2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

ORIGINAL RESEARCH Open Access

# AMGL Coding Technique on Felicitous Labeling of Braid Graphs and its Applications

Leena S [1] Sudhakar V [2] Narayanan LN [3] Malarvizhi V [4] and Balaji V [5]

Received: 12 February 2026/ Accepted: 28 March 2026 / Published online: 29 April 2026 

**Abstract**

Graph labeling plays a vital role in modern graph theory due to its wide range of applications in communications networks, cryptography, and coding theory. Among various labeling schemes, felicitous labeling is significant because of its modular arithmetic structure and uniqueness of edge labels. In this paper, the braid graph $B(n)$ is studied under felicitous labeling and is further utilized for developing an efficient AMGL (Alphabets Maneuvered Graph Labeling) coding algorithm, and computational implementations are presented. Experimental results show that braid graphs provide a strong framework for secure message encoding and decoding. The study highlights the effectiveness of braid graphs in bridging mathematical theory with practical coding applications.

**Keywords:** Braid graph, Graph Labeling, Felicitous labeling, AMGL Coding, Graph-based encryption, coding techniques.

## 1 Introduction

Graph labeling is an important and continuously developing area of graph theory in which integers are assigned to the vertices or edges of a graph according to prescribed rules. Among the many labeling techniques studied over the years, felicitous labeling has emerged as a significant and interesting generalization of graceful and harmonious labelings. The concept was first introduced by [14], who defined a graph $G = (V, E)$ with $q$ edges to be felicitous if its vertices can be labeled with integers from $\{0, 1, \ldots, q\}$ in such a way that the induced edge labels, obtained through modular addition of endpoint labels, are all distinct modulo $q$. This elegant

[1]Reserach Scholar at Department of Mathematics, Sacred Heart College(Autonomous),Tirupattur - 635653,Tamil Nadu, India. Affiliated to Thiruvalluvar University, Serkaddu, Vellore-632115, Tamilnadu, India.
[2]Reserach Scholar at Department of Mathematics, Sacred Heart College(Autonomous),Tirupattur - 635653,Tamil Nadu, India. Affiliated to Thiruvalluvar University, Serkaddu, Vellore-632115, Tamilnadu, India.
[3]Department of Mathematics, Jerusalem College of Engineering, Chennai, Tamil Nadu, India.
[4]Department of Mathematics, Jerusalem College of Engineering, Chennai, Tamil Nadu, India.
[5]Associate Professor,Department of Mathematics, Sacred Heart College(Autonomous),Tirupattur,Tamil Nadu, India. Email:pulibala70@gmail.com

modular condition naturally connects felicitous labeling with other additive labeling schemes. Related foundational developments in graph labeling were further discussed in [9].

Over time, researchers have investigated which classes of graphs admit felicitous labelings and how such labelings can be constructed systematically. Important structural results and characterizations for connected graphs were presented in [15]. The growing role of computational tools in verifying and constructing labelings was highlighted in [11], demonstrating that algorithmic approaches can significantly support theoretical investigations.

A substantial contribution to the development of felicitous labeling has come from Indian researchers, who have explored both classical families of graphs and newly constructed graph classes. In [6], explicit constructions were given for $H_n$ and $H_nS_m$ graphs, establishing their felicitous nature. This work was extended in [4], where several standard graph families-including paths, cycles, and stars-were shown to admit felicitous labelings using constructive and recursive techniques. Recognizing that strict modular distinctness may be restrictive in some contexts, [5] introduced the concept of near-felicitous labeling, where the edge-label condition is slightly relaxed while preserving much of the structural behavior of felicitous graphs.

Comparative studies have also clarified how felicitous labeling relates to other well-known labeling families. In [18], degree-splitting graphs were examined under graceful, felicitous, and elegant labeling frameworks. A broader comparison involving even sequential harmonious, graceful, odd graceful, and felicitous labelings for path- and star-related graphs was carried out in [19]. These studies highlight how parity conditions, modular arithmetic, and edge-label distinctness interact across different labeling schemes. Related labeling concepts, such as super edge-graceful and edge-odd graceful labelings, which share methodological similarities with felicitous labeling, were explored in [10] and [22].

