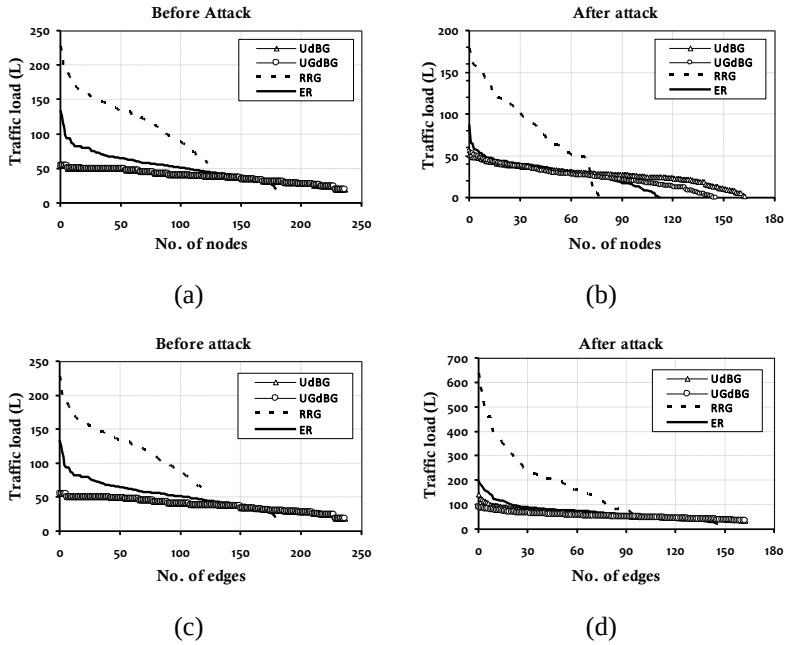


Figure 18: Traffic load variations (*L*) before and after attacks on nodes (panels (a) and (b)) and edges (panels (c) and (d)) of de Bruijn graphs, Erdős–Rényi random graph (ER), and Regular random graph (RRG); The network has 81 nodes, with an e*x*pected average number of edges of 200.

Table 3: The percentage increase in traffic load caused by random attacks on nodes ($f_{\text{node-attack}}$%) in de Bruijn graphs, ER, and RRG networks. Summary of graph characteristics in terms of network size (*N*), number of edges, diameter (*D*), optimal connectivity (OC), node similarity (NS), average traffic load (<*L*>), ma*x*imum load (Lma*x*), percentage of average traffic load variation (Δ<*L*>%), and ma*x*imum traffic load increase after attacking the nodes (L^{+}_{max}).

Graph	OC	NS	Symmetry	$<L>$	L_{max}	$f_{node\text{-}attack}$%	$\Delta<L>$%	L^{+}_{max}
UdBG(3,4)	True	False	False	38.73	55	1	-1.85	56.4
	True	False	False	38.73	55	10	-17.07	61.87
	True	False	False	38.73	55	20	-30.81	63.01
	True	False	False	38.73	55	30	-44.93	62.77
	True	False	False	38.73	55	40	-56.77	52.35
UGdBG(3,81)	True	False	False	38.73	55	1	-1.72	56.32
	True	False	False	38.73	55	10	-17.25	62.47
	True	False	False	38.73	55	20	-31.37	61.49
	True	False	False	38.73	55	30	-44.12	57.9
	True	False	False	38.73	55	40	-56.31	54.04
RRG	True	False	False	125.22	228	1	-1.12	231.22
	True	False	False	125.22	228	10	-9.72	251.83
	True	False	False	125.22	228	20	-21.97	269.84
	True	False	False	125.22	228	30	-44.76	210.15
	True	False	False	125.22	228	40	-79.68	90.94
ER	True	False	False	56.34	134	1	-2.27	133.18
	True	False	False	56.34	134	10	-19.3	121.86
	True	False	False	56.34	134	20	-37.52	100.4
	True	False	False	56.34	134	30	-48.9	83.19
	True	False	False	56.34	134	40	-65.91	64.87

Attacks on network nodes have a bigger impact than attacks on edges, and by eliminating one node, additional edges related to that node are removed from the network. As a result, the ratio of total traffic to the number of edges, or the average variation in traffic load, reduces even further. As shown in Table 3, the percentage of average traffic load changes ($\Delta<L>$%) in the case of node attacks is negative across all networks. This suggests that the numerator in this ratio, i.e. total traffic load, has reduced significantly more than traffic load itself. As the values in Tables 3 and 4 show the quantity of L_{max} has always increased in all scenarios of attacks on network edges and nodes. As a result, these attacks might cause bottlenecks in portions of the network, resulting in a sharp increase in traffic in these locations.