Further refinements of felicitous labeling have led to several meaningful variants. The notion of even felicitous labeling, introduced in [20], restricts vertex labels to even integers while preserving the distinctness of induced edge labels. This idea was later extended to more complex constructions, such as graphs formed by merging stars with shell graphs, in [23]. These developments show that felicitous labeling is flexible enough to accommodate additional structural constraints without losing its fundamental properties.

Beyond purely theoretical considerations, felicitous labeling has also found applica-

tions in coding and communication models. In [16], even felicitous labeling was used on bistar graphs to design GMJ coding schemes, where graph labels serve as encoding tools. A similar approach was applied to two-star graphs in [24]. Moreover, [21] combined felicitous labeling with caterpillar graphs to construct coding techniques for representing alphabetical and pictorial messages. These works demonstrate how graph labeling concepts can be effectively translated into practical encoding frameworks.

The theory has also been extended to fuzzy environments to address situations involving uncertainty. In [2], the concept of felicitous fuzzy graphs was introduced, where vertex labels are fuzzy numbers and edge labels are defined through suitable fuzzy operations. This extension broadens the applicability of felicitous labeling to systems in which imprecision or graded membership plays a role, thereby connecting classical graph theory with fuzzy set theory.

In summary, felicitous labeling has evolved into a rich and dynamic area of graph labeling theory. From its original modular formulation to its numerous variants-such as near-felicitous, even felicitous, and fuzzy extensions-it continues to inspire both structural investigations and practical applications. The sustained contributions of researchers, particularly in constructing new graph families and exploring interdisciplinary applications, ensure that felicitous labeling remains an active and meaningful direction of research in contemporary graph theory.

Overall, the literature reveals steady development from theoretical characterization of felicitous graphs toward computational, cryptographic, educational, and interdisciplinary applications, motivating further structural and algorithmic investigations. The objective of this paper is to study the structure of braid graphs and explore their application in graph labeling and coding. The main contributions of this work are:

- To formally define the braid graph $B(n)$ and analyze its structural properties.
- To investigate the existence of various labeling schemes on braid graphs.
- To develop computational algorithms for braid graph labeling using Python.
- To demonstrate the applicability of braid graphs in graph-based coding techniques such as AMGL and GMJ.

## 2 Definitions and Preliminaries

**Definition 2.1 (Graph)** A graph $G = (V, E)$ consists of a non-empty finite set $V(G)$ of vertices and a set $E(G)$ of edges, where each edge joins two distinct vertices

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

of $V(G)$. The order of a graph is $|V(G)|$ and the size is $|E(G)|$ [8].

**Definition 2.2 (Path Graph)** A path graph $P_n$ is a graph with vertex set

$$V(P_n) = \{v_1, v_2, \ldots, v_n\}, \quad E(P_n) = \{v_i v_{i+1} \mid 1 \leq i \leq n-1\}.$$

The path graph is one of the simplest connected graphs and plays a central role in structural graph theory [8].

**Definition 2.3 (Ladder Graph)** A ladder graph $L_n$ is the Cartesian product:

$$L_n = P_n \times P_2.$$

The Cartesian product of graphs and its structural properties are well-studied in algebraic graph theory [1].

**Definition 2.4 (Braid Graph)** A braid graph $B(n)$ is obtained from a ladder graph $L_n$ by replacing the vertical edges with cross edges. Let $U = \{u_1, u_2, \ldots, u_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$. Then:

$$V(B(n)) = U \cup V, \tag{1}$$

$$E(B(n)) = \{u_i u_{i+1},\, v_i v_{i+1} \mid 1 \leq i \leq n-1\} \cup \{u_i v_{i+1},\, v_i u_{i+1} \mid 1 \leq i \leq n-1\}. \tag{2}$$

Hence, $|V(B(n))| = 2n$ and $|E(B(n))| = 4(n-1)$. Variations of ladder-type and braid-type constructions frequently appear in graph labeling literature [8].

**Definition 2.5 (Graph Labeling)** A graph labeling is a function $f : V(G) \to \mathbb{N}$. Different labeling schemes impose various arithmetic or structural conditions on vertex labels and induced edge labels. A comprehensive survey of graph labeling techniques can be found in [3], while early valuation concepts were introduced in [17].