The degree to which a network may be split into modules or independent communities is referred to as modularity in network architecture. A modular network is made up of interconnected modules, with nodes inside each module densely coupled to each other and fewer connections between modules. Modularity is a measure of the degree of separation between modules in a network that is frequently used to study the structure and organization of complex networks. Figure 19 depicts the network modularity metric for several types of de Bruijn graph, ER, and RRG topologies. The horizontal axis in panel (a) represents the logarithm of network size, whereas the parameter p (number of switch ports) is assumed to be constant and equal to 3.

Panel (b) assumes that the network size is set and equal to 1024 nodes, but the parameter p is variable. The graphs in panel (a) indicate that as network size rose and the number of switch ports (degree of nodes) remained constant, the modularity of all networks increased and ultimately reached a constant value.

Even if the randomness of RRG and ER networks is higher than that of de Bruijn graphs of the same size, the modularity of de Bruijn graphs rises as the network size grows. The modularity diagrams are presented in panel (b) according to the degree of the network. It can be observed that keeping the network size constant while raising the degree of the network leads a complete loss of modularity; however this loss is significantly worse for ER and RRG than for de Bruijn graphs. High modularity is inversely related to the resilience of a regular modular graph and reduces the overall robustness of the network. According to the simulation data, higher modularity makes the network more vulnerable to attacks. In particular, regular modular graphs are more vulnerable to module-based (MB) attacks than other types of attacks that target interconnected nodes with high betweenness centrality. When the

modularity of the network exceeds a certain threshold, the robustness of regular modular graphs becomes much more vulnerable than that of scale-free (SF) networks. Consequently, high modularity in regular modular graphs significantly reduces their robustness and vulnerability to malicious attacks.

The results of the simulation depicted in Figure 20 analyze the impact of modularity on the robustness of various de Bruijn graphs, ER, and RRG topologies. The network size is fixed at 243 nodes. Panel (a) illustrates the correlation between the modularity measure and the random failure rate of network nodes. Across all graphs, an escalation in the percentage of node failures from f=10% to f=50% results in diminished resilience and an augmented level of network modularity, implying an inverse relationship between this criterion and network reliability.

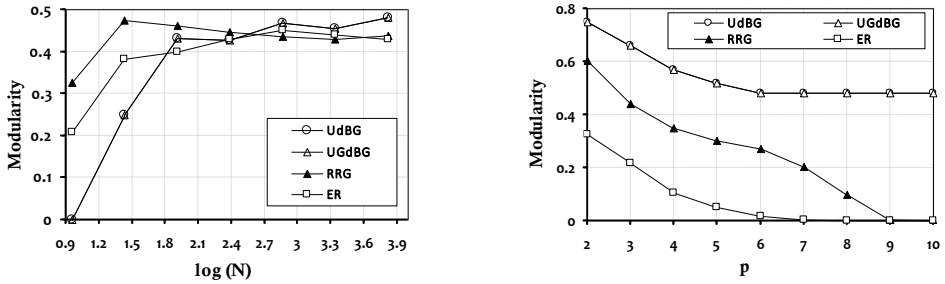
Table 4: The percentage increase in traffic load caused by random attacks on edges ($f_{edge\text{-}attack}$%) in de Bruijn graphs, ER, and RRG networks. Summary of graph characteristics in terms of network size (N), number of edges, diameter (D), optimal connectivity (OC), node similarity (NS), average traffic load ($<L>$), maximum load (L_{max}), percentage of average traffic load variation ($\Delta<L>$%), and maximum traffic load increase after attacking the nodes (L^{+}_{max}).

Graph	OC	NS	Symmetry	$<L>$	L_{max}	$f_{edge\text{-}attack}$%	$\Delta<L>$%	L^{+}_{max}
UdBG(3,4)	True	False	False	38.73	55	0.01	1.41	58.78
	True	False	False	38.73	55	0.1	17.33	79.96
	True	False	False	38.73	55	0.2	41.62	107.27
	True	False	False	38.73	55	0.3	73.64	145.85
	True	False	False	38.73	55	0.4	128.15	211.68
UGdBG(3,81)	True	False	False	38.73	55	0.01	1.4	58.33
	True	False	False	38.73	55	0.1	17.66	79.45
	True	False	False	38.73	55	0.2	41.23	108.49
	True	False	False	38.73	55	0.3	76.89	150.1
	True	False	False	38.73	55	0.4	122.6	206.33
RRG	True	False	False	125.22	228	0.01	2.24	236.07
	True	False	False	125.22	228	0.1	29.53	337.99
	True	False	False	125.22	228	0.2	76.46	595.63
	True	False	False	125.22	228	0.3	102.86	805.25
	True	False	False	125.22	228	0.4	23.26	606.42
ER	True	False	False	56.34	134	0.01	1.5	136.14
	True	False	False	56.34	134	0.1	17.2	163.9
	True	False	False	56.34	134	0.2	38.63	205.42
	True	False	False	56.34	134	0.3	66.42	255.6
	True	False	False	56.34	134	0.4	104.08	352.89

The de Bruijn graphs are positioned between the boundaries of ER and RRG random graphs. In panels (b) and (c), the correlation between modularity and fragility or resilience of the network is depicted at various node failure rates ranging from f=10% to f=50% for the same networks. Each data point in these graphs represents an average failure probability derived from multiple simulation runs. The figures vividly illustrate how the high modularity of RRG networks can impact their robustness compared to de Bruijn graphs and ER topologies. An interesting observation is that a low degree of modularity in graphs can create the illusion of a small-world with high performance. The reduction in modularity leads to a decrease in the density of connectivity within the network's modules, while simultaneously increasing the density of connectivity between the modules. This results in the establishment of shortcut connections between different clusters in the network. Consequently, the average path length decreases, and the small-world

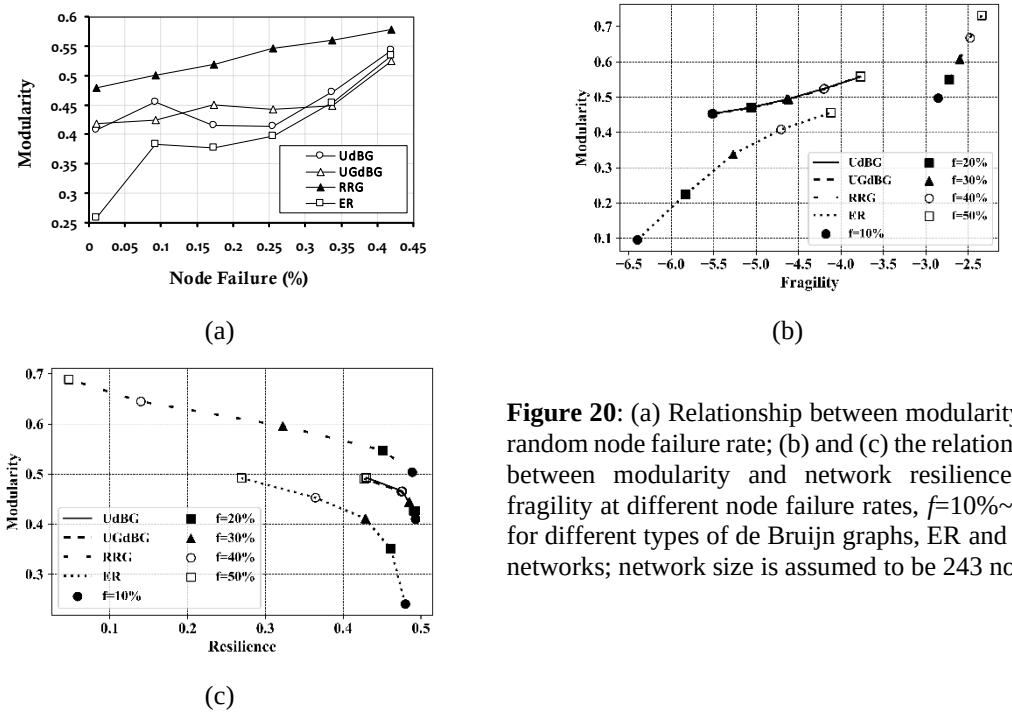
characteristic emerges. In this context, minimal modularity in graphs could enhance the coexistence of high resilience and path efficiency. This is achieved by facilitating effective communication between different modules while maintaining a certain level of connectivity within each module. Consequently, the modularity metric in network architecture proves to be beneficial for evaluating both resilience and performance. To summarize the contents, the resilience of de Bruijn graphs, as well as ER and RRG random networks, has been analyzed from several perspectives using the best known robustness criteria widely utilized in the literature. Table 5 highlights the features and numerical values of these networks various resilience parameters. The number of nodes is set to 81, while the average number of edges is set to 200.