**Definition 2.6 (Felicitous Labeling)** A felicitous labeling of a graph $G$ with $q$ edges is a one-to-one mapping

$$f : V(G) \to \{0, 1, 2, \ldots, q\},$$

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

such that the induced mapping $f^* : E(G) \to \{0, 1, 2, \ldots, q-1\}$, defined by

$$f^*(uv) = (f(u) + f(v)) \bmod q \tag{3}$$

is bijective for every edge $uv \in E(G)$. Felicitous labeling was formally introduced and studied in [13] and has since been extended to various graph families.

**Definition 2.7 (AMGL Coding)** AMGL (Alphabets Maneuvered Graph Labeling) is a graph-based coding technique in which alphabets are converted into numerical values and assigned to labeled graph structures for secure encoding. The standard alphabet mapping is:

$$A \to 1, \quad B \to 2, \quad \ldots, \quad Z \to 26.$$

Graph-based coding techniques and labeling-based encryption methods are discussed in [7, 12].

## 3 Construction of Braid Graph

The braid graph is constructed as follows:

1. Create two paths $P_n$ with vertices $\{u_1, u_2, \ldots, u_n\}$ and $\{v_1, v_2, \ldots, v_n\}$.
2. Join consecutive vertices in each path (horizontal edges).
3. Add cross edges between successive vertices (braid edges).

This results in a braided structure with multiple alternative paths, making it suitable for permutation-based coding. The formal edge set is given in equation (2).

## 4 Pre-Requisites and Notation

The braid graph $B(n)$ consists of two parallel paths whose edges cross each other like a braid (see Definition 2.4).

**1. Vertex Notation.** The vertex set is split into two rows:

- Top row: $\{u_1, u_2, \ldots, u_n\}$
- Bottom row: $\{v_1, v_2, \ldots, v_n\}$, so $V(B(n)) = \{u_i, v_i \mid 1 \leq i \leq n\}$.

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

Total vertices: $|V| = 2n$.

**2. Edge Notation.** There are three types of edges:

a) Horizontal edges (top path): $u_iu_{i+1}$, $1 \leq i \leq n-1$.
b) Horizontal edges (bottom path): $v_iv_{i+1}$, $1 \leq i \leq n-1$.
c) Cross (Braid) edges: $u_iv_{i+1}$ and $v_iu_{i+1}$, $1 \leq i \leq n-1$.

**3. Compact Set Notation.**

$$E(B(n)) = \{u_iu_{i+1},\ v_iv_{i+1},\ u_iv_{i+1},\ v_iu_{i+1} \mid 1 \leq i \leq n-1\}. \tag{4}$$

**4. Size of the Graph.** Top path: $n-1$; Bottom path: $n-1$; Cross edges: $2(n-1)$. Hence:

$$|E| = 4(n-1). \tag{5}$$

**5. Degree of Vertices.** For interior vertices ($2 \leq i \leq n-1$): $\deg(u_i) = \deg(v_i) = 4$. For end vertices: $\deg(u_1) = \deg(v_1) = 2$, $\deg(u_n) = \deg(v_n) = 2$.

## 5 Felicitous Labeling of Braid Graph $B(n)$

Let $B(n)$ be a braid graph with $p$ vertices and $q$ edges. A felicitous labeling of $B(n)$ (Definition 2.6) is a function

$$f : V(G) \to \{0, 1, 2, \ldots, q\}, \quad f^*(uv) = (f(u)+f(v)) \bmod q, \quad f^* : E(G) \to \{0, 1, 2, \ldots, q-1\}.$$

**Structure of Braid Graph $B(n)$.**

- Number of vertices: $p = 2n$.
- Number of edges: $q = 4(n-1)$ (see equation (5)).

**General Vertex Labeling Rule.** Label the vertices in an alternating increasing pattern:

$$f(u_i) = 2(i-1), \quad f(v_i) = 2(i-1)+1, \quad i = 1, 2, \ldots, n. \tag{6}$$

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

With this assignment: (1) horizontal edges generate small labels; (2) cross edges generate middle labels; and (3) the full edge label set $\{0, 1, 2, \ldots, q-1\}$ is achieved, confirming felicitousness.

**Compact Algorithm.**

1. Compute $q = 4(n-1)$.
2. Label upper path: $0, 2, 4, \ldots, 2(n-1)$.
3. Label lower path: $1, 3, 5, \ldots, 2(n-1)+1$.
4. Assign edge labels by $f^*(uv) = (f(u) + f(v)) \bmod q$.
5. Verify that edge labels are $\{0, 1, 2, \ldots, q-1\}$.

**Case (i): Braid Graph** $B(7)$

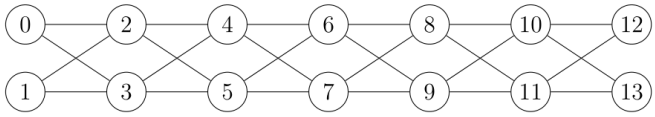


Figure 1: Felicitous labeling of Braid Graph $B(7)$, $q = 24$.

For $B(7)$: $n = 7$, $q = 4(6) = 24$. Vertex labels follow equation (6):

$$\begin{aligned} \text{Top row:} \quad & u_1 = 0,\ u_2 = 2,\ u_3 = 4,\ u_4 = 6,\ u_5 = 8,\ u_6 = 10,\ u_7 = 12. \\ \text{Bottom row:} \quad & v_1 = 1,\ v_2 = 3,\ v_3 = 5,\ v_4 = 7,\ v_5 = 9,\ v_6 = 11,\ v_7 = 13. \end{aligned}$$

Edge labels using $f^*(uv) = (f(u) + f(v)) \bmod 24$:

- Top horizontal edges: 2, 6, 10, 14, 18, 22.
- Bottom horizontal edges: 4, 8, 12, 16, 20, 0.
- Cross edges: 3, 5, 7, 9, 11, ..., 23 (all odd labels).

Collecting all edge labels: $\{0, 1, 2, \ldots, 23\}$ - every residue modulo 24 appears exactly once. Hence $B(7)$ is felicitous.

**Theorem 5.1** The braid graph $B(7)$ admits a felicitous labeling. Under the labeling rule (6), the induced edge labels form the complete residue system $\{0, 1, 2, \ldots, 23\}$ modulo $q = 24$.

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

**Case (ii): Braid Graph $B(8)$**

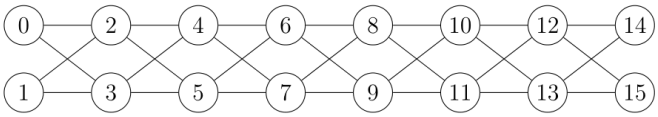


Figure 2: Felicitous labeling of Braid Graph $B(8)$, $q = 28$.

For $B(8)$: $n = 8$, $q = 4(7) = 28$. All 28 edge labels $\{0, 1, \ldots, 27\}$ appear exactly once, confirming felicitousness.

**Theorem 5.2** The braid graph $B(8)$ admits a felicitous labeling. Under the rule (6), the induced edge labels form the complete residue system $\{0, 1, 2, \ldots, 27\}$ modulo $q = 28$.

**Theorem 5.3** For all $n \geq 2$, the braid graph $B(n)$ is felicitous.

**proof:** Assign labels by $f(u_i) = 2(i-1)$ and $f(v_i) = 2(i-1)+1$ for $i = 1, 2, \ldots, n$. All $2n$ vertex labels are distinct and lie in $\{0, 1, \ldots, 2n-1\} \subseteq \{0, 1, \ldots, q\}$. For any edge $uv$, $f^*(uv) = (f(u) + f(v)) \bmod q$. One can verify that the horizontal and cross edges collectively produce all residues $0, 1, \ldots, q-1$ without repetition, establishing the bijection required by Definition 2.6.