In addition to the maximum degree, average degree, clustering coefficient, diameter, modularity, average distance, resilience and fragility discussed in this article, a variety of other metrics can be used to evaluate the robustness of such a network. Transitivity index [41], spectral radius [41], algebraic connectivity [41], spectral gap [41], network criticality [41], natural connectivity [41], effective resistance [41], energy [42], Laplacian energy [42], average subgraph centrality [43], and *Z*-Estrada index [44] are some of the commonly used criteria for graph resilience in the literature. In general, graphs with higher resilience have higher values for these criteria. The statistics in Table 5 clearly show that de Bruijn graphs have a high degree of resilience compared to similar graphs.



(a) (b)

Figure 19: Network modularity for various types of de Bruijn graph, ER, and RRG topologies; (a) The horizontal axis is the logarithm of the network size, while the parameter *p* (number of switch ports) is assumed to be constant and equal to 3; (b) The network size is assumed to be fixed and equal to 1024 nodes, while the parameter *p* is variable.



(a) (b)

(c)

Figure 20: (a) Relationship between modularity and random node failure rate; (b) and (c) the relationships between modularity and network resilience and fragility at different node failure rates, *f*=10%~50%, for different types of de Bruijn graphs, ER and RRG networks; network size is assumed to be 243 nodes.

The statistics in Table 5 also provide some important information. We analyzed the impact of the average clustering coefficient in de Bruijn graphs. Networks with a high clustering coefficient have short paths and vice versa. As a result, the average clustering index can quantify the number and intensity of nodes in the network that tend to find clusters and groups. A low value of this metric indicates that it is unlikely that network nodes have mutual neighbors. The lower this metric is, the easier it is to propagate and

transfer information across local connections and clusters throughout the network. Several factors affect the speed of information transmission in a network, including network size, average degree, clustering coefficient, network heterogeneity, etc. There are mainly three types of criteria used to evaluate network efficiency. The first category refers to the local structure around the nodes. For example, average degree, edge density, degree heterogeneity, clustering coefficients, modularity index, average distance and average node betweenness. The second category is related to those that send information not only through short paths, but also through any available path between pairs of corresponding nodes. In fact, these metrics are based on the concept of graph traversal. For example, we can refer to eigenvector centrality [1], subgraph centrality [43] and average communicability [43]. However, there is also a third category of criteria based on all-walks indices that penalize less heavily longer walks between pairs of nodes.

The measures of the second category usually consider all walks connecting the nodes and penalize relatively long walks heavily. To include longer walks in the analysis, Estrada [44] has defined a measure, called Z-Estrada, for the adjacency matrix A as $Z = \sum_{q=0}^{\infty} A^q / q!!$ that penalizes the walks of length q with $q!!$ (double factorial). In the matrix Z, which is extracted from the underlying graph, the average of the elements on the main diagonal, i.e. $< Z_{ii} >= (\sum_{i=1}^{n} Z_{ii}) / n$, is used to involve a node such as i in all subgraphs of the graph, so that it includes larger subgraphs compared to the centrality measure of the subgraph. It is also possible to perform an averaging over all elements of the matrix Z and consider it as a measure of the global capacity of the network in transferring information between pairs of nodes. In this way, it is possible to send information over a longer range.

Table 5: Summary of specifications and numerical values for the robustness parameters of different de Bruijn graphs, ER and RRG networks.