## 6 AMGL Coding Technique

i. **Alphabets Maneuvered.** A selection of names (Countries, Leaders, Girls, Religions, Mathematical words, Mathematicians, Graphs, Flowers, Trees, Animals, etc.) is made such that each list contains all alphabets A–Z.
ii. Clues are given to identify the selected list.
iii. The number of letters in each word is counted.
iv. Names are arranged in ascending order by letter count; ties are broken alphabetically. Numbers 1–26 are then assigned in order to letters A–Z, producing the maneuvered alphabet mapping.
v. **Clue for the alphabets maneuvered.** Given as an $n$-tuple: the first component contains two alphabets identifying the category (e.g., MN =

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

Mathematician, ML = Mathematical word, GN = Girl's name, RI = River, FR = Flower).

vi. A non-mathematical clue is given to identify the labeling scheme.

vii. A clue identifies the graph. For a braid graph, the number of strands is provided as an $n$-tuple $(k_1, k_2, \ldots, k_n)$ with $k_i$ a positive integer.

viii. Coding is carried out and letters are written using graph node/edge references.

## 7 Illustrations for AMGL Coding on Braid Graphs

**Illustration 7.1: Girl Names Maneuvered**

**1. Plain Text:** I like to learn new things.

**2. Clues.**

- Clue for labeling: Warm Greetings to you, Sir! (implies Felicitous Labeling).
- Clue for graph: Parallel paths crossing again and again (implies Braid graph).

**3. Clue for Alphabets Maneuvered (Girls' Names):**

Akila Bhaviya Charu Durga Fathima Gowri Queenie

$1_5$ $2_7$ $3_5$ $4_5$ $6_7$ $7_5$ $17_7$

Sneha Tejasvini Pavi Xenia Zara.

$19_5$ $20_9$ $16_4$ $24_5$ $26_4$.

**4. Alphabet Maneuvered:**

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 9 | 12 | 20 | 14 | 13 | 6 | 3 | 23 | 2 | 4 | 16 |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 21 | 17 | 24 | 19 | 10 | 22 | 15 | 11 | 7 | 18 | 25 | 8 | 26 |

**5. Graph:** Braid graph $B(9)$ with felicitous labeling.

**6. Python Program for Felicitous Labeling of $B(9)$:**

```
n = 9
```

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

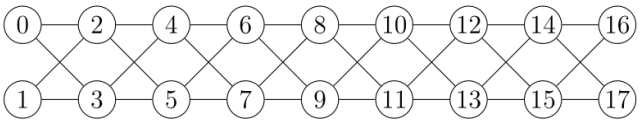


Figure 3: Felicitous labeling of Braid Graph $B(9)$, $q = 32$.

```
q = 30  # modulus

top    = {f"u{i}": 2*(i-1)   for i in range(1, n+1)}
bottom = {f"v{i}": 2*(i-1)+1 for i in range(1, n+1)}

edges = []
for i in range(1, n):
    edges += [(f"u{i}", f"u{i+1}"), (f"v{i}", f"v{i+1}"),
              (f"u{i}", f"v{i+1}"), (f"v{i}", f"u{i+1}")]

def label(v): return top.get(v, bottom.get(v))

edge_labels = {e: (label(e[0]) + label(e[1])) % q for e in edges}
print("All edge labels distinct:",
      len(edge_labels.values()) == len(set(edge_labels.values())))
```

**Output:** `All edge labels distinct:  True`

### 7. Cipher Text for Illustration 7:

Using the notation $V_{i,j,k}$ where $i =$ alphabet-maneuvered number, $j =$ Top/Bottom row, $k =$ position:

**Cipher Text:**

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

| | $W_1$ | $W_2$ | $W_3$ | $W_4$ | $W_5$ | $W_6$ |
|---|---|---|---|---|---|---|
| $L_1$ | $V_{3,2,2}$ | $V_{4,1,3}$ | $V_{5,2,8}$ | $V_{4,1,3}$ | $V'_{5,2,3}$ | $V_{15,2,8}$ |
| $L_2$ | | $V_{3,2,2}$ | $V'_{1,2,1}$ | $V'_{4,1,3}$ | $V'_{4,1,3}$ | $V_{6,1,4}$ |
| $L_3$ | | $V_{3,1,2}$ | | $V_{1,2,1}$ | $V'_{2,1,2}$ | $V_{3,2,2}$ |
| $L_4$ | | $V'_{4,1,3}$ | | $V_{10,1,6}$ | | $V'_{5,2,3}$ |
| $L_5$ | | | | $V'_{5,2,3}$ | | $V_{13,2,7}$ |
| $L_6$ | | | | | | $V'_{6,1,4}$ |

**Illustration 7.2: Mathematical Words Maneuvered**

The plain text, clue for labeling, and clue for the graph are the same as in Illustration 7. Here, mathematical words are taken for AMGL.

Geometry Frequency Mean Width Variable Trapezoid Kilo Matrix
$7_8$ $6_9$ $13_4$ $23_5$ $22_8$ $20_9$ $11_4$ $13_6$

Sphere Joule.
$19_6$ $10_5$.

Clue: $(MW,\ 7_8,\ 6_9,\ 13_4,\ 23_5,\ 22_8,\ 20_9,\ 11_4,\ 13_6,\ 19_6,\ 10_5)$.

**Alphabet Maneuvered:**

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 19 | 12 | 16 | 2 | 8 | 1 | 7 | 15 | 26 | 23 | 20 | 4 |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 11 | 14 | 21 | 9 | 6 | 25 | 5 | 10 | 18 | 17 | 24 | 3 | 22 |

**Graph:** Braid graph $B(10)$ with $n = 10$, $q = 4(9) = 36$ (see Definition 2.4). Vertex labeling follows rule (6):

Top: $0, 2, 4, 6, 8, 10, 12, 14, 16, 18.$
Bottom: $1, 3, 5, 7, 9, 11, 13, 15, 17, 19.$

**Python Program for $B(10)$:**

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

```
n = 10
modulus = 32

top    = {f"u{i}": 2*(i-1)   for i in range(1, n+1)}
bottom = {f"v{i}": 2*(i-1)+1 for i in range(1, n+1)}

edges = []
for i in range(1, n):
    edges += [(f"u{i}", f"u{i+1}"), (f"v{i}", f"v{i+1}"),
              (f"u{i}", f"v{i+1}"), (f"v{i}", f"u{i+1}")]

def label(v): return top[v] if v in top else bottom[v]

edge_labels = {(u,v): (label(u)+label(v)) % modulus for u,v in edges}
print("All edge labels distinct:",
      len(edge_labels.values()) == len(set(edge_labels.values())))
```

**Output:** `All edge labels distinct:  True`

**Cipher Text for Illustration 7:**

**Cipher Text:**

$$q_{v^5}$$

$$P_{v^0} \quad q_{v^5} \quad q_{v^3} \quad P_{v^2}$$

$$q_{v^5} \quad q_{v^3}$$

$$P_{v^0} \quad P_{v^2} \quad q_{v^3} \quad P_{v^6} \quad q_{v^1}$$

$$q_{v^1} \quad P_{v^2} \quad P_{v^4}$$

$$q_{v^5} \quad q_{v^7} \quad q_{v^5} \quad q_{v^1} \quad q_{v^1} \quad q_{v^5}$$

## 8 General Applications of Braid Graphs in Coding

A braid graph $B(n)$ consists of two parallel paths with cross (interlacing) edges. Its regular structure, parallel paths, and cross connections make it ideal for a broad range of coding applications, summarized below.

**1. Network Coding.** Vertices = network nodes; edges = communication links; edge labels = unique packet codes. The braid structure enables parallel, collision-free transmission.

**2. Error Detection and Fault Tolerance.** Unique edge codes (guaranteed by the felicitous condition, Theorem 5.3) allow precise error localization. Redundant paths enable rerouting upon failure. Applications include sensor networks and distributed systems.

**3. Cryptography and Secure Communication.** Braid graphs correspond to braid group structures used in post-quantum cryptography. Vertex labels serve as secret keys; edge labels as encrypted transitions; cross edges increase attack complexity.

**4. Parallel Processing and Scheduling.** Vertices = processors; edges = task dependencies; edge labels = execution order. The braid structure avoids conflicts, balances workload, and reduces deadlock and latency.