		Networks			
Robustness metrics		UdBG (3,4)	UGdBG (3,81)	RRG	ER
Avg. clustering coef. (Γ)		0.0642	0.0642	0.0732	0.0467
Transitivity		0.0518	0.0518	0.0732	0.0476
Avg. degree		5.8519	5.8519	3.0	4.4444
Max degree (Δ)		6	6	3	10
Spectral Radius		4	4	6	4
Diameter (D)		4	4	8	6
Modularity		0.3124	0.3124	0.4621	0.0109
Avg. path length		2.8333	2.8333	4.6378	3.1302
Algebraic connectivity		1.1459	1.1459	0.2429	0.5509
Spectral gap		1.2597	1.2597	0.2429	1.1578
Network criticality		0.4374	0.4374	1.3222	0.7823
Natural connectivity		2.5577	2.5577	1.2212	2.0026
Effective resistance		1417.0924	1417.0924	4390.8867	2534.5605
Energy		140.1098	140.1098	125.7438	141.4955
Laplacian energy		144.7304	144.7304	125.7438	188.7909
Avg. subgraph centrality		12.9056	12.9056	3.3913	7.4081
Z-Estrada	$<Z_{ii}>$	951012.5793	951012.5793	12.7372	34075.0017
	$<Z_{ij}>$	940678.6713	940678.6713	2.3469	26607.3889
Fragility		-5.8898	-5.8898	-3.0	-5.2939
Resilience		0.4938	0.4938	0.4939	0.4938

The Z-Estrada metric was derived in experimental simulation results on de Bruijn graphs and on ER and RRG random networks and the numerical results are shown in Table 5. The results show that the Z-Estrada metric can better explain the degree of communication and information propagation in networks. In reality, this metric can be used to study the information transmission capacity in networks since it considers longer walks. By comparing of the numerical results of this quantity in Table 5 for the underlying graphs, it can be clearly seen that the values of the Z-Estrada measure for de Bruijn graphs are several times the RRG graph and even larger than the ER random graph. This interesting feature helps to understand the argument why the speed of spreading information in de Bruijn graphs, which are used as the infrastructure of interconnection networks and distributed hash tables (DHT) and are especially used in genome assembly, can be much more compared with other graphs.

Comparing the numerical results of this measure for the underlying graphs in Table 5, it is clear that the values of the Z-Estrada measure for de Bruijn graphs are several times higher than for the RRG graph and much higher than for the ER random graph. This intriguing property helps to explain why the speed of information dissemination in de Bruijn graphs, which are used as the infrastructure of interconnection networks and distributed hash tables (DHT) and are particularly useful in genome assembly, can be much faster than in other graphs.

8. Concluding Remarks and Future Directions

To construct a successful communication system, various considerations come into play. Key factors include cost, security, integrity, scalability, and notably, robustness. Robustness is a crucial aspect of any communication system and network, making the analysis of network robustness and resilience a significant focus in complex network research. De Bruijn graphs have emerged as viable options for organizing network connections. In communication networks, the structure of interconnected components is pivotal. De Bruijn graphs possess desirable characteristics that render them logical and suitable for assessing robustness. These include a relatively large number of nodes, a small number of connections per node, a small diameter, and the presence of short paths between nodes. In this article, we introduced and examined various types of de Bruijn graphs, explored their properties from different perspectives within graph theory. Properties such as optimal connectivity (OC) offer advantages in reducing the impact of failures and alleviating link traffic load. The small diameter of these graphs is observed to contribute to traffic load reduction. Moreover, an increase in network symmetry is identified as a means to lower traffic load and enhance the robustness. A comprehensive investigation into the reliability of de Bruijn graphs is conducted, employing different robustness measures to assess resilience and fragility against node and edge failures. As a suggestion for future work, exploring additional applications for de Bruijn graphs and evaluating their effectiveness in terms of robustness and resilience is proposed. Examples include applications in genome assembly, information theory and coding, robotics, game theory, and beyond. Despite their diverse applications, it is evident that de Bruijn graphs stand out as versatile and valuable tools, providing profound insights for researchers across various domains.

References

[1] A-L. Barabási, Network science book, Boston, MA: Center for Complex Network, Northeastern University, Available online at: *http://barabasi.com/networksciencebook*, 2023.

[2] I. Koren and C. M. Krishna, Fault Tolerant Systems, Morgan Kaufmann, 2020.

[3] J. Xu, Topological structure and analysis of interconnection networks, Springer Science & Business Media, Vol. 17, 2013.

[4] P.E. Compeau, P.A. Pevzner and G. Tesler, How to apply de Bruijn graphs to genome assembly, Nature biotechnology, Vol. 29, No. 11, pp.987-991, 2011.

[5] D. Loguinov, J. Casas, and X. Wang, Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience, IEEE/ACM transactions on Networking, Vol. 13, No. 5, pp.1107-1120, 2005.