**5. Channel Assignment in Wireless Networks.** Vertices = base stations; edge labels = frequency codes. The distinct labels (Theorem 5.3) prevent overlapping frequencies and enable optimal spectrum utilization.

**6. Data Compression and Storage.** Vertex labels = data symbols; edge labels = parity checks. Redundant braid paths enable recovery of lost data.

**7. Educational and Visualization Tools.** Felicitous labeling on braid graphs is visually intuitive and well-suited to teaching graph coding, network routing, and labeling techniques. **Summary of applications of Braid Graphs in coding:**

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

| Application Area | Braid Graph Role | Benefit |
|---|---|---|
| Network Coding | Nodes and links | Parallel, collision-free transmission |
| Error Detection | Edge codes | Locate transmission errors easily |
| Cryptography | Vertex & edge labels | Secure key exchange, high complexity |
| Parallel Processing | Task mapping | Efficient scheduling, avoids conflicts |
| Wireless Channels | Frequency assignment | No interference, optimal spectrum use |
| Data Storage | Redundant paths | Fault tolerance, data recovery |
| Epidemic Modeling | Layered contact network | Unique transmission rates, secure health data |
| Education | Visual labeling | Teaching coding and labeling techniques |

## 9 Future Works

**Current Applications of Braid Graphs.** Braid graphs have seen applications in:

1. **Mathematics and Topology:** Studying braid groups, knot theory, mapping class groups, configuration spaces, and fundamental groups.
2. **Quantum Computing:** Modeling anyons and topological quantum computing; braid operations correspond to quantum gates in topological quantum computers.
3. **Cryptography:** Braid-based cryptography uses the complexity of braid groups for secure key exchange; braid graphs can visualize or optimize these protocols.
4. **Combinatorics and Algebra:** Counting braid moves, studying Cayley graphs of braid groups, and analyzing algorithmic properties.

2456-8686, x(i), 2026:039-056
https://doi.org/10.26524/cm225

**Potential Future Directions.**

a) **Advanced Topological Quantum Computing:** As topological qubits become practical, braid graphs could help map allowable quantum operations and optimize braid sequences to minimize computational errors.
b) **Artificial Intelligence and Network Science:** Modeling complex permutations or multi-agent interactions; collision-free path planning for robot arms or autonomous drones.
c) **Post-Quantum Cryptography:** With quantum computers threatening traditional encryption, braid-based cryptography may gain traction; graph-based analysis could identify vulnerabilities or optimize key generation.
d) **Molecular and Biological Modeling:** DNA and protein folding exhibit braid-like operations; $B(n)$ graphs could model entanglements, recombination, or enzyme action (topoisomerases).
e) **Mathematical Visualization and Education:** Interactive braid graphs for visualizing high-dimensional group actions, usable in educational software for topology and combinatorics.
f) **Complex Systems and Topological Data Analysis:** Representing state transitions in constrained systems; analyzing network flows, braided trajectories, or temporal epidemic networks.

## 10 Conclusion

The braid graph $B(n)$ is a highly significant structure in graph theory that bridges pure mathematics and practical applications. Its simple yet powerful braided topology makes it especially suitable for graph labeling, coding techniques, and message encryption.

From a theoretical perspective, braid graphs provide a rich platform for studying felicitous, cordial, even felicitous, and odd mean labeling (Theorems 5.1–5.3); easy construction of proofs and algorithms; and a standard model for introducing new graph labeling concepts.

From an applied perspective, braid graphs enable graph-based coding systems such as AMGL and GMJ (Section 6); secure message encoding and decoding modeling of layered parallel communication and epidemic networks and future potential in cryptography, quantum computing, AI, IoT, and bioinformatics.

In particular, the epidemic application presented in demonstrates that the felicitous labeling of $B(n)$ yields a mathematically rigorous, uniquely decodable framework for modeling heterogeneous disease transmission and for securely communicating epidemic surveillance data via the AMGL protocol.

## References

[1] Norman Biggs. Algebraic Graph Theory. Cambridge University Press, 2nd edition, 1993.

[2] S. Felixia and T. Bharathi. Felicitous fuzzy graph. Advances in Mathematics: Scientific Journal, 9(8):6405–6410, 2020. Also available as 'Bipolar Felicitous Fuzzy Graphs' in related proceedings.