[6] J. Baker, De Bruijn graphs and their applications to fault tolerant networks, 2011.

[7] J.L. Gross, J. Yellen, and M. Anderson, Graph theory and its applications, Chapman and Hall/CRC, 2018.

[8] A.H. Esfahanian and S.L. Hakimi, Fault-tolerant routing in debruijn comrnunication networks, IEEE Transactions on Computers, Vol.100, No. 9, pp.777-788, 1985.

[9] O. Collins, S. Dolinar, R. McEliece, and F. Pollara, A VLSI decomposition of the deBruijn graph, Journal of the ACM (JACM), Vol. 39, No. 4, pp.931-948, 1992.

[10] P. Faizian, et al., Random regular graph and generalized De Bruijn graph with k-shortest path routing, IEEE Transactions on Parallel and Distributed Systems, Vol. 29, No. 1, pp.144-155, 2017.

[11] A. Singla, C.Y. Hong, L. Popa and P.B Godfrey, Jellyfish: Networking data centers randomly, In 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12), pp. 225-238, 2012.

[12] P. Faizian, M. A. Mollah, X. Yuan, S. Pakin and M. Lang, Random Regular Graph and Generalized De Bruijn Graph with k-Shortest Path Routing, IEEE International Parallel and Distributed Processing Symposium (IPDPS), Chicago, IL, USA, 2016, pp. 103-112, 2016.

[13] F. Kaashoek and D. R. Karger, Koorde: A simple degree-optimal hash table, IPTPS, pp. 98-107, Feb. 2003.

[14] M. Naor and U. Wieder, Novel architectures for P2P applications: The continuous-discrete approach, ACM SPAA, pp. 50-59, June 2003.

[15] P. Fraigniaud and P. Gauron, An overview of the content-addressable network D2B, ACM PODC, pp. 151, July 2003.

[16] M.M. Rahman et al., HaVec: An efficient de Bruijn graph construction algorithm for genome assembly, International Journal of Genomics, 2017.

[17] N.H. Tran et al., Complete de novo assembly of monoclonal antibody sequences, Scientific Reports, Vol. 6, No. 1, p.31730, 2016.

[18] R. Chikhi et al., On the representation of de Bruijn graphs, In Research in Computational Molecular Biology, 18th Annual International Conference, RECOMB 2014, Pittsburgh, PA, USA, April 2-5, 2014, pp. 35-55, Springer International Publishing, 2014.

[19] M.J. Daly et al., High-resolution haplotype structure in the human genome, Nature genetics, Vol. 29, No. 2, pp.229-232, 2001.

[20] J.C. Venter et al., The sequence of the human genome, Science, Vol. 291, No. 5507, pp.1304-1351, 2001.

[21] N. Bandeira et al., Automated de novo protein sequencing of monoclonal antibodies, Nature biotechnology, Vol. 26, No. 12, pp.1336-1338, 2008.

[22] S. K. Pham, and P.A. Pevzner DRIMM-Synteny: Decomposing genomes into evolutionary conserved segments. Bioinformatics, Vol. 26, No. 20, pp. 2509-2516, 2010.

[23] M.G. Grabherr et al., Full-length transcriptome assembly from RNA-Seq data without a reference genome, Nature biotechnology, Vol. 29, N0. 7, pp. 644-652, 2011.

[24] A.V. Petyuk, On word-representability of simplified de Bruijn graphs, arXiv preprint, arXiv: 2210.14762, 2022.

[25] P.J. Roig et al., Review on de Bruijn shapes in one, two and three dimensions. In Journal of Physics: Conference Series, Vol. 2090, No. 1, p. 012047, IOP Publishing, Nov. 2021.

[26] N.G. De Bruijn, A combinatorial problem. Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam, Vol. 49, No. 7, pp.758-764, 1946.

[27] I.J. Good, Normal recurring decimals, Journal of the London Mathematical Society, Vol. 1, No. 3, pp.167-169, 1946.

[28] M.L. Schlumberger, De Bruijn communications networks, Stanford University, 1974.

[29] M.A. Fiol et al., Line digraph iterations and the (d, k)-problem for directed graphs, ACM SIGARCH Computer Architecture News, Vol. 11, No. 3, pp.174-177, 1983.

[30] Imase and Itoh, Design to minimize diameter on building-block network, IEEE Transactions on Computers, Vol. 100, No. 6, pp.439-442, 1981.