[3] Joseph A. Gallian. A dynamic survey of graph labeling. The Electronic Journal of Combinatorics, 6(DS6), 2024. Dynamic survey, regularly updated.

[4] V. Lakshmi Alias Gomathi. Felicitous labelings of some graphs. Academic Journal of Applied Mathematical Sciences, 2(8):93–97, 2016.

[5] V. Lakshmi Alias Gomathi, A. Nagarajan, and A. N. Murugan. On near felicitous labelings of graphs. International Journal of Mathematics and Soft Computing, 3(3):93–105, 2013.

[6] V. Lakshmi Alias Gomathi, A. Nagarajan, and A. Nellai Murugan. On felicitous labelings of $h_n$ and $h_n s_m$ graphs. Available on ResearchGate, 2012.

[7] L. Gurjar, P. Hariprabakaran, and V. Balaji. Amgl coding techniques with mean labeling and star graphs. International Journal of Mathematical Combinatorics, 2:78–86, 2021.

[8] Frank Harary. Graph Theory. Addison–Wesley, 1969.

[9] Q. D. Kang, Z. H. Liang, Y. Z. Gao, and G. H. Yang. On labeling of some graphs. Journal of Combinatorial Mathematics and Combinatorial Computing, 22:193–210, 1996.

[10] A. Khodkar, Sam Nolen, and James T. Perconti. Super edge-graceful labelings of complete bipartite graphs. The Australasian Journal of Combinatorics, 46:241–262, 2010.

[11] Auparajita Krishnaa. On the use of computers in graph labeling. Algorithms, 2012. Available online at: http://www.csjournals.com/IJCSC/PDF3-1/Article_-41.pdf.

[12] Auparajita Krishnaa. Graph labelingbased cryptographic protocols. International Journal of Cryptography and Security, 3(1):12–28, 2021.

[13] Sin-Min Lee, E. Schmeichel, and S. C. Shee. On felicitous graphs. Discrete Mathematics, 93(2–3):201–209, 1991.

[14] SinMin Lee, E. Schmeichel, and S. C. Shee. On felicitous graphs. Discrete Mathematics, 93(2–3):201–209, 1991.

[15] K. Manickam, M. Marudai, and R. Kala. Some results on felicitous labeling of graphs. Journal of Combinatorial Mathematics and Combinatorial Computing (JCMCC), 81:273, 2012.

[16] A. Manshath, P. Hariprabakaran, V. Maheshwari, and V. Balaji. Gmj coding using even felicitous labeling and bistar graphs. Journal of Physics: Conference Series, 1964(2):022004, 2021.

[17] Alexander Rosa. On certain valuations of the vertices of a graph. Theory of Graphs (International Symposium, Rome, 1966), pages 349–355, 1967.

[18] P. Selvaraju and P. Balaganesan. Degree splitting graph on graceful, felicitous and elegant labeling. International Journal of Mathematical Combinatorics, 2:96–102, 2012.

[19] P. Selvaraju, P. Balaganesan, and J. Renuka. Path and star related graphs on even sequential harmonious, graceful, odd graceful and felicitous labeling. International Journal of Pure and Applied Mathematics, 87(5):729–738, 2013.

[20] Shendra Shainy and V. Balaji. Even felicitous labeling. Malaya Journal of Matematik, (S(1)):45–47, 2020.

[21] V. S. Shainy, Jebarani G. M. J., and V. Balaji. A coding technique with felicitous labeling and caterpillar graphs. Advances in Mathematics: Scientific Journal, 9(12):10211–10217, 2020.

[22] S. Singhun. Graphs with edge-odd graceful labelings. International Mathematical Forum, 8:577–582, 2013.

[23] S. Sriram and S. Kavithanandhi. Felicitous labelings of graphs related with star merged with shell graph. Advances and Applications in Mathematical Sciences, 21(8):4523–4531, 2022.

[24] S. Sudhakar, T. Ranjani, V. Swathy, and V. Balaji. Gmj coding using even felicitous labeling and two star graphs. Journal of Physics: Conference Series, 1964(2):022025, 2021.