[31] T. van Aardenne-Ehrenfest and N.G. de Bruijn, Circuits and trees in oriented linear graphs, Classic papers in combinatorics, pp.149-163, 1987.

[32] E. Dubrova, M. Teslenko, and H. Tenhunen, On analysis and synthesis of (n, k)-non-linear feedback shift registers, In Proceedings of the conference on Design, automation and test in Europe, pp. 1286-1291, March 2008.

[33] C.H. Wong, Novel universal cycle constructions for a variety of combinatorial objects, Doctoral dissertation, University of Guelph, 2015.

[34] B.G. Kenkireth and A.S. Malhotra, On Word-Representable and Multi-Word-Representable Graphs, In International Conference on Developments in Language Theory, pp. 156-167, Cham: Springer Nature Switzerland, May 2023.

[35] M.M. Halldórsson, S. Kitaev, and A. Pyatkin, Semi-transitive orientations and word-representable graphs, Discrete Applied Mathematics, Vol. 201, pp.164-171, 2016.

[36] A.H. Dekker and B.D. Colbert, Network Robustness and Graph Topology, In Proceedings of the 27th Australasian conference on Computer science, Vol. 26, pp. 359-368, Jan. 2004.

[37] C. Godsil and C.F. Royle, Algebraic graph theory, Vol. 207, Springer Science & Business Media, 2001.

[38] P. Erdös and A. Rényi, On the Evolution of Random Graphs, Publications of the Math. Inst. of the Hungarian Academy of Sci., Vol. 5, pp. 17-61, 1960.

[39] A.H. Dekker and B. Colbert, The symmetry ratio of a network, In Proceedings of the 2005 Australasian symposium on Theory of computing, Vol. 41, pp.13-20, Jan. 2005.

[40] L. Zhang, L. Xiang and J. Zhu, Relationship between fragility and resilience in complex networks, Physica A: Statistical Mechanics and its Applications, 605, p.128039, 2022.

[41] S. Freitas et al., Graph vulnerability and robustness: A survey, IEEE Transactions on Knowledge and Data Engineering, Vol. 35, No. 6, pp.5915-5934, 2022.

[42] F. Safaei, S. Tabrizchi, A.H. Rasanan, M. Zare, An energy-based heterogeneity measure for quantifying structural irregularity in complex networks, Journal of Computational Science, Vol. 36, p.101011, 2019.

[43] E. Estrada, and N. Hatano, Communicability angle and the spatial efficiency of networks, SIAM Review, Vol. 58, No. 4, pp.692-715, 2016.

[44] E. Estrada, Topological analysis of SARS COV-2 main protease, Chaos: An Interdisciplinary Journal of Nonlinear Science, Vol. 30, No. 6, 2020.

Farshad Safaei received the B.*Sc*., M.*Sc*., and Ph.D. degrees in Computer Engineering from Iran University of Science and Technology (IUST) in 1994, 1997 and 2007, respectively. He is currently an associate professor in the Department of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran. His-research interests are Performance Evaluation of Computer Systems, Networks-on-Chips, and Complex Networks.

Email: f_safaei@sbu.ac.ir

Orcid Code: 0000-0002-8546-3148

Office phone: +9821-22904183

Cellular phone: +989123893561

Mohammad Mehdi Emadi Kouchak obtained his Bachelor's degree in Computer Engineering, Hardware, from the Islamic Azad University, South Tehran Branch in 2015. He further received his Master's degree in Computer System Architecture from the Islamic Azad University, South Tehran Branch in 2018. Currently, he is a Ph.D. student in Computer System Architecture at the Islamic Azad University, Science and Research Branch. His-research interests revolve around Complex and Social Networks, Design of Deep Learning Accelerators, and Quantum Computing.
Email: m.m.emadi@srbiau.ac.ir
Orcid Code: 0009-0009-2572-3356
Office phone: -
Cellular phone: +989127981452

Mehrnaz Moudi received her B.Sc. in Software Engineering, Iran in 2009 and M.Sc. in Computer Network, University Putra Malaysia (UPM) in July 2012. In 2017, she completed her Ph.D. research in computer network at Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, UPM. Her research interests are in Network Architecture, Interconnection Networks, and Parallel and Distributed Computing.
Email: mmoudi@torbath.ac.ir
Orcid Code: 0000-0002-9081-5347